3D SHAPE ESTIMATION OF TENDON-DRIVEN CATHETERS USING ULTRASOUND IMAGING

by

Niloufaralsadat Hashemi
MEng, University of Toronto, 2013
BASc, University of Toronto, 2012

A thesis presented to Ryerson University

in partial fulfillment of the requirements

for the degree of

Master of Applied Science

in the program of

Biomedical Engineering

Toronto, Ontario, Canada, 2018

AUTHOR'S DECLARATION

AUTHOR'S DECLARATION FOR ELECTRONIC SUBMISSION OF A THESIS

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I authorize Ryerson University to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize Ryerson University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

I understand that my thesis may be made electronically available to the public

RYERSON UNIVERSITY

ABSTRACT

3D Shape Estimation of Tendon-Driven Catheters Using Ultrasound Imaging

by Niloufaralsadat Hashemi

Master of Applied Science

Biomedical Engineering, 2018

Active cable/tendon-driven catheters are becoming an established part of the minimally invasive surgical procedures. Therefore, there has been growing interest in literature in estimating the shape of their distal end especially using clinical ultrasound (US) imaging systems. The purpose of this thesis is to use a B-mode US imaging system to design time-efficient, accurate and robust algorithm for 3D shape estimation of tendon-driven catheters. Kalman filter (KF), Adaptive Kalman filter (AKF) and Particle filter (PF) algorithms were developed for this purpose. First, they were applied to a series of simulated US B-mode images where AKF provided the best estimate (error: 0.2 ± 0.1 mm). Second, they were applied to a series of experimentally obtained US B-mode images. Calibration procedures were carried out to calibrate these US images in the experiment's workspace. The PF was shown to provide the best 3D shape estimate (error: 8.6 ± 0.1 mm). However, since almost the same accuracy could be achieved with AKF in ten times less computational time, AKF was concluded to be the best method, in terms of accuracy and efficiency, to estimate the 3D shape of tendon-driven catheters.

ACKNOWLEDGEMENTS

Firstly, I would like to express my sincere gratitude to my supervisor, Professor Farrokh Janabi Sharifi and co-supervisor, Professor Jahan Tavakkoli for their guidance and providing the funds and resources required by this thesis.

I would also like to thank my fellow lab mates, Kaiqi Cheng, Yasser Ali, Ali Mehrkish and Somayeh Norouzi Ghazbi for their kind support throughout this research.

Finally, I am grateful to be a part of Ryerson University and the department of Biomedical Engineering that provided an opportunity for me to learn and grow with the support of an amazing faculty and staff.

DEDICATION

I would like to dedicate this work to my parents, Seyed Gholam Reza Hashemi & Maryam Pasaran, and to my brother and sister, Amir Farokh Hashemi & Bahar Hashemi, for their continuous encouragement and unconditional support.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Number *Page*

LIST OF ABBREVATIONS

ABVS	Automated Breast Volume Scanner
AE	Active Element
AKF	Adaptive Kalman Filter
ARFI	Acoustic Radiation Force Impulse Imaging
AR	Autoregressive model
bpm	Beats per minute
C	Conference Paper
CAD	Computer Aided Design
CL	Closed Loop Control System
CMM	Coordinate Measuring Machine
CR	Continuum Robots
CT	Fluoroscopy, Preoperative Computed Tomography
CTA	Comet Tail Artifacts
DH	Denavit–Hartenberg
DICE	Sørensen–Dice index
DOF	Degree(s) Of Freedom
DSL	Diffractive Side Lobe Artifacts
FBG	Fiber Bragg Grating
ECG	Electrocardiography
EM	Electromagnetic
EKF	Extended Kalman Filter
FOB	Flock of Birds
FOS	Fiber Optic Sensor
FPS	Frames Per Second
G	Global Coordinate System
GPU	Graphic Processing Unit
HAM	Homotopy Analysis Method
HD	Haussdorff Distance
I	Image Coordinate System
IBVS	Image-Based Visual Servoing
IP	Image Processing
IVUS	Transcatheter Intravascular Ultrasound
J	Journal Paper
KF	Kalman Filter
LARS Robot	Laparoscopic Assistance Robotic System
MCI	Motion Compensation Instrument
MI	Mutual Information
MI	Mechanical Index
MIS	Minimally Invasive Surgery
MRI	Magnetic Resonance Imaging

N/A	Not Available/Specified
NCC	Normalized Cross Correlation
NID	Needle Insertion Device
P	Ultrasound Probe Coordinate System
PH	Phantom (fCal2.0) Coordinate System
PBVS	Position-Based Visual Servoing
PCA	Principle Component Analysis
PD	Proportional Derivative controller
PF	Particle Filter
PID	Proportional Integral Derivative Controller
PLA	Polylactic Acid
R	Reference Coordinate System
RAA	Range Ambiguity Artifacts
RANSAC	Random Sample Consensus
RCM	Remote Center of Motion
RF	Radiofrequency
RMS	Root Mean Square
RMSE	Root Mean Square Error
ROI	Region of Interest
Rot	Rotational motion about x-, y-, z- axes
$\mathrm{Rot}_{x,\alpha}$	Rotation about the x- axis by angle of α
$\mathrm{Rot}_{y,\beta}$	Rotation about the y- axis by angle of β
$\mathrm{Rot}_{z,\gamma}$	Rotation about the z- axis by angle of γ
RSPR	Revolute, Spherical, Prismatic, Revolute joints
RRT	Rapidly Exploring Random Trees
RT	Real Time
SCV	Sum of Conditional Variances
SIR	Sequential Importance Resampling filter
SNR	Signal-to-Noise Ratio
SOS	Speed Of Sound
SSD	Sum of Squared Differences
SVD	Singular Value Decomposition
T	Thesis Paper
TAVI	Trans-catheter Aortic Valve Implantation
TEGAC	Tubular Enhanced Geodesic Active Contours
TGC	Time Gain Compensation
TIS	Thermal Index
TPS	Thin Plate Spline
Trans	Translational motion along x-, y-, z- axes
$\mathrm{Trans}_{x,\alpha}$	Translation along x- axis by α
$\mathrm{Trans}_{y,\beta}$	Translation along y- axis by β
$\mathrm{Trans}_{z,\gamma}$	Translation along z- axis by γ

UKF	Unscented Kalman Filter
US	Ultrasound
VMI	Volumetric Medical Imaging
VOI	Volume of Interest
VRCM	Virtual Remote Center of Motion
wrt	With respect to

LIST OF SYMBOLS

Symbol	Description	Units
$\hat{x}$	An estimate of x	
$\bar{x}$	Mean of x or relative value	
x^*	Desired value of x	
$\mathbf{v}$	A vector	
$\left\|\|\mathbf{v}\|\right\|$	Scalar norm or length of vector **v**	
$\tilde{\mathbf{v}}$	Homogeneous representation of vector **v**	
$\mathbf{v}^{\{\boldsymbol{F}\}}$	Vector **v** with respect to the coordinate frame F	
$\mathbf{A}$	A matrix	
$\mathbf{A}^{+}$	Pseudo inverse of matrix **A**	
$\mathbf{A}^{-1}$	Inverse of **A**	
$\mathbf{A}^{T}$	Transverse of **A**	
$\mathbf{A}_{i,j}$	The element (i, j) of **A**	
$\mathbf{A}[\mathrm{i},\mathrm{j}]$	The element (i, j) of **A**	
$\oplus$	Pose composition	
$\ominus$	Inverse of a pose (unary operator)	
$\cdot$	Transformation of a point by a relative pose, e.g., $\xi.\mathbf{p}$	
{F}	Coordinate frame F	
a	Ratio of two sides of similar triangle: mm/mm	
$\left\{a_t^{[j]}\right\}_{t=1}^{T}$	Measure of goodness (i.e. normalized probability weight) of the [j]th particle in total of T stages in the PF algorithm, $t \rightarrow 1\ to\ T\colon \left\{a_1^{[j]}, \ldots a_T^{[j]}\right\};\ j \rightarrow 1\ to\ J$	
c_θ, s_θ	$\cos\theta\,, \sin\theta$	
$d(\mathbf{r}_1, \mathbf{r}_2)$	The Euclidean distance between two points $\mathbf{r}_1$ and $\mathbf{r}_2$	
$D_{\mathbf{w}}$	Larger dimension of the vector **w**, representing world state in KF, AKF, and PF	
$D_{\mathbf{x}}$	Larger dimension of the vector **x**, representing the measurement state in KF, AKF, PF	
Δ_k	Observation noise sample matrix at step k in AKF	
Δt	Change in time/time increment	s

$\boldsymbol{\Sigma}_{ij}$	Covariance between i and j: the degree of correlation between ith state variable and jth state variable	
$\boldsymbol{\Sigma}_{m}$	Measurement model parameter in KF: measurement noise covariance matrix	
$\boldsymbol{\Sigma}_{p}$	Transition process noise covariance matrix (i.e. dynamic noise covariance matrix) in KF	
$\boldsymbol{\Sigma}_{+}$	Estimated/Predicted covariance matrix of the marginal posterior probability in KF	
$\boldsymbol{\Sigma}_{0}$	Initial estimate of covariance matrix of the marginal posterior probability in KF	
$\boldsymbol{\Sigma}_{t}$	Covariance update matrix (at stage t) of marginal posterior probability in KF	
$\boldsymbol{\epsilon}_{m}$	Measurement noise matrix	
$\boldsymbol{\epsilon}_{p}$	Transition noise matrix	
F_d	Desired force	N/m
F_e	Applied force	N/m
J	Total number of particles in PF algorithm (i.e. total # of predictions/hypotheses)	
$\mathbf{K}$	Kalman gain	
k_f, k_{fi}	Controller gains	
$\hat{\mathbf{L}}_{\mathbf{s}}^{+}$	Pseudo inverse of an estimation interaction matrix	
λ	Controller gain	
$\boldsymbol{\mu}_{m}$	Measurement model parameter in KF algorithm: Vector representing mean change in measurements	
$\boldsymbol{\mu}_{p}$	Temporal model parameter in KF: vector representing mean change in state	
$\boldsymbol{\mu}_{+}$	Estimated/Predicted mean state vector of the marginal posterior probability in KF	
$\boldsymbol{\mu}_{0}$	Initial estimated mean vector of marginal posterior probability in KF	
$\boldsymbol{\mu}_{t}$	Mean state update vector (at stage t) of marginal posterior probability in KF	
$\boldsymbol{\mu}_{\boldsymbol{P}\{\boldsymbol{F}\}}$	Mean of data set P that is measured in coordinate frame of F	
$\boldsymbol{\Phi}$	Measurement model parameter in KF: Matrix relating measurement vector to state vector	
$\boldsymbol{\Psi}$	Temporal model parameter in KF: transition matrix relating the current estimated state to the next state	

$\mathbf{q_R}$	Four-vector unit quaternion representing rotation between two data sets	
$\mathbf{q_T}$	Translation vector	
$\mathbf{q}_i, \mathbf{q}_k$	State noise sample at iterative stages i, k in AKF	
$\widehat{\mathbf{q}}, \overline{\mathbf{q}}$	Sample mean: unbiased state estimator for q in AKF	
$\mathbf{R}$	An orthonormal rotation matrix	
SE(n)	Special Euclidean group, the set of all poses in n dimensions	
$\mathbf{t}$	Translation vector	
$\mathbf{T}$	Homogeneous transformation matrix, T ∈ SE(2) or SE(3)	
$\mathbf{T}_B^A$	Homogeneous transform representing frame {B} with respect to frame {A}. If {A} is not given then assumed relative to world coordinate frame	
$\theta_r, \theta_p, \theta_y$	Roll pitch yaw angles	rad
α, β, γ	Roll pitch yaw angles in the Rotational motion about x-, y-, z- axes	
u, v	Image plane coordinates	Pixels
$\dot{u}, \dot{v}$	Rate of change of image plane coordinates	Pixels/FPS
u_c, v_c	Central pixel coordinate of the ROI	Pixels
$\dot{u}_c, \dot{v}_c$	Rate of change of central pixel coordinate of the ROI	Pixels/FPS
u_0, v_0	Coordinates of the principle point	Pixels
v_p	Velocity applied to virtual US probe	m/s
$\mathbf{w}_t$	World state vector at stage t	
$\left\{\mathbf{w}_t^{[j]}\right\}_{t=1}^{T}$	World state hypothesis of particle # j in total of T stages in PF algorithm: t → 1 to T, $\left\{\mathbf{w}_1^{[j]}, \dots \mathbf{w}_T^{[j]}\right\}$; $j \rightarrow 1\ to\ J$	
$P(\mathbf{w}_t^{[j]})$	The probability associated with state vector **w** of particle #j at timestep t (in PF algorithm)	
$\widehat{\mathbf{w}}_+^{[j]}$	Evolved states sampled from temporal distribution in PF algorithm: $\Pr(w_t \mid w_{t-1} = \widehat{w}_{t-1}^{[n]})$	
x_d	Desired position	m
x_e	Position of moving range	m
x_f	Position offset required to maintain the desired force	m
$\mathbf{x}_t$	Measurement state vector at stage t	
X, Y, Z	Cartesian coordinates	

ξ	Abstract representation of 3-dimensional Cartesian pose
ξ_B^A	Abstract representation of 3-dimensional relative pose, frame {B} with respect to frame {A}

Chapter I

INTRODUCTION

1.1. Background

Active cable/tendon-driven catheters are a subclass of continuum robots (CRs), which are defined as actuatable structures whose constitutive material forms curves with continuous tangent vectors [1]. These interventional catheters are becoming an established part of the minimally invasive surgical (MIS) procedures as of 2010 [2], notably in neurosurgery (Figure 1.a) [3]-[5], otolaryngology (Figure 1.b) [6]-[7], ophthalmic surgery (Figure 1.c[1]) [8]-[9], cardiac surgery (Figure 1.d) [10]-[13], abdominal surgery (Figure 1.e) [14]-[15], vascular surgery (Figure 1.f) [16]-[17], and urologic surgery (Figure 1.g) [18]-[19]. Not only do they provide curvilinear and flexible accessibility through dynamic anatomical environments and small incisions of MIS procedures, but also allow 1-2 degrees of freedom (DOF) mechanism at their proximal end to actuate their distal end and generate curvatures in two or three dimensions [20]-[22].

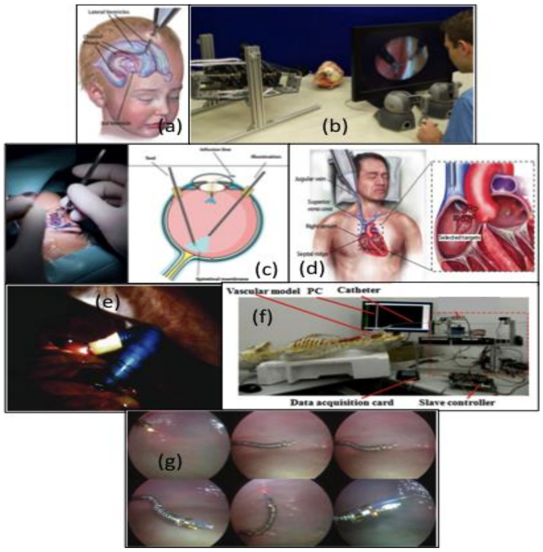


Figure 1 - Examples of tendon-driven catheters in clinical applications: a) neurosurgery [3]-[5] b) otolaryngology [6]-[7] c) ophthalmic surgery[1] [8]-[9] d) cardiac surgery [10]-[13] e) abdominal surgery [14]-[15] f) vascular surgery [16]-[17] g) urologic surgery [18]-[19]

[1] https://www.aop.org.uk/ot/CET/2016/11/14/keeping-an-eye-on-robotics/article

Despite their advantageous characteristics as applicable to MIS procedures, accurately sensing and estimating the 3D shape of tendon-driven catheters in real-time remains a challenge [20], [23]. This is due to their inherent hyper flexibility and inevitable collisions with the anatomy in remote operations while inside the organs, which present unknown and dynamic payloads [20], [23]-[25]. Even so, the real-time 3D shape information of the interventional catheter is beneficial in that it helps to estimate external incident forces [26], to plan obstacle-free and occlusion-free trajectory toward the target [27], and to control its shape in navigation.

The recent advances in 3D shape estimation are based on three categories of alternative emerging techniques [19], [23]: fiber-optic-sensor (FOS)[2] [25], [28], electromagnetic (EM) tracking [29] and intraoperative imaging modality [24], [30]-[31]. In FOS-based shape reconstruction method, multiple Bragg Grating Sensor fibers (typically 3-4) are arranged around the circumference of the catheter at fixed distance about the neutral axis of bending [32] to measure strain, force, torque, displacement, temperature and pressure [20], [33]. Although FOS/FBG-based shape sensing methods can provide accurate and fast measurements of the shape of tendon-driven catheters, their integration with small and hyper flexible catheters is challenging and adds significant cost [24], [34]. EM-based sensors have also been used in-vivo to position the tip of the catheter or isolated number of points along its length; however, these sensors are easily susceptible to interference of ferromagnetic materials and electrical noise [32], [33]. They also introduce extra loading and wiring requirements, which could affect catheters operation. In comparison to the aforementioned, vision-based shape sensing techniques with available imaging modalities are preferred since they do not impose hardware modifications to the tendon-driven catheters and therefore can measure the catheter's shape without obstructing its flexibility or interfering with its kinematics [34]. Additionally, imaging modalities are readily available and do not introduce extra expenses into the operation costs.

[2] Fabricated with Fiber Bragg Grating (FBG) technology

Imaging modalities studied for intraoperative image-based shape estimation techniques are magnetic resonance imaging (MRI) [35]-[36], x-ray fluoroscopic imaging [37]-[38], endoscopic imaging [10], [39], and ultrasound (US) imaging [39]-[40]. In MRI, soft tissue is imaged with better resolution and there are no radiation exposures; however, they are expensive, bulky and their intraoperative use is limited because of their inherent magnetic forces and electromagnetic interferences [35]. In addition, MRI has a low imaging frame rate and is not appropriate for most real-time applications [33]. X-ray fluoroscopic imaging depends on biplane C-arm systems that expose the patient to high ionizing radiation dosage and are therefore not suitable for continuous use. In addition, they are expensive and impose operational workspace constraints during intravascular interventions [20]. Conventional 2D x-ray fluoroscopy also lacks depth perception and cannot directly visualize the anatomic structures [53]. As an alternative to MRI and x-ray, US imaging could provide shape (and in particular depth) detection of interventional catheters at low cost, in real-time, and without exposure to ionizing radiation [20]. The use of US has been recognized as intraoperative image guidance in MIS procedures such as peripheral and central venous access [42], needle-based biopsy [43]-[44] and RF ablation for liver [45] and lung cancers/tumors [46], and cardiac catheterization [13], [20], [47]-[49]. Previous studies did use US not only to track the instrument’s tip [50]-[52] but also to realize the sensing and estimation of its shape in real time [20], [53]-[54]. Nonetheless, ultrasound-driven shape reconstruction algorithms suffer high computation cost and low accuracy mainly due to ultrasound’s low resolution, signal-to-noise ratio (SNR), and a variety of imaging artifacts [20].

1.2. Research Objectives, Specific Aims and Contribution

Given its prevalence and aforementioned advantages compared to other imaging and sensing techniques, the objective of this research project is to use conventional 2D US imaging system to design time-efficient, accurate and robust algorithm for 3D shape estimation of the tendon-driven cardiac catheter.

Due to the dynamic nature of an operation, robust tracking algorithms must be designed and tested for the application at hand. Specifically the aims of this project is to apply Bayesian tracking algorithms of Kalman filter (KF), Adaptive Kalman filter (AKF) and Particle filter (PF) to a series of transvers US images obtained from scanning across the length of at least three differently configured catheters. Algorithms are first applied to a series of artificially obtained US transverse images (numerical simulations in MATLAB) and then are applied to a series of experimentally obtained US transverse images.

The contribution of this thesis is to present shape estimation of catheter in 3D space of a global coordinate frame by syncing together US images and Transducer probe pose information obtained while scanning across catheters. On one hand, Bayesian tracking algorithms (i.e., KF, AKF, PF) are developed to track catheter cross section in the sequence of transverse 2D US images. On the other hand, Optotracker sensor is used to keep track of the US probe pose information as the probe scans over the catheter. Image-based calibration is also conducted to map imaging coordinates to the probe coordinates.

1.3. Outline

The remaining of this thesis is organized into five chapters. In the second chapter a background review of US-guided robotic interventions is provided. Within the context of this chapter the importance of 3D shape estimation of surgical instruments becomes apparent. This chapter ends with a review of literature published solely considering different methods that estimate the 3D shape of cardiac catheters using US imaging modality. In chapter three, a solution to the problem of 3D shape estimation of tendon-driven catheters using US data is presented using three different Bayesian tracking algorithms: Kalman filter (KF), Adaptive Kalman filter (AKF) and particle filter (PF). This chapter further discusses how numerical simulations can validate the applicability and reliability of these algorithms by applying them to a series of synthesized US image frames within which a known structure of catheter is depicted. Chapter four describes *ex-vivo* experiments conducted using SONIXTOUCH Q+ research US machine[3] and NDI Optotracker[4]. A series of conventional transverse 2D US images of 3D printed configurations emulating tendon-driven catheters are

[3] Ultrasonix Medical ULC, Richmond, BC

[4] Northern Digital Inc., Waterloo, Ontario

obtained in the coordinate frame of the NDI Optotracker system. In chapter five the KF, AKF and PF algorithms are implemented on data obtained from the experiments. The results are compared and validated against the actual 3D shape of the catheter (which is independently obtained using Coordinate Measuring Machine (CMM[5])). Finally in chapter six, the results of the KF, AKF and PF algorithms from the numerical simulation are discussed against the ones obtained from experimental data. The speed and accuracy of each of the three algorithms is further weighed against its applicability in the context of the complete surgical platform involving real-time US-guided robotic targeting.

[5] Mitutoyo, Takatsu-ku, Kawasaki, Kanagawa

Chapter II

ULTRASOUND-GUIDED FLEXIBLE ROBOT-ASSISTED MINIMALLY INVASIVE SURGERY: A REVIEW

There is a growing interest on US image-guided robotic MIS (RMIS) [13], [41], [51], [55]-[103]. These surgical platforms are comprised of many components pertaining to their intended clinical application. In what follows, a survey of the state of the art of each component within this surgical platform is provided. Throughout the following sections, open problems in US-guided RMIS are identified and more specifically, the importance of 3D shape estimation of the surgical tool for the optimization of US-guided robotic visual servo control systems becomes more apparent.

2.1. Surgical Instruments considered in US-guided RMIS

The type of surgical tool used dictates the type of robot chosen for its manipulation, modeling mathematics used in robotic control systems, path/trajectory planning strategy, tracking and shape estimation algorithms. Therefore a discussion of commonly chosen surgical tools in literature for US-guided minimally invasive surgical robotic platforms is in place.

The primary surgical tool around which the US-guided minimally invasive surgical robotics is designed is the asymmetric beveled-tip flexible needle [55]-[62], [64], [78]-[83], [92]-[94], [96]-[100]. Percutaneous needle insertion used for diagnostic and therapeutic applications such as biopsy and brachytherapy is one of the most common minimally invasive surgical procedures [55]-[62], [96]-[97]. These 0.5-1 mm diameter Nitinol needles are usually beveled at 30° to easily penetrate a soft tissue and therefore minimize patient discomfort [55]. In addition, given their thin diameter, they are flexible and can be steered around obstacles (e.g. sensitive tissues) through curved needle paths. Steering is performed by a combination of insertion and rotation at the base of these needles using a robotic system [55]. As the needle is robotically steered in the soft tissue, it deflects along a curved trajectory in the direction of the bevel tip [59]. Consequently, there are two main mathematical models discussed in literature that concern bevel-tip needles. These are kinematics-based unicycle model of the needle, which assumes that the needle tip follows a circular path, and needle-tissue interaction mechanics-based model, which predicts deflection using needle-tissue interaction forces [55]-[62]. These mathematical models are at the core of any control system and path planning algorithms concerning bevel-tip flexible needles.

Radiofrequency (RF) ablator needle probes are the next in the list of surgical tools used in US-guided robotic surgical platforms. Radiofrequency ablation is one of the most promising minimally invasive techniques for the treatment of liver cancer where malignant tumors are unresectable such as hepatocellular carcinoma, colorectal metastases, neuroendocrine tumors, or other types of metastases [73]-[76], [104]-[105]. The rigid needle probes are usually 14–17.5 gauge, 15–25 cm long, insulated cannulas containing one to three straight needle electrodes or five to ten individual hook-shaped electrode arms or tines used for liver cancer biopsy and ablation [105]. In US-guided robotic surgical platforms, these needles are either inserted manually using a passive needle guide [75] or with a driver robot which aligns and positions the needle automatically [76]. In either case, there is no need for path/trajectory planning inside the liver tissue as the rigid needle's angle and length of penetration is planned before the insertion. Therefore, there is no mathematical modeling of these ablator needles in literature [73]-[76]. Aside from RF ablator needles, other rigid hollow cannula needles of 1.5-2 mm in diameter have been considered in the literature [66], [71]. However, if not for liver malignant tumor biopsy/ablation, these rigid needles are not favorable for MIS applications (e.g. prostate brachytherapy [55]) as they cause deformation of tissue, which can result in target motion affecting the targeting accuracy.

Forceps are also discussed in literature [85]-[88], [91], as the surgical instrument within the US-guided robotic surgical platform. These minimally invasive instruments with a tooltip composed of two jaws [91], are usually made out of polyurethane (PUR), polyvinyl chloride (PVC) or nylon, which yields good echoes for the purpose of optimum US detection [85]. The geometrical model and kinematic model are two main mathematical representation of forceps upon which the controlling algorithms rely on. In the geometrical model [86]-[88], the instrument is modeled with three straight lines intersecting at a point which is used to express the instrument coordinates in the US images as a function of instrument pose. Furthermore, the kinematic model of the forceps is devised by observing the angular and translational velocity of the tip of the forceps in the US plane [86]-[88].

Again these mathematical models are an essential component of the US-guided surgical robot control system algorithm.

Finally, next only to bevel-tip needles, continuum/concentric tube robots (e.g., cardiac catheters [13], [41], [70], [72] and flexible actuated tip needle [63]) are the most common minimally invasive surgical instruments considered in the literature for US-guided surgical robotic platforms. Cardiac catheters (e.g. commercially available Artisan Control Catheters), which are also the surgical tool of choice in this thesis, are long and thin flexible tubes that are inserted into the vascular system and passed into the heart to perform minimally invasive procedures such as measuring cardiac physiological function, dilating vessels and valves, and implanting prosthetics and devices [70]. Due to their flexible structure and since they are inserted through turbulent organ such as the heart, which altogether may impose unpredictable backlashes, experiments conducted with cardiac catheters in literature are designed to restrict the motion of the catheter in a controlled manner. For example, in a study a flexible sheath (out of Teflon or Nylon tubing) is already manually inserted through the heart vasculature before the catheter guidewire is robotically driven through it [13], [70], [72]. This frame of work reduces the catheter performance considerations only to guidewire-sheath interactions such as friction forces and backlash behavior due to gap size between the sheath and guidewire [13], [70], [72]. Friction forces are described with Coulombic model approximation and backlash behaviors are quantified as the width of the backlash hysteresis curve [13], [70], [72]. Of course, there has also been a study where the insertion of the cardiac catheter is not restricted by any pre-inserted sheath [41]. In this case, the cardiac catheter is a concentric tube robot comprised of three telescoping curved sections inserted into the jugular vein and navigated to the right atrium of the heart [41]. Another type of concentric tube robot is the flexible actuated tip needle used for diagnostic and therapeutic purposes such as biopsy and ablation [63]. These active needles can change their shape either at the tip (consisted of a conical tip mounted on a ball joint) or along the entire length (consisted of four tendons routed through the shaft of the needle and attached to the conical tip) and can be steered in any direction [63]. The non-holonomic kinematics of a bicycle that models the bevel-tip flexible needle is adapted for this actuated-tip needle as well which assumes that the needle tip follows a circular path [63]. In addition, the tissue surrounding this needle, which prevents sideways motion of the needle, is modeled using four Pfaffian constrains[6] assigning zero velocities at the needle tip in x and y directions [63].

2.2. Ultrasound Imaging Modalities used in US-guided RMIS

In this section, different US imaging modalities used in US-guided surgical robotics is discussed which is a useful information in that the same modalities must be used in these platforms to estimate the shape of the intended surgical tool.

In US-guided robotic surgical platforms, the US imaging is used to localize the target tissue and the surgical instrument during insertion [61]. After processing the acquired US frames, these images are used as feedback in visual servo algorithms to guide the robotic system to steer the surgical instrument to reach the localized target position while avoiding obstacles [61].

In US-guided robotic surgical platforms, B-mode/2D linear probes are most commonly used to acquire US frames continuously during the insertion procedure [55]-[62], [64]-[65], [73]-[77], [82]-[83], [85]-[93], [102]-[103]. The US frames have also been reported to be acquired using B-mode/2D curvilinear transducers [51], [81], [94]. The curvilinear or convex transducers are similar to linear transducers except that their elements are arranged on a curved rather than a flat surface [106]. This format produces B-mode images in a sector or pie slice shape and is often described by a field of view (FOV) angle specifying its lateral angular extent [106]. The curved transducers are more accommodating for scans over curved surfaced specimen than the linear transducers [106]. Figures 2 and 3 illustrate typical linear and curvilinear US images.

[6] In motion planning, a Pfaffian constraint is a set of k linearly independent constraints over velocity in the form of $\mathbf{A}(\mathbf{q})\dot{\mathbf{q}} = 0$. [132]

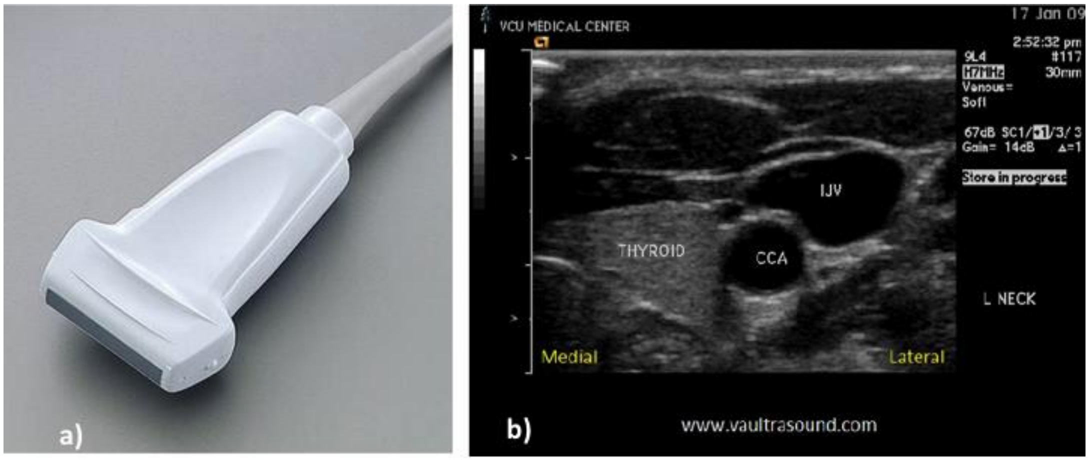


Figure 2 – a) Linear US transducer probe, b) Linear US probe B-mode image plane [133]

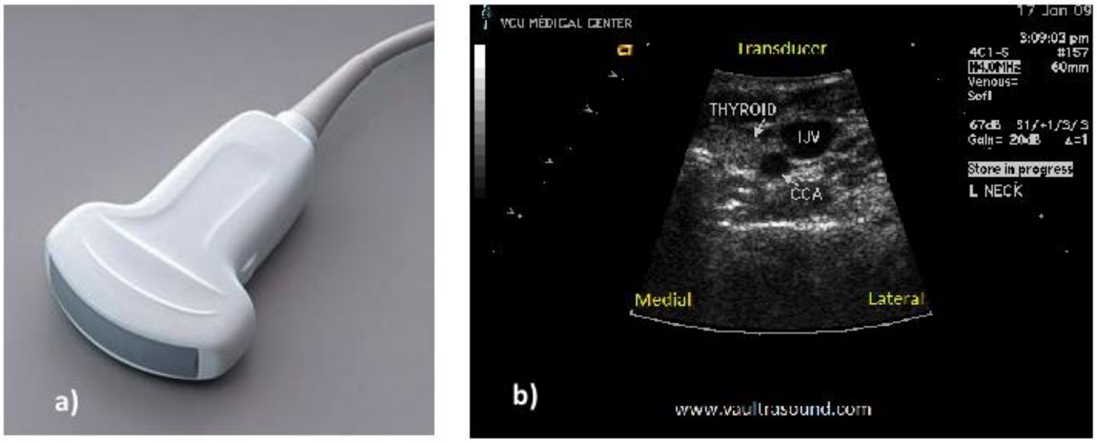


Figure 3 – a) Curvilinear US transducer probe, b) Curvilinear US probe B-mode image plane [133]

In addition to B-mode 2D linear/curved US probes, the use of 3D US scanners are also well established in US-guided surgical robotics [13], [41], [63], [66]-[72], [78]-[80], [84], [95]-[99]. In one study, the use of convex 3D transducer is also introduced [94]. In 3D imaging, scanning in both xz and yz planes are combined to form a pyramidal shaped volume scan [106]. Such volumetric acquisition can either be obtained through a matrix array transducer [95]-[97] or through a motorized 2D transducer [78]-[80]. Figure 4 illustrates the mechanism of matrix array transducer and Figure 5 sketches the mechanism of motorized 2D transducer.

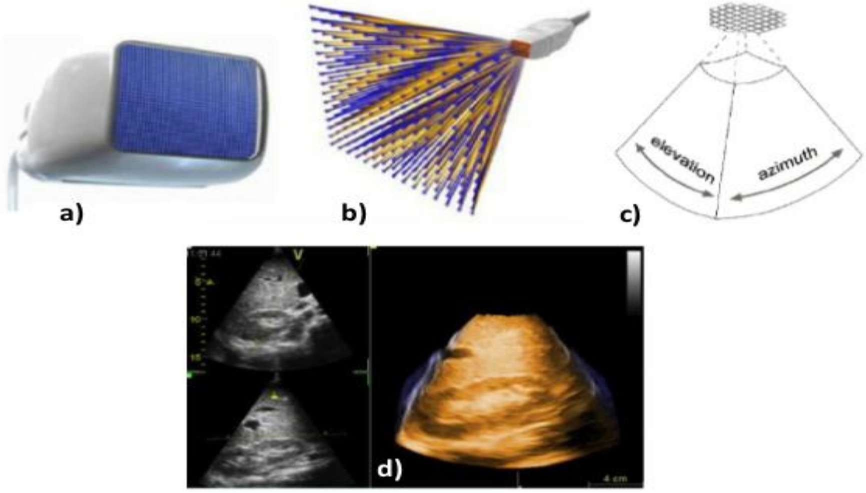


Figure 4 – a) Matrix probe using a 2D array of elements, b) the beam can be steered in two directions, c) a truncated pyramid of data is acquired, d) 4D (3D+t) US of the liver [134]

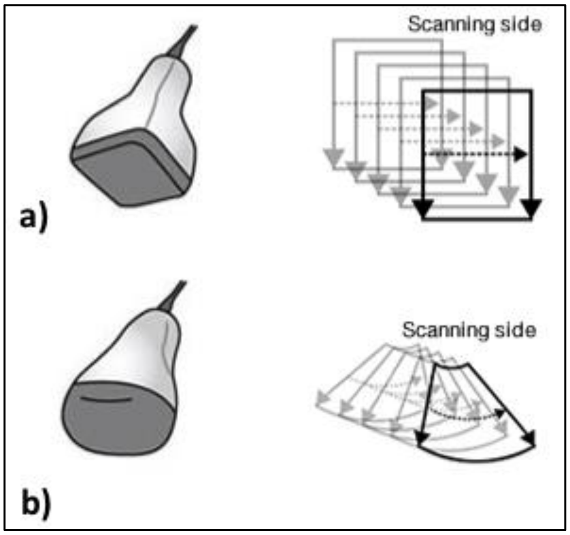


Figure 5 – a) Motorized linear type: 3D probe with linear scanning type, b) motorized convex (curved) type: 3D probe with the wide field of view [135]

Aside from conventional linear/curved US transducers, specialized US transducers are considered specifically for minimally invasive procedure of brachytherapy, namely, Transrectal US (TRUS), which exist both in linear (2D) [107] and matrix arrays (3D) [84]. TRUS, which is also known as prostate sonogram or endorectal US, is used to look at the prostate and tissue around it in US-guided robotic brachytherapy or biopsy procedures. In these procedures, the thin hollow needle, while being monitored by the US probe, is steered through the wall of the rectum into the prostate to either perform a biopsy [108] or plant seeds for brachytherapy [84], [108].

Summing up, robotic surgical instrument insertion platforms commonly use conventional clinical US imagers to guide the instrument and to improve the instrument's placement accuracy [56]. In some literature 2D ultrasound is used to assist the robotically inserted instrument, but the movement is limited to the 2D image plane [51], [56], [90]. On the other hand, real-time 2D ultrasound images can be used to construct a volume within which instruments can be tracked [109]. Such volumetric construction with 2D US frames is also focus of this thesis within which the shape of the cardiac catheter is estimated or in other words the cross section of the catheter is tracked using probability temporal models, which will be explained in the next chapter. However, it is important to note that the volume reconstruction using incoming 2D US frames is a compromise between its size and 2D frame acquisition time [56]. Furthermore, as also mentioned in the previous paragraph, tracking surgical instruments such as cardiac catheters using 3D US images has also been demonstrated [50], [56], [110]. These modern 3D US transducers used especially for real-time applications have limited voxel resolution, which limits accurate surgical instrument tip detection up to 3 mm [56], [111]. In addition to poor accuracy, 3D US suffers from low frame rate (28 Hz) and a time delay of up to about 100 ms in acquisition and processing of 3D US volumes, during which time the heart's annulus can potentially recoil 15 mm during the minimally invasive catheter heart surgery [67]-[69].

2.3. Robot Hardware Specifications in US-guided RMIS

An understanding of common robotic hardware specifications considered for experiments using US-guided surgical robotics is useful in developing shape estimation algorithms for the surgical tools, as it is favorable to use the available robots to automate the procedure of shape estimation.

Minimizing the invasiveness of surgeries involves the development of procedures in which surgeons no longer need to directly touch or see the structures on which they operate [112]. Robots are the integral part of such procedures by taking the task out of the hands of the surgeon and performing it through miniscule cuts and insertions with much more accuracy, reliability and repeatability. That being said, the robots in the current clinical platforms are far from the intended accuracy, reliability and repeatability, which is why a look into literature on complexity of robots commonly used in respective experiments is of interest.

In US-guided robotic minimally invasive surgical platforms either only the surgical instrument is robotically controlled [13], [51], [55], [65]-[72], [76], [79]-[80], [85]-[91], [94], [96], or both the US probe and the surgical instrument are robotically controlled [56]-[64], [73], [75], [80]-[83], [89]-[90], [92]-[93], [95], [97]. In studies where both the surgical instrument and the US probe are robotically controlled, some have both the instrument and the probe controlled by a single robot [80], [95], [97] and some have each controlled by a separate robot [56]-[62], [81].

The type of robot used to maneuver the surgical instrument depends on the type of the instrument. For example, the bevel-tip flexible needles are usually driven by needle insertion devices, which have 2 degrees of freedom. These devices manipulate the needle at its base by providing one translational DOF along the insertion axis and one rotational DOF about the insertion axis [55]-[64], [83], [94]. These robots can derive the needle transnationally with speed of about 0.4-10 mm/s and can rotate the needle with rotational speed of about 31.4 rad/s [55]-[64]. Aside from these conventional 2DOFs needle insertion devices, there are other mechanisms of needle insertion for bevel-tip flexible needles discussed in literature. For example, in one brachytherapy study, the needle insertion is semi-automatic, where the surgeon manually inserts the needle (manual lateral manipulation of the needle base) while the bevel-tip location is controlled robotically (automatic axial rotation of the bevel-tip needle) [82]. This study argues that such needle insertion mechanism where the surgeon is in charge of needle insertion is favorable since it ensures safer operation and continuous professional engagement, while the needle tip bevel location is controlled robotically [82]. To derive bevel-tip flexible needles in their respective experiments, other studies also have reported the use of 3DOF Gantry III Cartesian robot linear motion system with an xy stage and z-axis slide inclined at 45°, 6DOF revolute, spherical, prismatic, revolute (RSPR) parallel robot, 6DOFs Viper s650[7] with 6 rotational joints, 7DOFs revolute joints iARM assistive robotic manipulator, and 7DOFs revolute joints KUKA LBR iiwa R800 robot[8] [51], [79]-[81], [95]-[97]. The more degrees of freedom the chosen robot has in positioning and insertion of the needle, the more complicated the control of robotic steering will be which in turn introduces more insertion uncertainties and inaccuracies. That is why in these experiments 6DOFs force/torque sensors are attached at each joint of the robots to measure needle insertion forces and torques [51].

In papers experimenting with thermal RF ablator needles, the needle driver robot that aligns and positions the needle automatically is comprised of 3DOFs Cartesian stage with three prismatic joints and a 2DOF motorized rotational stage [74]-[76]. In these experiments EM tracking system (6DOFs FOB) interfaces the surgical workstation for easier calibration of needle holder with US probe [74]-[76]. Other types of automation for rigid needles include 3DOF PHANToM robotic arm[9] [65]-[68] and Motion Compensation Instrument (MCI) [69]. MCI includes a linear actuator for 1DOF translational 5.4 cm range of motion and includes a high linearity potentiometer for position sensing with accuracy of 0.01 mm [69].

In literatures related to the control and steering of cardiac catheters, a prototype is proposed where it works pretty much like the aforementioned MCI device. This drive system at the base of the catheter consists of a single actuated linear DOF accommodating a maximum speed of 210 mm/s [13], [70], [72]. Specifically, it consists of a linear voice coil actuator, linear slide, linear potentiometer position sensor and miniature force sensor [13], [70], [72]. Using this mechanism, the cardiac catheter is driven inside a flexible sheath, which has already been manually inserted through the heart vasculature [13], [70], [72].

Other types of continuum robots, in particular flexible actuated-tip needle and concentric tube robots are also worth mentioning in terms of their mechanism of automation. In one study, flexible actuated-tip needle is steered via a linear stage [63]. However, in addition to this simple 1DOF translational

[7] Omron Industrial Automation, Kyoto, Japan

[8] KUKA Robotics, Mississauga, Ontario. The 7th DOF corresponds to the gripping ability of the robot.

[9] SensAble Technologies, Washington, MA.

motion, this needle can also actuate its tip using the four tendons through its flexible sheath. These four tendons accommodate steering direction angle in addition to the needle tip orientation [63]. In another study, the mechanism of a concentric tube robot that can be inserted into the jugular vein and navigated to the right atrium of the heart is also discussed [41]. This concentric tube robot consists of three telescoping curved sections: a long proximal section for navigation through the jugular vein (segment 1N) and two distal manipulation sections (segments 2M and 3M) for tissue manipulation and device deployment inside the atrium (in other words, used to position and orient the robot tip) [41].

Automation of forceps is also mentioned in a number of papers [85]-[88], [91]. In one study, the forceps is automated using 7DOFs Mitsubishi PA10[10] robot comprised of revolute joints where only joints 2-7 are utilized. Another example of forceps automation is with MC2E robot, which provides 4DOFs at the instrument tip [85]-[88]. In the surgical scenario, the instrument is introduced into the heart through a trocar fixed on the heart wall [85]-[88]. Hence, only 4 Intracardiac DOFs remain [85]-[88]. Three DOFs are the x-, y-, z- components of the angular velocity of the instrument tip with respect to the US probe and the fourth DOF is the translational velocity of the instrument tip along the instrument axis [85]-[88].

This marks the end of the survey on automation mechanisms for flexible robotic-assisted surgical systems and the rest of this section focuses on surveying the type of robots used for maneuvering the US probes. It is important to note that the robot maneuvering the US probe strives to fulfill the following goals as best as possible. The first goal is that the US image should visualize the instrument tip and overall track the instrument, keeping it well within the field of view defined by the specific US probe [61]. The second goal is that the image plane should be always perpendicular to instrument's insertion axis as much as possible, which can usually be achieved by moving the transducer with variable velocities to keep the instrument in the field of view [61]. The third goal is the maintenance of constant contact force between the US probe and surface being scanned which

[10] DFKI Robotics Innovation, Bremen, Osnabrück.

can be accomplished using force control [61]. The fourth goal is that the robot should allow the US probe to scan over curved surfaces [61].

In an attempt to simplify the experiments to achieve the first three goals intuitively, a lot of studies discard the consideration for the fourth goal in their set up. In these studies the US probe is usually maneuvered using Cartesian robots that provide only translational motion along xy plane (2DOFs) [56]-[57], [59]-[60], [62], [64], xyz volume (3DOFs) [58], [95], [102] or only along a 1DOF linear stage [63], [82]-[84]. In addition in these experiments the US only scans straight surfaces.

To better the aforementioned studies, which discard the fourth goal of US robots, one study adds on top of the Cartesian robot a 2DOFs rotational mechanism for transducer's roll and pitch movement [61]. In addition force/torque sensors are employed to measure the contact force applied to the transducer [61]. In another study, for the similar purpose of achieving the fourth goal, an IBM/JHU[11] LARS robot is used where in addition to three axes Cartesian base, it employs two axis instrument carrier providing rotation about an instrument shaft and translation motion toward or away from the remote center of motion (RCM) point [75]. In this study, US probe instrument carrier is equipped with 6DOFs force/torque sensor to keep track of the contact force of the US probe with the uneven surface being scanned [75]. In addition, a Flock of Birds (FOB) EM tracking system (6DOFs interfaces the surgical workstation for easier calibration of the robots, US probe and surgical instrument [75]. The aforementioned studies in this paragraph add 2DOFs mechanisms to 3DOFs Cartesian robot, which in total provides the automation of US probe with a 5DOF robotic unit.

Robots with 6DOFs and 7DOFs are also considered in literature for the manipulation of the US probe in their respective experiments. In one study, a UR5 (which is a 6DOFs lightweight industrial robot arm with rotational joints) is chosen to manipulate the US probe [77], [100]. The advantage of using UR5 robot is that the force exerted by it on any surface can be limited by presetting a threshold, which guarantees the operator and patient's safety [77], [100]. In other studies choosing 6DOFs robots, one can note Viper s650[12] [78] and KUKA KR 6 R9000 sixx[13] [92]-[93] both with all

[11] John Hopkins University, Baltimore, MD.

[12] Omron Industrial Automation, Kyoto, Japan

[13] KUKA Robotics, Mississauga, Ontario.

rotational joints manipulated to keep the surgical instrument tip in the 2D US image plane. In another study a prototyped 6DOFs base positioning system is used where the US automation is a combination of conventional three prismatic joints Cartesian positioner and three revolute joints added on top of it allowing the US transducer and needle manipulator to be aligned with the vessel orientation in venipuncture applications [101], [103].

Studies choosing 7DOFs robots to manipulate the US probe in their experiments include Laparoscopic Assistance Robotic system (LARS robot) [73], KUKA LBR Intelligent Industrial Work Assistant (iiwa) R800 robot[13] [80]-[81] and iARM Assistive Robotic Manipulator [97]. The LARS robot is a kinematically redundant manipulator with three-axis linear Cartesian motion stage, two-axis parallel four bar linkage providing two rotations along the *x* and *y* axis about the RCM point, and two-axis distal component providing an insertion motion and rotation about the instrument axis [73]. KUKA LBR only consists of revolute joints with one of them being redundant as it corresponds to gripping abilities of the robot [80]-[81]. In these studies, high accuracy torque sensors in the seven joints of the robot are incorporated [80]-[81]. In the iARM robot coordinates are read and written in arrays of six values, three of which describe Cartesian coordinates and three describe the orientation of the robot gripper hand (yaw, pitch, roll) [97]. The seventh and redundant DOF of the iARM robot corresponds to its gripping ability [97].

2.4. Control Algorithms in US-guided RMIS

An understanding of control system algorithms used in literature for US-guided surgical robotics is useful in developing applicable shape estimation algorithms that can be used to optimize the performance of these control system.

In this section different control system algorithms for US-guided minimally invasive robotic surgery platforms are discussed. Most notable of the main control algorithms include: position based visual servo control algorithms (PBVS) [51], [55]-[64], [95], [99], [102]-[103], image based visual servo control algorithms (IBVS) [41], [66]-[67], [78], [85]-[90], [96], [100], nonlinear model predictive control [91], motion compensation control for involuntary rhythmic motions (e.g. heartbeat) which often consists of feed-forward Coulomb friction model compensation and backlash inverse compensation [13], [68]-[72], force control algorithms [72], and virtual RCM control algorithm [74], [76].

Position-based visual servoing uses visual features observed through the US and the known geometric model of the target to determine the pose of the target with respect to the US probe [113]. For instance, in a study the target's 3D center coordinate is estimated by combining electromagnetic (EM) tracking of the US probe with centroid of segmented US cross section images of target (i.e. surgical instrument being inserted) [65]. The robot then moves toward that pose and the control is performed in the three-dimensional task space [113]. Several robust algorithms exist for pose estimation but they are computationally expensive and rely critically on the accuracy of the US probe calibration and the model of the object's geometry [113]. In a number of studies, closed loop (CL) PBVS control is used to derive both Cartesian robot manipulating the US probe and the 2DOF needle insertion device [55]-[62]. In these papers, the inputs to the control system are (the preoperatively determined) centroid of the target (or/and obstacles [57]) and the real-time US detected surgical instrument (i.e., needle) tip pose in the form of Cartesian coordinates (x, y, z, roll, pitch, yaw) [55]-[62]. It is important to mention that Kalman state observer[14] is also implemented in these papers to minimize the noise in real-time US estimation of the needle tip pose [55]-[62]. Given these inputs, the CL PBVS control system commands the needle insertion device to perform duty cycled needle steering[15] [55]-[62], [64]. Also, it derives the US's Cartesian robot x-axis using compensator and gain scheduler, y-axis using a simple PD controller and z-axis using alignment control algorithm based on force and torque feedback from a sensor attached to transducer control robot [57], [64].

In a different study, the US scanner does not employ any control system; instead, it scans the surface of interest at a constant speed of 1.55 mm/s [63]. At the same time US linear stage motor encoder measures the position of the transducer and therefore the anticipated needle tip [63]. This information

[14] A statistical optimal estimation algorithm used to estimate instantaneous states of a system from indirect and uncertain measurements [114]

[15] As flexible bevel tip needles are inserted into tissue, the stiffness of the tissue acting on the non-symmetric needle tip deflects the needle [115]. By constantly spinning the needle during insertion, the bevel angle is essentially used to direct the needle through a specific trajectory. Therefore, incorporating duty-cycled spinning during needle insertion provides proportional control of the curvature of the needle trajectory through tissue. [115]

is fed into the CL PBVS control system of the needle insertion device which compensates for the needle velocity out of US plane by reducing the needle insertion speed to 1.4 mm/s or increasing it to 1.7 mm/s [63].

Image-based visual servoing omits the pose estimation step and uses the US image features directly [113]. The control is performed in the US image 2D space [113]. The desired US probe pose with respect to the target is defined implicitly by the image feature values at the goal pose [113]. IBVS is a difficult control problem since the US image features are a highly non-linear function of camera pose [113]. Despite the challenge, some studies have employed IBVS in the steering of their surgical instruments within the US-guided robotic surgical platform [41], [66]-[67], [78], [85]-[90]. IBVS can be used to control the automatic steering of the surgical instrument by measuring its tip position error in the US image and employing PD controller for compensation and position control [66]-[67]. In one study, CL IBVS control is used to provide feedback for duty-cycling control strategy that guides the needle toward a surgeon defined target whose coordinates are expressed in US volume frame [79]. Usually in literature implementing IBVS, the surgeon selects a desired instrument target location on US image [86]-[88].

CL IBVS is also employed in compensating for target anatomy motion visible through 3D US imaging [80]. In this case, intensity based similarity functions (e.g., Normalized Cross Correlation or NCC) are used to compute the misalignment and using this information, the visual control law determines new desired pose for US transducer [80]. Not only for the robot controlling the US probe but also for the insertion robot, CL IBVS can be used to compensate for involuntary patient motion in milliseconds and any other possible instrument slippages [90].

One study also simulates IBVS control on a virtual US probe [41]. In this case, the control velocity applied to the virtual US probe is computed to minimize the visual error between the current visual features extracted from US image and the desired visual features [41]. To exponentially decrease the visual error, a standard classic proportional control law is used [41], [78], [85]-[88], [100]:

$$\mathbf{v}_p = -\lambda \hat{\mathbf{L}}_{\mathbf{s}}^{+}(s(t) - s^{*}),$$

where, $\mathbf{v}_p$ is the velocity applied to the virtual US probe, λ is the controller gain, $\hat{\mathbf{L}}_{\mathbf{s}}^{+}$ is the pseudo-inverse of an estimate of interaction matrix that relates the variations of visual features to $\mathbf{v}_p$, $s(t)$ is the current visual feature extracted from US image and s^* is the desired visual feature [41]. In other non-virtual scenarios, force feedback or impedance control scheme[16] is used in conjunction with IBVS to maintain constant pressure of the US probe on the body while the probe is automatically guided to keep the surgical instrument within the field of view [78].

Nonlinear model predictive control is also studied in the context of visual servoing (applicable to both PBVS and IBVS) [91]. The nonlinear predictive model ensures that the instrument is positioned where desired more productively while satisfying constraints such as joint limits, actuator saturation, and visibility preserving [91]. Therefore, the proposed nonlinear model predictive control system incorporates the robot model, US image projection model, physical limits and model error adjustments [91]. Another study employs Unscented Kalman Filter (UKF), which is a nonlinear temporal model, in the context of IBVS [94]. In this study, UKF accommodates accurate closed loop robotic steering control of the needle tip in constant 5 mm incremental insertions toward the target defined in 3D US coordinate system [91].

In a number of studies motion compensation control for involuntary rhythmic motions such as heartbeat is implemented [13], [68]-[72]. In these studies, the rhythmic movement of the heart is modeled using Extended Kalman Filter (EKF) and this trajectory is fed-forwarded to the robot controller for synchronization in real-time [68]. In other words, the robot (i.e., the MCI machine) is to follow the motion of the mitral valve at e.g., 60 beats per minute (bpm) [69]. Simultaneously, the instrument and target measurements (i.e., segmented from 3D US volumes in real-time) are used to automatically register the robot and imaging coordinate frames and also to position control at 1kHz using PD or PID algorithms [13], [68], [70]. To accommodate the robotic motion compensation, these studies also employ feed forward Coulomb friction compensation, which uses friction predictor to feed forward an additional force to compensate for the existing friction (i.e., in this case between the cardiac catheter and the pre-inserted sheath) [13]. In addition to friction compensation, backlash

[16] i.e. mass-damper-spring relation between robot end effector position and external force for required US contact force [80]

inverse compensation is also employed which measures the backlash (i.e., in this case for the catheter guidewire inside the pre-inserted sheath within the heart valve) and uses the inverse value to specify the correction offset required to drive the system through the desired trajectory [13]. Another control algorithm employed in the progression, is the force control with the purpose of applying a desired force on a fast moving target (i.e. within the beating heart) with the robotic catheter end effector [72]. To achieve this, the drive system follows a desired position, x_d, defined by:

$$x_d = x_e + x_f,$$

where x_e is the position of the moving target and x_f is the position offset required to maintain the desired force [72]. The PID controller running at 1kHz takes care of this position offset [72]:

$$x_f = K_f(F_d - F_e) + \ K_{fi} \int (F_d - F_e)dt,$$

where K_f and K_{fi} are controller gains, F_d is the desired force and F_e is the applied force [72].

There are also control methods in literature to manipulate the needle placement robot to align percutaneous needles using virtual RCM control algorithms, which eliminate the need for expensive, difficult to calibrate RCM mechanisms [74], [76]. Specifically, for percutaneous needle placement procedures, surgical robots with remote center of motion (RCM) wrists have demonstrated utility [116]. However, due to the complicated mechanical design and their need for calibration and registration to the medical imager prior to each use, these robots are not widespread in clinical applications [116]. As the result, some studies propose a virtual RCM control algorithm that only requires online tracking or registering of the needle to the imager (i.e., 6DOF pose of the needle from EM tracker) and a 5DOF un-calibrated robot of three prismatic and two rotational joints that does not require preoperative registration (i.e., the orientation of the robot base with respect to the tracker base station suffices) [74], [116]. With the virtual RCM algorithm, the robot executes RCM motion "virtually" without having a physically constrained fulcrum point [116]. In the virtual RCM algorithm, a fast convergent incremental adaptive motion cycle running on heuristic function[17]

[17] An Artificial Intelligence searching method

guides the needle to the insertion point and aligns it with the target within very few cycles [74], [116]. In other words, the robot performs incremental motions and after each motion it checks to see if the needle is becoming more aligned or less [74], [116]. This information determines which direction is likely to result in more accurate alignment with the directional vector from body entry point to the target [74], [116].

2.5. Path/Trajectory Planning in US-guided RMIS

Shape estimation algorithms can also be used to optimize the performance of path/trajectory planning algorithms. Therefore an understanding of commonly used path/trajectory planning algorithms in literature is in place.

In minimally invasive surgeries using US-guided surgical robotics, the path or trajectory through which the surgical instrument must travel to get to the intended target must be clearly defined [57]-[58]. The clearly defined path is then used to command the control system toward the intended trajectory. Most of the time in literature, the surgeon or operator manually sets the path of the surgical instrument toward the target (i.e., preoperatively at the image guided surgical planning station) [41], [51], [55], [74]-[76], [80], [84], [98]-[99], [101]. To avoid complicated path/trajectory planning, some studies instruct the insertion robot to follow a constant velocity path [66]-[67]. Another study employs a simple trajectory planner that calculates the needle path by interpolating from start and end points at a 20 Hz cycle rate using an exponential decay function [102].

Notably, the Rapidly Exploring Random Trees (RRT) algorithm is used in a number of papers to calculate the path/trajectory of the surgical instrument (i.e. bevel-tip needle) [57], [59]-[62], [64]. In this method, the desired path is calculated in a sequence of 6 mm milestones based on needle curvature kinematic model and pre-operative US target and obstacle(s) localization [57], [59]. In addition, this calculated path is updated intra-operatively according to real time needle tip position obtained from US images while avoiding virtual (stationary or moving)/real (stationary only) obstacles [57], [59]. The RRT algorithm is placed inside the closed loop control system and executed repeatedly at 1 Hz until target is reached [60]. At each re-planning step, out of hundreds of calculated plans, the one with the shortest needle path while avoiding obstacles is selected [60]. The output of

RRT path planning algorithm is a set of calculated needle rotation and insertion specifications that can be used in duty-cycled spinning strategy to steer the bevel-tip flexible needle to the target [61].

In another study, a path/trajectory planner based on graph-based search algorithm is reported [82]. In this method, given target and obstacle locations, the path planner computes a large number of paths. These calculated paths are updated intra-operatively using Homotopy Analysis Method[18] (HAM) [82]. HAM-based needle deflection predictor is a real time predictor based on needle curvature information obtained from US images [82]. It is used for estimating future needle deflection as it is steered inside soft tissue [82]. The output of the path planner is a set of needle rotations at 180° that will steer the needle tip through the shortest path to the target while avoiding obstacles [82].

2.6. Visual Tracking methods used in US-guided RMIS

US-based instrument tracking can also benefit from the incorporation of shape estimation algorithms for the surgical robots. Therefore a survey of different tracking methods in literature provides a context of how the incorporation of shape estimation algorithms can be applicable.

In literature concerning the US-guided minimally invasive robotic surgical platforms, tracking in 2D/3D US is either only implemented for the localization of the surgical tool tip [13], [51], [56]-[64], [66], [68]-[70], [74]-[75], [77]-[79], [83], [85]-[88], [91]-[92], [94]-[99], [100]-[101], or both the localization of the tool tip and the target at the same time [41], [55], [67], [71]-[73], [76], [80]-[81], [89]-[90], [93], [98]-[99]. The real time surgical tool and/or target localization in 2D/3D US provided by tracking algorithms is used to update the closed loop control systems and path/trajectory planners and therefore it plays an integral part in literature regarding US-guided surgical robotics.

Many different image processing algorithms applicable to B-mode US images are the central part of the solution to the tracking problem. For instance, the tracking of the flexible needle tip in 2D

[18] HAM is a mathematical tool based on concepts in topology and differential geometry used to describe continuous variation or deformation (e.g. a continuous deformation of a circle into an ellipse can be mathematically described using homotopy between the two functions describing the circle and the ellipse) [82]. In this paper, the concept of HAM is used to estimate the continuous deformation of the flexible bevel-tip needle as it is being inserted into the tissue [82].

coordinates, given as 3DOF (x, y, θ), can be achieved using real time 2D US images, with image-processing algorithms such as: Kalman filter, needle enhancing filter, Hough transform, Harris corner detection, contrast stretching, intensity thresholding, adaptive thresholding based on Otsu’s method, morphological operations, fast labeling algorithm based on run length coding, template matching, similarity functions (e.g. MI, NCC, SSD, SCV), image moments and blob analysis [55], [83], [85], [89]-[93], [100].

Similarly 3D needle tip coordinates, given as 5DOF (x, y, z, θ, ϕ), can be tracked by keeping the calibrated 2D US transducer perpendicular to the direction of needle insertion and executing image processing algorithms such as Hough transform on respective real time 2D US images and therefore localizing the needle tip in these images [56]-[62], [64]. An implicit force control can be used to keep transducer orthogonal to the needle insertion axis over any curved surfaces [61].

Adebar et al. describes a recursive estimation approach based on UKF and nonlinear kinematic process model of needle steering to estimate the 6DOF pose, in $(x, y, z, \theta, \phi, \varphi)$, of the steerable needle tip in 3D US volume [94]. In studies conducted by Liang et al. the 6DOF pose of the needle tip is tracked in 3D US volume using simple image segmentation algorithms such as first-arrival thresholding [96]-[97]. In supplement, Hungr et al. makes the point that 6DOF tracking of the needle tip in 3D US volume can be optimized by improving the visibility through placing a rubber sleeve over the needle tip [98]-[99].

In a study the needle tip is tracked in 3D by fusing together the data from FBG-sensor with the data from 3D US volumes using UKF algorithm [63]. In this case, the FBG sensor helps track the needle tip when it is masked by anatomical structured [63]. On the other hand, the data from 3D US, obtained as the transducer is kept perpendicular to needle insertion direction, are processed using filtering, thresholding, contour tracing and shape matching using Fourier descriptors so that the needle tip can be localized in US volumes [63].

Stoll et al., Novotny et al. and Yuen et al. propose the tracking of rigid needle tip in 3D US volume, given as 6DOF $(x, y, z, \theta, \phi, \varphi)$, using a passive marker attached to the tip of the needle [66]-[67], [71]. Four out of the six DOFs, namely x, y, ϕ (pitch) and φ (yaw), are obtained by measuring the surgical instrument's shaft axis using Modified Radon Transform (a 3D line detection algorithm based on Singular Value Decomposition or SVD mathematical model) given a number of image points along the shaft and assuming that the rigid needle tip points away from the probe [66]-[67], [71]. The other two DOFs, namely z (tip position) and θ (roll angle of the needle tip), are obtained using the passive marker attached to the rigid needle tip [66]-[67], [71]. A vertical 2D slice in the 3D US volume through the centerline of the rigid needle gives the cross section image of the passive marker surface and ridges [66]-[67], [71]. By applying contour extraction using filtering and edge detection on this 2D slice, position (z) and orientation (θ) of the needle tip can be determined [66]-[67], [71].

Nadeau et al. on the other hand, reports tracking the tip of the concentric tube robot using 3D US visual servoing without requiring assumptions on object geometry or any segmentation steps [41]. This is accomplished solely by using the intensity information of the visual feature from raw 3D US data [41]. Kesner et al. also similarly tracks cardiac catheter tip position, employing in addition a GPU-based Radon transform algorithm to find the catheter's axis in real time [13], [70], [72].

Boctor et al. also propose tracking the ablator needle tip in 3D coordinates intra-operatively by collecting 2D US transverse images and compounding them into a 3D US volume using the slicer3D medical data visualization software package [73]-[76]. To compound the 2D US images into a US volume, slicer3D software requires the 6DOF pose of the US probe as it scans over the surface of interest. To this end, in this study, EM tracker is used to record the path of the probe during the scan [73]-[76]. In practical terms, the needle is inserted according to predetermined distance and real time 2D US is used as a monitoring tool while the slicer3D visualization software package synchronizes real time capture of 2D US data and the probe position information to assemble a spatially registered 3D volume which displays the needle with its current ablating range [73]-[76]. In another study by the same research group, Boctor et al., the tip of the catheter is tracked by embedding an active element (AE) near the tip of the catheter [77]. Specifically, the signal generated by the AE element under US wave effect is used for tracking the tip of the catheter [77].

In a study conducted by Chatelain et al. the needle tip is tracked in 3D US volume from the moment it is inserted without any prior knowledge on the insertion direction [78]. Using the information obtained from image differences, specifically local displacement of high intensity voxels, random sample consensus (RANSAC) algorithm with Kalman filtering (KF) in a closed loop is used to find the axis of the needle [78]. Kojcev et al. also employ the same technique but instead of KF it uses Extended KF (EKF) with RANSAC algorithm to find the axis of the needle [81]. Once the axis of the needle is found, the position of the tip along this axis is determined by looking for substantial intensity drop [78]. Consequently, robust real time tracking of the needle in 3D is achieved [78]. In another study by the same group, Chatelain et al., real time 3D needle tip tracking in 3D US volume is achieved by Particle Filtering (PF) [79]. In the employed PF algorithm, each particle represents a possible hypothesis of the location of the tip of the needle (modeled as polynomial curve) [79]. The measure of goodness of each particle/hypothesis is updated based on the data from 3D US using the Sequential Importance Re-sampling filter (SIR) algorithm which is designed to avoid the degeneracy phenomenon [79]. The degeneracy phenomenon is when the weight (i.e. probability) of one particle tends to 1 while the weight of others become negligible in which case it renders the recursive PF algorithm dysfunctional [79].

In a series of studies, Vitrani et al. discuss the 2D tracking of the forceps instrument tip in real time 2D US images [87]-[88]. Each Jaw of the forceps intersects the US plane, thus two blobs are visible [87]-[88]. The center of gravity of the two blobs (denoted P_1 and P_2 is calculated and the distance between P_1 and P_2 (denoted δ) is tracked in 2D US images along with θ which is the angle between the horizontal axis and the line through P_1 and P_2[87]-[88].

So far, the survey is this section only covers tracking of surgical tool tip. However, it is also important to mention that in some of the aforementioned papers, tracking of the target plays an essential role in the closed loops control governing the US-guided robotic surgery platform. Target tracking in 2D can be obtained from 2D US images obtained in real time by simple image processing techniques such as thresholding and calculating the centroid of the blob moments [55]. Active contour model can also be used to conduct a fast and robust segmentation for tumor motion in real time 2D US images [89]-[90]. Kaya et al. also report the use of NCC and MI similarity functions to track both small and moving targets (associated with affine motion model) and deformable targets (associated with thin plate spline motion model) in real time 2D US captures [93]. To track the target in 3D using 2D US images on the other hand, tracking of the 2D US probe during its scan is required [65]. In such case, the 3D target localization first involves calibration of US image plane with reference frame of the EM tracking unit [65]. Second, US images must be segmented using thresholding, boundary and texture identification to extract the target cross section [65]. Finally, registration is employed where 2D US frames are translated or rotated with respect to the EM tracking device base frame [65].

Nadeau et al., Zetting et al. and Kojcev et al. execute intraoperative tracking of both the surgical instrument and target at the same time in real time using 3D US visual servoing [41], [80]-[81]. Without any prior knowledge of target shape, the 3D US probe is controlled to keep the target centered within the probe's field of view (defined by 3D region of interest (ROI)) utilizing image intensity information as visual feature [41]. Novotny et al. and Yuen et al. also discuss intraoperative tracking of the target in 3D US volume by tracking an X marker attached to the surgical patch [67]-[69]. The X marker is two intersecting and perpendicular strands of nylon, where the intersection point of the two lines defines the target centroid position and the cross product defines its orientation [67]. In these studies, the modified Radon based algorithm is used to segment 3D US data [68]. Using the segmented 3D US data, target positions are determined which are passed to EKF algorithm that estimates target positions for 132 milliseconds in the future [68]. Similarly, in studies conducted by Kesner et al., to compensate for 50-100 milliseconds of delay in 3D US volume acquisition and processing, EKF estimates the current target (i.e. cardiac tissue) location based on Fourier decomposition of the cardiac cycle [13], [70], [72].

Yuen et al. introduce a real time tissue tracking technique in 3D US volume called Flashlight Tracker which tracks the tissue that the instrument is pointed toward [71]. First, a 2D image slice is constructed through the US volume that contains the shaft of the rigid needle [71]. K-means algorithm is used to group pixels that exceed intensity threshold, denoted by I_t, into two clusters based on Euclidean distance in which it is assumed that the more distal cluster is the target [71]. Next, active contour algorithm governed by real-time minimization of the energy equation through a greedy algorithm is used to segment the target [71]. Finally, the first intersection point of the contour with the ray along the instrument shaft is taken as the target position [71].

Boctor et al. also report the use of slicer3D medical data visualization package to track the target in 3D using 2D US scans [73]-[74], [76]. To use the slicer3D package for target localization in 3D, Hopkins calibration is used to determine the 6DOF transformation between the EM position sensor and the corner of the US image plane [73]. Using this transformation information and generic 3D US processing modules, slicer3D synchronizes real time 2D US data with EM tracked position information to robustly assemble a spatially registered 3D Ultrasound volume visualizing the target [73]-[74], [76].

Hungr et al. and Long et al. specifically discuss intraoperative prostate (i.e., the target in brachytherapy procedure) motion tracking in 3D using 3D TRUS data [98]-[99]. In these studies, a fast 3D US registration algorithm is used where the image of the prostate before implanting the brachytherapy seed is registered with the image of the prostate after implanting the brachytherapy seed and the amount by which the target (i.e. prostate) has moved is determined [98]-[99]. The registration algorithm is solely based on analysis of image intensity variations and uses a multistep pipeline, where each step refines the registration on increasingly more complex motion models [98]-[99].

2.7. Shape Estimation of Surgical Tools in US-guided RMIS

Estimating the shape of the surgical tool using US is also considered in the context of US-guided minimally invasive surgical robotics due to its applications in planning the trajectory and tracking control of surgical tools. The surgical tools under our study (e.g. flexible bevel-tip needles) deform as they are inserted into the tissue or surgical site, which must be taken into account in the whole control scheme. By updating the path planner and therefore the control system with real time shape estimation of these tools, better targeting accuracies can be achieved in US-guided surgery robotics [60], [62], [63], [78], [79], [83], [92], [95]-[97].

Moreira et al. implement on-line curvature estimation of the flexible needle in 3D by using 2D transverse US images [60]. To this end, the needle path is divided into sub-trajectories [60]. Each sub-trajectory is comprised of one-second duty cycling period of the needle insertion device, which is captured by thirty 2D US top-pose captures [60]. First, fast PCA algorithm based on SVD is used to find the plane of each sub-trajectory [60]. Second, circle-fitting algorithm estimates the radius of curvature in each sub-trajectory [60]. Finally, KF is used to reduce noise and estimate the maximum curvature of the flexible needle [60]. In progression of this work, this real-time flexible needle curvature estimation is fused with offline needle curvature estimation through indirect feed-forward KF [62]. Offline needle curvature estimation is based on biomechanics properties defining the relationship between tissue elasticity and the needle curvature [62]. In this study, tissue elasticity is expressed as Young's modulus, which is estimated by Acoustic Radiation Force Impulse Imaging (ARFI) [62]. On the other hand, the needle curvature is estimated by fitting a circle to each of the ten sets of 3D US needle tip position through 50 mm insertions of 6 different phantoms [62].

Shahriari et al. reconstruct the needle shape in 3D space and real-time using measurements from an array of FBG sensors fused with 3D US images using UKF [63]. In another study conducted by Chatelain et al., needle shape in 3D space is estimated using RANSAC algorithm where a polynomial curve is fitted to a candidate set obtained from 3D US captures in real-time [78]. In this study, a cost function is used to classify the voxels as belonging to the needle or to the background [78]. In another paper by Chatelain et al., SIR-based needle tracking algorithm is designed to track a curved needle in 3D US volume using estimation of insertion velocity [79]. Then, PF is used to detect the bending of the needle in 3D [79]. Liang et al. also reported 3D shape estimation of a needle using 3D US data [95]-[97]. The 3D needle shape is reconstructed in 3D voxel plot using 3D image segmentation algorithm based on first-arrival thresholding method [95]-[97].

Summing up, the aforementioned studies discuss shape estimation in 3D space using 2D/3D US only for flexible needles. 3D shape estimation of catheters or other continuum robots, a more challenging problem due to the hyper flexibility of catheters compared to needles, are overlooked in the context of US-guided surgical robotics platform. Nevertheless, the 3D shape estimation of continuum robots is important in developing techniques of closed-loop control, path planning, human-robot interaction and surgical manipulation safety concerns in minimally invasive surgeries that utilize them [20]. To fill this gap in literature, only a handful of papers are produced that attempt shape estimation of continuum robots in 3D space using the US system. Due to the importance and high relevance of this topic to the study at hand, these papers are summarized separately in the following section.

2.8. Ultrasound 3D Shape Estimation of Continuum Robots: A Survey

Continuum robots (i.e. robotic catheters, robotic instrument sheaths, snake-like robots, concentric tube robots), whose shape comprises a smooth curve along their length, are common in MIS such as endovascular intervention procedures (i.e. transcatheter intravascular US or IVUS diagnosis, transcatheter aortic calve implantation or TAVI, and catheter ablation) [53], [117]. An estimation of the shape of continuum robots in 3D space provides the surgeon with relative perspective of the surrounding vessels and tissues that surrounds the robot [53]. In practical terms, such 3D shape information can be used in tracking and servoing algorithms to prevent the formation of undesired loops, and damage to the vessel walls and tissues, while at the same time ensuring the stability of the continuum robot's contact especially when compensating for tissue motion or during surgical task automation [53], [117].

During MIS, it is desirable to provide navigational cues to the surgeon in the form of image overlays or virtual fixtures [117]. To this end, image-based tracking and servoing is implemented to perform procedures using medical continuum robots [117]. Ultrasound (US) imaging systems are preferably used for image-based tracking and servoing mainly due to their real time capability. In addition, its low cost, no radiation exposure, and good depth perception, makes clinical US widespread for interventional tasks [117].

The main disadvantages of US are low special resolution and contrast and high level of imaging artifacts [54]. Therefore, accurate continuum robot shape detection is of vital importance in the US image-guided minimally invasive interventions [69], [110], [118]-[120]. In addition, current tracking and servoing methods of interventional devices in the context of US-guided minimally invasive surgical robotics are very limited in their incorporation of shape estimation algorithms [13], [41], [51], [55]-[103]. Mostly the previous works either track/servo a single tip point [13], [55]-[59], [61],

[64], [66]-[67], [70]-[72], [74], [77], [81], [85]-[88], [91] or are arranged to only be applicable to needles (as opposed to long and flexible composition of tendon-driven catheters) [60], [62], [63], [78], [79], [83], [92], [95]-[97]. These methods of tracking and servoing are not applicable to steering flexible devices such as catheters, particularly in tortuous vessels [53].

Toward the goal of developing tracking and servoing algorithms for continuum robots, only a handful of papers have investigated estimation of curved robots in 3D space using US. These studies are summarized as follow.

Ren et al. present a shape estimation algorithm for continuum robots of known constant curvature given the limited field of view of US imaging where only the distal curved section of the robot appears [117]. In this study the shape estimation of the curved robot is done in 3D space using 3D US. The proposed shape estimation algorithm is based on circle parameter estimation [117]. A circle in Cartesian coordinates of a gray-level US volume, can be parameterized with six variables, $p = [x_0, y_0, z_0, \theta, \phi, R]$, where (x_0, y_0, z_0) correspond to the center of the circle, (θ, ϕ) are the angular parameters defining the unit normal to the circular plane $\mathbf{n} = (cos\theta cos\phi, sin\theta cos\phi, sin\phi)$, and R is the radius of the circle [117]. The shape estimation process is accomplished in three steps. First, a five-step pipeline image processing is used to remove the effects of imaging artifacts. The five-step pipeline imaging process is comprised of automatic thresholding, median filtering, connected component filter, morphological erosion and skeletonization [117]. Circle detection algorithm is then implemented on the skeletonized image in two parts. The non-unit length normal vector (**n**) to the plane containing the circle is first estimated using RANSAC algorithm [117]. Next, all points determined to lie in the estimated plane are projected onto a plane perpendicular to $\vec{n}$ for estimation of the radius and center of the planar circle [117].

In progression of the study explained in the last paragraph, Ren et al. develop a method called Tubular Enhanced Geodesic Active Contours (TEGAC) to estimate the shape of the continuum robot in 3D space using 3D US [54]. TEGAC is intended to replace the five-step pipeline image processing discussed in the previous study by Ren et al. This scheme combines geodesic active contours with a speed function to emphasize the tubular structure of the continuum robot while suppressing the other non-tubular structures in the 3D US image [54]. Non-tubular structures in 3D US volume can be known artifacts such as comet tail artifacts (CTA), diffractive side lobe (DSL) artifacts and range ambiguity artifacts (RAA) [54]. By incorporating the characteristics of these 3D US imaging artifacts and the robot tubular prior information (i.e., diameter of the robot's cross section and that the robot surface facing the transducer produces the clearest boundary), TEGAC detects the shape of the catheter in a cluttered environment [54]. In this scheme, first, the tubular enhancement module enhances the tubular structure of the continuum robot while suppressing the other non-tubular structures (i.e. artifacts) in the 3D US image [54]. This is accomplished using analysis of the eigensystem of the image volume's Hessian matrix, which reveals the geometrical dissimilarity of structures [54]. The results from the tubular enhancement module are used to derive a speed function that guides the active contour evolution, smoothing and pushing the contour to tubular boundaries [54].

Finally, Chen et al. propose shape estimation of cardiac catheters in 3D space with 2D US, using two-step PF [53]. In this study, a 2D US probe scans an endovascular catheter fixed in place in a water tank while an optical tracking device collects positional information of the US probe [53]. First, since catheter may be occluded due to US image artifacts, a multi-feature and multi-template particle filter algorithm is applied to US images for catheter tracking [53]. These catheter-tracking results are transformed into the optical tracking device coordinates. Second, to reduce the localization error due to low quality of US images and obtain a more accurate estimate of the 3D shape of the catheter, the motion of the catheter is modeled and a particle filter shape optimization algorithm is applied [53]. Finally, the 3D estimated shape of the catheter is overlaid on the preoperative cardiac 3D structures for an intuitive 3D visualization for use in minimally invasive cardiac surgeries [53].

The three aforementioned proposed methods of continuum robot shape estimation have been tested through experiments and validated against the ground truth as explained in the respective papers [53], [54], [117]. In Table 1, the average values of the error metrics reported in these papers are summarized. This table will act as an evaluation chart for the 3D shape estimation techniques that is developed for the purposes of this thesis.

Table 1

Comparison of US shape estimation errors among existing studies

Studies	Year	Method of Validation & The Ground Truth	US Modality	Error (mm)
H. Ren [117]	2011	Manual segmentation using the degree of volumetric overlap as computed with the DICE metric	3D US	2.3 ± 1.5
H. Ren [54]	2012	Manual segmentation using the degree of volumetric overlap as computed with the DICE metric	3D US	0.804 ± 0.015
F. Chen [53]	2017	CT scan image coordinates were mapped onto the tracking device coordinates	2D US	2.23 ± 0.87

2.9. Summary

In this chapter, the possible constituents of the US-guided flexible RMIS platform that was of specific interest to the literature was reviewed in sections 2.1-2.7 (i.e., surgical instruments, US imaging modalities, robot hardware specifications, control algorithms, path/trajectory planning, visual tracking methods, and shape estimation of surgical tools). In the body of the reviewed literature major limitations were noted as follow. Firstly, many of these publications did not implement the preferred visual servoing method of IBVS to solve their tracking problem and instead sufficed their experimental set up to PBVS [13], [55]-[65], [70], [72]-[73], [76]-[80], [83]-[84], [92]-[93], [95]-[97]. PBVS uses observed features, a calibrated camera and a known geometric model of the target to determine the pose of the target with respect to the camera. The robot then moves toward that pose and the control is performed in the 3D task space. Therefore, PBVS is not only computationally expensive but also relies critically on the accuracy of the camera calibration and the model of the object's geometry [113]. On the other hand, IBVS omits the pose estimation step and uses the image features directly. The control is performed in 2D image coordinate space and the desired camera pose with respect to the target is defined implicitly by the image feature values at the goal pose [133]. The aforementioned IBVS specifications makes it computationally less expensive and more accurate. Secondly, many publications fail to account for involuntary motions inherent in MIS such as respiration and heartbeat in their control mechanism. In other words, there is a lack of real time tracking of the target or obstacles in literature [13], [41], [51], [56]-[66], [69]-[70], [72]-[97]. Finally, and most relevant to the purpose of this thesis is the lack of real-time 3D shape estimation of the surgical tools in the reviewed literature [13], [41], [51], [55]-[72], [74]-[78], [80]-[81], [84]-[103]. This not only forsakes the effect of involuntary motions such as heart beat and respiration on the shape of the flexible surgical instrument but also affects the accuracy and efficiency of the control systems implemented to solve the targeting problems addressed in these studies.

Aside from the literature concerning US-guided flexible robotic minimally invasive surgical platforms, section 2.8 specifically reviewed papers that separately attempted 3D shape estimation of tendon-driven catheter using US imaging systems. The total of these publications amounts to only three papers dated from 2011 – 2017. Two of these papers ([54], [117]) estimate the 3D shape of the catheter using 3D US systems. Compared to conventional 2D US systems, 3D US systems suffer from poor accuracy (~1mm), low sampling frequency, and slow acquisition and processing of 3D US volumes [68]-[69]. On the other hand, the latest of these papers, [53], estimates the 3D shape of the catheter using 2D US system but with a computationally expensive particle filter (PF) algorithm. Overall the small number of publications devoted to the estimation of the 3D shape of the tendon-driven catheters using US imaging system is indicative of potential for further research in this arena. In addition, among the three mentioned published papers, none provides contribution applicable to real-time applications. This further justifies the motivation of research under the topic of real-time 3D shape estimation of tendon-driven catheters using US imaging system.

Summing up, the title of this thesis is motivated by the lack of contributions in research pertaining to 3D shape estimation of tendon-driven catheters using US imaging system. Even though, the real-time functionality of the methods studied in this thesis have not been implemented or validated, the predictive nature of the chosen tracking algorithms (i.e., KF, AKF, and PF), which only depends on the present measured state, makes it possible to extend their development to real-time applications.

Chapter III

THEORY: ALGORITHMS & SIMULATIONS

The catheter which is fixed inside a gel-filled container is scanned by the US probe transversely across its length. Using a sequence of US transverse images as input, the primary purpose of catheter 3D shape estimation algorithms and simulations is to track the cross section of the catheter within these images. These tracked cross sections, assembled in a correct configuration in a global coordinate frame, can provide an estimate of the 3D shape of the catheter.

In this thesis temporal probability models based on Bayesian statistics are deployed to track catheter cross sections in US frames. Bayes' theorem is a fundamental theorem in Bayesian statistics as it is used to update probabilities and estimates in the evidence of new data[19]. Specifically, Kalman filter (KF) and adaptive Kalman filter algorithms (AKF) are used to track the centers of the catheter cross sections while particle filter (PF) algorithm is used to track the contour of these cross sections in US frames. These three algorithms are chosen to solve the problem of 3D catheter shape estimation because of their applicability to real-time applications. In other words, KF, AKF and PF can predict the future location of the catheter cross section, which is to be captured by the US probe in the next US image, only based on the current location of the catheter cross section in the current US image. The real time tracking of the shape of the catheter is valuable especially in real time robotic MIS. KF, AKF and PF tracking algorithms are explained in general terms in the first three sections of this chapter, respectively.

Other than KF, AKF and PF, another method under the umbrella of Bayesian statistics using temporal probability models exist that even though not applicable to the problem of this thesis is worth mentioning, namely, Extended Kalman Filter (EKF). EKF is used for nonlinear system models exhibiting nonlinear dynamics which is absent in the problem studied in this thesis. More

[19] Given two events A and B, the conditional probability of A given that B is true is expressed as follows: $P(A|B) = P(B|A)P(A)/\ P(B)$

specifically, EKF is a heuristic solution to the nonlinear tracking problem [122] and may diverge from the true solution if applied to the problem of this thesis.

To validate their applicability to the problem of catheter 3D shape estimation, each of these three algorithms were applied to a series of MATLAB originated US images simulating transverse scanning across the length of the catheter. The known 3D shape of the catheter, which was deliberately drawn into these MATLAB synthesized US frames, serves as the basis of validation (or ground truth) when it is compared with the 3D shape of the catheter estimated by each of the KF, AKF and PF algorithms. The generation of this demo US scan is discussed in section 4 of this chapter and the implementations of KF, AKF and PF on it are presented in sections 5, 6 and 7, respectively. In addition, an evaluation of the accuracy for the catheter 3D shape estimate via each algorithm is provided in section 8 along with the CPU time each algorithm takes to produce the estimates. Finally, in section 9, the results are discussed, specifically KF, AKF and PF are evaluated against each other in terms of their speed and accuracy.

3.1. Kalman Filter Algorithm

Kalman filter provides mathematical framework for inferring the unmeasured variables from indirect and noisy measurements [121]. It is also used for predicting the likely future courses of dynamic systems [121]. Especially in computer vision tracking applications, it can be used to cope with non-rigid deformations of the object, background clutter, blurring, and occasional occlusion in the images [122]. Therefore, KF algorithm is applicable to the problem of tracking the catheter cross section across a series of US frames. In this section the catheter 3D shape estimation algorithm based on KF tracking of its cross sections across a series of US frames is presented.

To define the KF, temporal and measurement models must be specified [122]. First, based on Markov assumption[20], the temporal model relates the world states, $\{\mathbf{w}\}_{t=1}^{T}$, at times $t-1$ and t, and is given by equation 3.1 [122]. In this section, the world states, $\mathbf{w}_{t-1}$ and $\mathbf{w}_t$, refer to the estimated state of the catheter cross section when the US probe is scanning across it at times $t-1$ and t.

[20] Markov assumption assumes that each state depends only upon its predecessor [122].

Therefore, the world state is composed of the US image pixel position coordinates (u, v) of the catheter cross-section along with time rate of these coordinates (from one US frame to the next) represented by $(\dot{u}, \dot{v})$: $\mathbf{w} = [u \quad v \quad \dot{u} \quad \dot{v}]^T$.

$$\mathbf{w}_t = \boldsymbol{\mu}_p + \boldsymbol{\Psi}\mathbf{w}_{t-1} + \boldsymbol{\epsilon}_p \tag{3.1}$$

The term $\boldsymbol{\epsilon}_p$ is a realization of the transition noise, which is normally distributed with covariance $\boldsymbol{\Sigma}_p$ [122].

Alternatively, we can write equation 3.1 in probabilistic form [122]:

$$Pr(\mathbf{w}_t|\mathbf{w}_{t-1}) = Norm_{\mathbf{w}_t}[\boldsymbol{\mu}_p + \boldsymbol{\Psi}\mathbf{w}_{t-1}, \boldsymbol{\Sigma}_p]$$
(3.2)

$\boldsymbol{\mu}_p$ is a 4×1 $(D_{\mathbf{w}} \times 1)$[21] vector representing the mean change in the state. In other words, in this thesis KF assumes that variables $(u, v, \dot{u}, \dot{v})$ are each normally distributed random variable having a mean value, which for the purposes of this section can be best represented by the center of the catheter cross-section. Therefore, $\boldsymbol{\mu}_p = [u_c \quad v_c \quad \dot{u}_c \quad \dot{v}_c]^T$where (u_c, v_c) are the US image pixel coordinates of the position of the center of the catheter cross sections and $(\dot{u}_c, \dot{v}_c)$ are the rate of positional change of the catheter cross-section center from one US image to the next. $\boldsymbol{\Psi}$ is a 4×4 $(D_{\mathbf{w}} \times D_{\mathbf{w}})$ transition matrix relating the mean of the state at time t to the state at time $t-1$, and Σ_p is also a constant 4×4 $(D_{\mathbf{w}} \times D_{\mathbf{w}})$ transition noise matrix which determines how closely related the states are at times t and $t-1$ [122].

Second, the measurement model relates the noisy measurement data $\mathbf{x}_t$ at time t to the state $\mathbf{w}_t$ as given in equation 3.3 [122]. In this section, the measured state, x_t, refers to the catheter cross section as segmented in the US transverse image at time t. Therefore, the measurement state consists of only US image position coordinates (u, v):

[21] $D_{\mathbf{w}}$ refers to the larger dimension of the world state $\mathbf{w}$. $\mathbf{w}$ being a 4×1 vector makes $D_{\mathbf{w}} = 4$.

$$\mathbf{x} = \begin{bmatrix} u \\ v \end{bmatrix},$$

$$\mathbf{x}_t = \boldsymbol{\mu}_m + \boldsymbol{\Phi}\mathbf{w}_t + \boldsymbol{\epsilon}_m \tag{3.3}$$

The term $\boldsymbol{\epsilon}_m$ is a realization of the measurement noise, which is normally distributed with covariance $\boldsymbol{\Sigma}_m$ [122].

In probabilistic notation equation 3.3 can be written as [122]:

$$\Pr(\mathbf{x}_t|\mathbf{w}_t) = Norm_{\mathbf{x}_t}[\boldsymbol{\mu}_m + \boldsymbol{\Phi}\mathbf{w}_t, \boldsymbol{\Sigma}_m] \tag{3.4}$$

$\boldsymbol{\mu}_m$ is a 2×1 $(D_{\mathbf{x}} \times 1)$[22] mean vector. Again, in this thesis, it is assumed that variables (u, v) are each normally distributed random variable having a mean value, which for the purposes of this section can be best represented by the center of the US segmented catheter cross-section. Therefore, $\boldsymbol{\mu}_m = \begin{bmatrix} u_c \\ v_c \end{bmatrix}$, where (u_c, v_c) are the US image pixel coordinates of the position of the center of the segmented catheter cross sections. $\boldsymbol{\Phi}$ is a 2×4 $(D_{\mathbf{x}} \times D_{\mathbf{w}})$ matrix relating the 2×1 $(D_{\mathbf{x}} \times 1)$ measurement vector to 4×1 $(D_{\mathbf{w}} \times 1)$ state [122], and $\boldsymbol{\Sigma}_m$ is the 2×2 measurement noise matrix defining additional uncertainty on the measurements that cannot be described by the state [122].

In textbook terms [122], KF algorithm is a set of rules for computing the marginal posterior probability $\Pr(\mathbf{w}_t|\mathbf{x}_{1\ldots t})$ given the marginal posterior probability at the previous time $\Pr(\mathbf{w}_{1\ldots t-1}|\mathbf{x}_{1\ldots t-1})$ and a new measurement $\mathbf{x}_t$. The marginal posterior probability $\Pr(\mathbf{w}_{1\ldots t-1}|\mathbf{x}_{1\ldots t-1})$ is normally distributed with mean of $\boldsymbol{\mu}_{t-1}$ and variance of $\boldsymbol{\Sigma}_{t-1}$. Finally, the KF algorithm is given below.

Algorithm 3.1: The Generic Kalman Filter Algorithm [122]

Input: Measurements $\{\mathbf{x}\}_{t=1}^{T}$, temporal parameters $\boldsymbol{\mu}_p, \boldsymbol{\Psi}$, $\boldsymbol{\Sigma}_p$, measurement parameters $\boldsymbol{\mu}_m, \boldsymbol{\Phi}$, $\boldsymbol{\Sigma}_m$

[22] $D_{\mathbf{x}}$ refers to the larger dimension of the measurement state $\mathbf{x}$. $\mathbf{x}$ being a 2×1 vector makes $D_{\mathbf{x}} = 2$.

```
Output: Means $\{\boldsymbol{\mu}_t\}_{t=1}^{T}$ and covariances $\{\boldsymbol{\Sigma}_t\}_{t=1}^{T}$ of marginal posterior distributions

Begin
    // Initialize mean and covariance
    $\boldsymbol{\mu}_0 = 0$
    $\boldsymbol{\Sigma}_0 = \boldsymbol{\Sigma}_0$        // Typically set to a large multiple of identity
    // For each time step
    for $t = 1$ to $T$ do
        // State estimation
        $\boldsymbol{\mu}_+ = \boldsymbol{\mu}_p + \boldsymbol{\Psi}\boldsymbol{\mu}_{t-1}$
        // Covariance estimation
        $\boldsymbol{\Sigma}_+ = \boldsymbol{\Sigma}_p + \boldsymbol{\Psi}\boldsymbol{\Sigma}_{t-1}\boldsymbol{\Psi}^{\mathrm{T}}$
        // Compute Kalman gain
        $\mathbf{K} = \boldsymbol{\Sigma}_+\boldsymbol{\Phi}^{\mathrm{T}}(\boldsymbol{\Sigma}_m + \boldsymbol{\Phi}\boldsymbol{\Sigma}_+\boldsymbol{\Phi}^{\mathrm{T}})^{-1}$
        // State update
        $\boldsymbol{\mu}_t = \boldsymbol{\mu}_+ + \mathbf{K}(\mathbf{x}_{\mathrm{t}} - \boldsymbol{\mu}_m - \boldsymbol{\Phi}\boldsymbol{\mu}_+)$
        // Covariance update
        $\boldsymbol{\Sigma}_t = (\mathbf{I} - \mathbf{K}\boldsymbol{\Phi})\boldsymbol{\Sigma}_+$
    end
end
```

The choice of temporal model in the Kalman filter is restricted to be linear and is dictated by the transition matrix $\boldsymbol{\Psi}$ [122]. For the purposes of this project, extended Brownian motion is used as the temporal model:

$$\begin{aligned} u_t &= u_{t-1} + \Delta t \dot{u}_{t-1} \\ \dot{u}_t &= \qquad\quad \dot{u}_{t-1} \\ v_t &= v_{t-1} + \Delta t \dot{v}_{t-1} \\ \dot{v}_t &= \qquad\quad \dot{v}_{t-1} \end{aligned} \tag{3.5}$$

Therefore, the transition matrix can be derived to be:

$$\mathbf{\Psi}_t = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.6}$$

We can write equations 3.5 in matrix form:

$$\begin{bmatrix} u_t \\ v_t \\ \dot{u}_t \\ \dot{v}_t \end{bmatrix} = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_{t-1} \\ v_{t-1} \\ \dot{u}_{t-1} \\ \dot{v}_{t-1} \end{bmatrix} \tag{3.7}$$

Equation 3.7 corresponds to equation 3.8 as used in the algorithm 3.1 for state estimation:

$$\boldsymbol{\mu}_{+} = \mathbf{\Psi}\boldsymbol{\mu}_{t-1} \tag{3.8}$$

Additive Gaussian white noise is a common assumption in KF algorithm. Kalman filters inherently make the assumption that all noise processes are zero mean Gaussian, which often is a good assumption even when it is strictly not true. This is because as a result of assuming white Gaussian noise, it is a filter that does not amplify high frequency noise in its predictions. Another assumption in KF algorithm for optimal estimation is the choice of Brownian motion model to represent the problem of tracking the catheter cross section in a sequence of US transverse images. This model is appropriate given the fast US frame acquisition at 25-33 Hz through which the catheter cross section is assumed to be displaced in any random direction. Brownian motion model is a mathematical model used to describe such random movements through which the underlying dynamics of the system does not change over time, so that the distribution of the possible displacements of the catheter cross section in a time step only depends on the length of timestep Δt. In other words, the underlying dynamics of the Brownian motion model is a random walk defined by a sequence of normally distributed (i.e., Gaussian) independent variables denoting the process by which randomly moving catheter cross section wanders away from where it started at the point of insertion. Random walk requires miniscule time steps which is met given the fact that US frames are acquired at around $\Delta t = 0.04\ s$. The normal distribution is justified to model the random variables in this model considering spatial homogeneity in which it is assumed that the catheter cross section is no more or less likely to be jostled to the right than to the left.

On the other hand, in the measurement model, the 2×4 matrix $\mathbf{\Phi}$ relates the measurements to the state predictions. Given the measurement being a 2×1 vector, $\mathbf{x} = \begin{bmatrix} u \\ v \end{bmatrix}$ and the state prediction being a 4×1 vector, $\mathbf{w} = [u \quad v \quad \dot{u} \quad \dot{v}]^T$, Φ is chosen to be a 2×4 matrix given as below to make the relation between **x** and **w** mathematically possible since the multiplication of 2×4 matrix with 4×1 vector is a 2×1 vector.

$$\mathbf{\Phi} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{3.9}$$

Next, in the absence of prior information at time $t = 1$, the prior mean $\boldsymbol{\mu}_0$ is initialized to a reasonable value according to the specific problem at hand. For the problem of catheter 3D shape estimation, $\boldsymbol{\mu}_0 = [u_0 \quad v_0 \quad \dot{u}_0 \quad \dot{v}_0]^T$ is initialized to the catheter cross-section center localized at the very first transverse US frame in the sequence. This initialization is reasonable since it is assumed that one always knows the insertion point coordinates. In addition, the prior covariance, $\mathbf{\Sigma}_0$, is set to a large multiple of identity matrix indicating our lack of knowledge at the start of the algorithm.

As the KF algorithm runs recursively through US frames from $t = 1 \; to \; T$, the center of the catheter cross-section is tracked in: $\boldsymbol{\mu}_+ = [u \quad v \quad \dot{u} \quad \dot{v}]^T$.

In addition, the covariance matrix,

$$\mathbf{\Sigma}_+ = \begin{bmatrix} \mathit{\Sigma}_{uu} & \mathit{\Sigma}_{uv} & \mathit{\Sigma}_{u\dot{u}} & \mathit{\Sigma}_{u\dot{v}} \\ \mathit{\Sigma}_{vu} & \mathit{\Sigma}_{vv} & \mathit{\Sigma}_{v\dot{u}} & \mathit{\Sigma}_{v\dot{v}} \\ \mathit{\Sigma}_{\dot{u}u} & \mathit{\Sigma}_{\dot{u}v} & \mathit{\Sigma}_{\dot{u}\dot{u}} & \mathit{\Sigma}_{\dot{u}\dot{v}} \\ \mathit{\Sigma}_{\dot{v}u} & \mathit{\Sigma}_{\dot{v}v} & \mathit{\Sigma}_{\dot{v}\dot{u}} & \mathit{\Sigma}_{\dot{v}\dot{v}} \end{bmatrix},$$

keeps track of the correlation between each of the normally distributed random variables $(u, v, \dot{u}, \dot{v})$, where each element of the matrix $\mathbf{\Sigma}_{+_{ij}}$ is the degree of correlation between the ith state variable and the jth state variable.

The Kalman gain, denoted by $\mathbf{K}$ in the algorithm 3.1, plays an important role in incorporating the measurement data into the algorithm. A small Kalman gain implies that the new measurement is unreliable relative to the marginal posterior and therefore should be weighted less in the update of the state estimation [122]. A large Kalman gain on the other hand implies that the measurement is more reliable than the prior and should be weighted more highly [122].

3.2. Adaptive Kalman Filter Algorithm

Similar to KF, AKF is also used in this thesis to track the centers of the catheter cross section in US transverse images. AKF is analogous in algorithm to KF with the exception that the process noise (also called the dynamic noise covariance matrix $\boldsymbol{\Sigma}_p$) is set to be constant throughout KF algorithm but in AKF it is continuously updated. If the constant defining $\boldsymbol{\Sigma}_p$ is a poor estimation of the dynamic noise, large errors and divergence can occur in tracking[23] [123]. By continuously updating $\boldsymbol{\Sigma}_p$, AKF achieves a more robust tracking of the catheter cross section [123]. To elaborate further, $\boldsymbol{\Sigma}_p$ represents the level of confidence that exists in the dynamic model's ability to accurately estimate the actual trajectory [123]. However, without prior knowledge of the catheter 3D shape, it is not possible to pre-select $\boldsymbol{\Sigma}_p$ to be some constant matrix. In addition, $\boldsymbol{\Sigma}_p$ may have varying statistics throughout the catheter length. Therefore, selecting a constant $\boldsymbol{\Sigma}_p$, offline, may not be satisfactory. In such a case, $\boldsymbol{\Sigma}_p$ is better modeled as varying with time and AKF allows $\boldsymbol{\Sigma}_p$ to be updated every sampling period based on past data [123].

A maximum likelihood method is used to update $\boldsymbol{\Sigma}_p$ at each iterative step, k, as follow [123]:

$$\boldsymbol{\Sigma}_p(k) = \left(\frac{1}{k-1}\right) \sum_{i=k-N+1}^{k} \left((\mathbf{q}_i - \overline{\mathbf{q}})(\mathbf{q}_i - \overline{\mathbf{q}})^T - \frac{N-1}{N} [\boldsymbol{\Psi}\boldsymbol{\Sigma}_{i-1}\boldsymbol{\Psi}^{\mathrm{T}} - \boldsymbol{\Sigma}_i] \right), \tag{3.10}$$

[24]

[23] Other Kalman filter assumptions that may contribute to low accuracy are the inherent constant dynamics and zero mean Gaussian noise model which do not hold in real world processes.

[24] The T in this equation refers to the matrix transpose mathematical operation.

where $\mathbf{\Sigma}_{i-1}$ is the estimated state error covariance matrix calculated in the KF loop notated as $\mathbf{\Sigma}_{+}$ (algorithm 3.1) at $(i-1)^{th}$ summation step and N is the number of past measurements chosen to base the process noise calculation on. In other words, the first N state and covariance estimates calculated by the filter are decided to not be accurate and so the update on process noise will therefore start after the N samples.

The adaptation rule obviously adds to the computation load. Using limited memory filter algorithm the number of numerical computations required by equation 3.10 can be reduced by using the following equation instead [123]:

$$\mathbf{\Sigma}_p(k) = \mathbf{\Sigma}_p(k-1) + \left(\frac{1}{N-1}\right)(\mathbf{q}_k - \overline{\mathbf{q}}_k)(\mathbf{q}_k - \overline{\mathbf{q}}_k)^T$$

$$-\left(\frac{1}{N-1}\right)(\mathbf{q}_{k-N} - \overline{\mathbf{q}}_k)(\mathbf{q}_{k-N} - \overline{\mathbf{q}}_k)^T$$

$$+\left(\frac{1}{N^2-N}\right)(\mathbf{q}_k - \mathbf{q}_{k-N})(\mathbf{q}_k - \mathbf{q}_{k-N})^T$$

$$+\left(\frac{1}{N}\right)(\mathbf{\Delta}_{k-N} - \mathbf{\Delta}_k), \tag{3.11}$$

where $\mathbf{q}_k, \overline{\mathbf{q}}_k, and\ \mathbf{\Delta}_k$ in equation 3.11 are given by [123]

$$\mathbf{q}_k = \boldsymbol{\mu}_k + \mathbf{\Psi}\boldsymbol{\mu}_{k-1}, \tag{3.12}$$

$$\overline{\mathbf{q}}_k = \overline{\mathbf{q}}_{k-1} + \left(\frac{1}{N}\right)(\mathbf{q}_k - \mathbf{q}_{k-N}), \tag{3.13}$$

$$\mathbf{\Delta}_t = \mathbf{\Psi}\mathbf{\Sigma}_{t-1}\mathbf{\Psi}^T - \mathbf{\Sigma}_t. \tag{3.14}$$

In equation 3.13, when $k-1 = N, \overline{\mathbf{q}}_k$ can be calculated through the following equation:

$$\overline{\mathbf{q}}_{(k-1=N)} = \left(\frac{1}{N}\right)\sum_{i=k-N+1}^{k} \mathbf{q}_i, \tag{3.15}$$

where $\mathbf{q}_i$ is given by $\mathbf{q}_i = \boldsymbol{\mu}_k + \boldsymbol{\Psi}\boldsymbol{\mu}_{k-1}$ from the first N iterations of the filter algorithm.

3.3. Particle Filter Algorithm

KF and AKF only deal with linear temporal and measurement models, which could be valid in low dynamics operations or those with high sampling rates [122]. In addition, they represent uncertainty over the state as a normal distribution and are therefore not suitable in settings where the probability over the state is multimodal. PF addresses this limitation by representing the probability density as a set of particles in the state space where each particle can be thought of as representing a hypothesis about the possible state. The particles can project through time to simulate measurements no matter how nonlinear the functions are and therefore since the estimated state is multi-modal, the measurement density may be multi-modal as well. This means that the PF copes much better with clutter in the scene, which is characteristic challenge of tracking in US images. Of course, this comes with the cost of significantly higher computational load, which may not be appropriate for some real-time operations. Therefore as long as some of the predicted state (i.e. some of the particles) agrees with the measurement density, the tracker should remain stable [122]. In this thesis PF is used to track the contour of the catheter cross-section as visible across the US transverse images. Among PF methods, condensation algorithm was used to accomplish this tracking task.

In condensation algorithm, the probability distribution can be found by a weighted sum of J weighted particles [122]:

$$\Pr(\mathbf{w}_{t-1}|\mathbf{x}_{1\ldots t-1}) = \sum_{j=1}^{J} a_j \delta[\mathbf{w}_{t-1} - \widehat{\mathbf{w}}_{t-1}^{[j]}], \tag{3.16}$$

where the weights represented by a_j are positive and sum up to 1. Each particle represents a possible estimation of the state, and the weight of the particle indicates our confidence in that hypothesis. The goal of the condensation algorithm is to calculate the probability distribution $\Pr(\mathbf{w}_t|\mathbf{x}_{1\ldots t})$ at the next time step. The condensation algorithm is given below [122].

Algorithm 3.2: The Generic Condensation Algorithm [122]

```
Input: Measurements $\{\mathbf{x}\}_{t=1}^{T}$, temporal model $\Pr(\mathbf{w}_t|\mathbf{w}_{t-1})$, measurement model $\Pr(\mathbf{x}_t|\mathbf{w}_t)$
Output: weights $\left\{a_t^{[j]}\right\}_{t=1}^{T}$, hypothesis $\left\{\mathbf{w}_t^{[j]}\right\}_{t=1}^{T}$

Begin
    // Initialize weights to equal
    $a_0 = [\frac{1}{J}, \frac{1}{J}, \dots, \frac{1}{J}]$
    // Initialize hypothesis to plausible values for state
    for $j = 1$ to $J$ do
        $\mathbf{w}_0^{[j]} = Initialize[]$
    end
    // For each time step
    for $t = 1$ to $T$ do
        // For each particle
        for $j = 1$ to $J$ do
            // Sample from 1 ... $J$ according to probabilities $a_{t-1}^{[1]}$ ... $a_{t-1}^{[j]}$
            $n = sampleFromCategorial[a_{t-1}]$
            // Draw sample from temporal update model
            $\hat{\mathbf{w}}_t^{[j]} = sample[\Pr\left(\mathbf{w}_t \middle| \mathbf{w}_{t-1} = \hat{\mathbf{w}}_{t-1}^{[n]}\right)]$
            // Set weight for particle according to measurement model
            $a_t^{[j]} = \Pr\left(\mathbf{x}_t \middle| \hat{\mathbf{w}}_t^{[j]}\right)]$
        end
        // Normalize weights
        $a_t = \frac{a_t}{\sum_{j=1}^{J} a_t^{[j]}}$
    end
end
```

Assuming that the point of insertion of the catheter is always known (i.e. the catheter cross section centroid location in the very first US image is known), random particles about this point are defined with equal weight of goodness assigned to each.

In the condensation algorithm implemented in this project, the temporal model is defined by KF dynamics equations as derived in the previous sections. This is due to the fact that given the random nature of the problem at hand, where the next catheter cross section can be anywhere (similar to a random walk motion model), the best mechanism to predict the next state of each particle (i.e., hypothesis) is by KF based on statistical observation of the data. In other words, given J hypothesis (i.e. particles) at time $t-1$, $\{\mathbf{w}_{t-1}^{[1]}, \mathbf{w}_{t-1}^{[2]} \dots \mathbf{w}_{t-1}^{[J]}\}$, with associated probabilities, $\{P\left(\mathbf{w}_{t-1}^{[1]}\right), P\left(\mathbf{w}_{t-1}^{[2]}\right) \dots\ P\left(\mathbf{w}_{t-1}^{[J]}\right)\}$, KF dynamic equations can be used to estimate the next state for each particle, $\{\mathbf{w}_{t}^{[1]}, \mathbf{w}_{t}^{[2]} \dots\ \mathbf{w}_{t}^{[J]}\}$. At this point, a new set of weights, $\{a_{t}^{[1]}, a_{t}^{[2]} \dots a_{t}^{[J]}\}$, for each particle is allocated, based on the measurement model which represents a measure of correctness of each particle's state.

The measurement model consists of catheter cross sections segmented across the US images by thresholding the ROI encompassing the most particles. The goal of tracking with condensation algorithm is to have particles track the contour of the catheter cross section across the sequence of US images. For this purpose, edge detection can be applied to the thresholded binary images of the catheter cross-section and the average distance from the centroid to the detected edge can be extracted. In addition, the distance of each particle from the catheter cross-section centroid is calculated. The weight of each particle is then assigned by comparing the distance of each particle from catheter cross-section centroid against the average distance from the catheter cross-section centroid to its detected edge. The closer the particle is to the edge, the higher its weight would be.

A number of particles with highest weights are sampled and updated (i.e. corrected) using KF gain function (as outlined in algorithm 3.1). A fast non-iterative ellipse [124] is fitted through these sampled and updated particles. Such an ellipse serves as estimated representation of the catheter's cross section contour captured by US images. Finally out of the sampled and updated particles, a new series of particles are produced for the next round of iteration. Again, all the particles are set to have an equal weight for the next round to start with.

3.4. B-Mode Ultrasound Validation Simulation

Before experimental validation of the aforementioned temporal probability algorithms (i.e., KF, AKF, and PF) as the solution to the problem of catheter 3D shape estimation, MATLAB simulation validations are presented in this section. To this end three different catheter 3D configurations are programmed into a sequence of MATLAB generated US images. These catheter configurations are illustrated in Figures 6, 7, and 8.

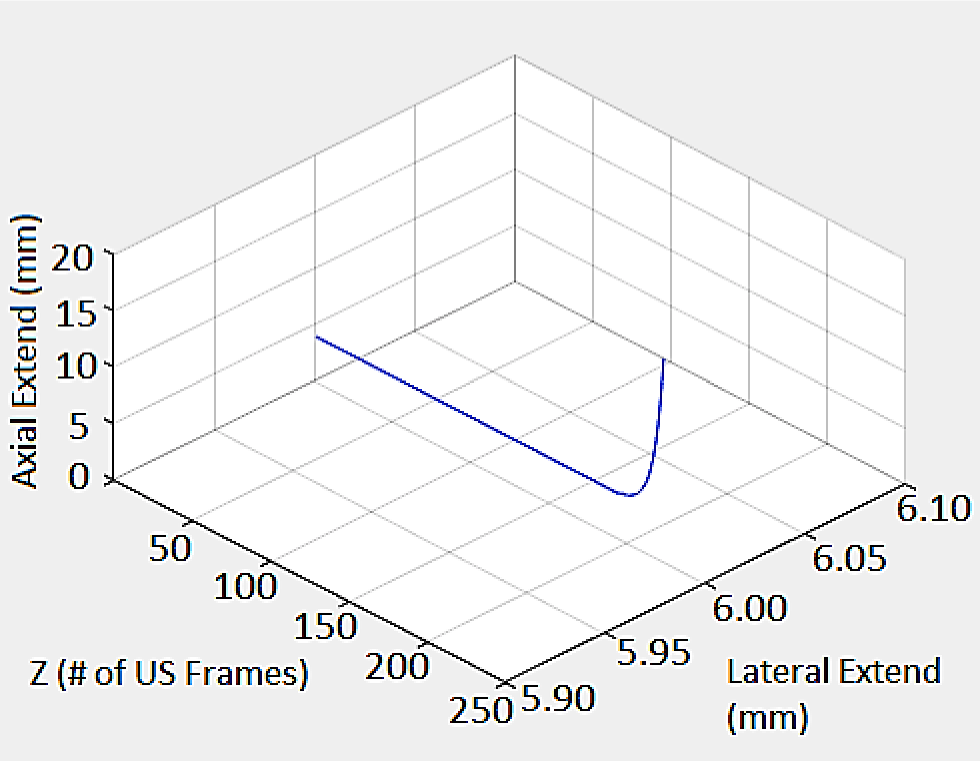


Figure 6 – Catheter 3D configuration # 1 for validation simulations

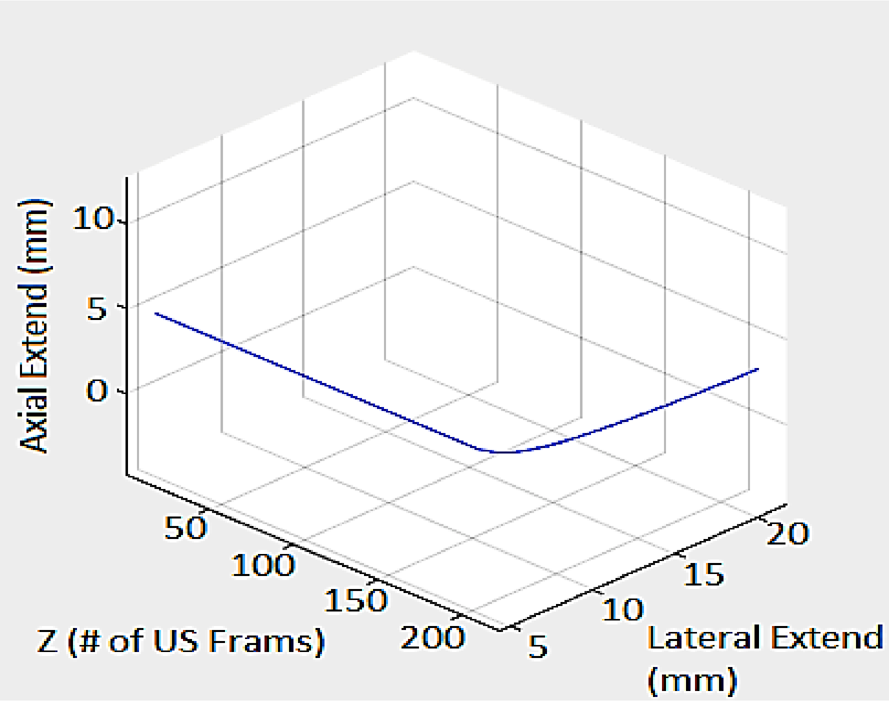


Figure 7 – Catheter 3D configuration # 2 for validation simulations

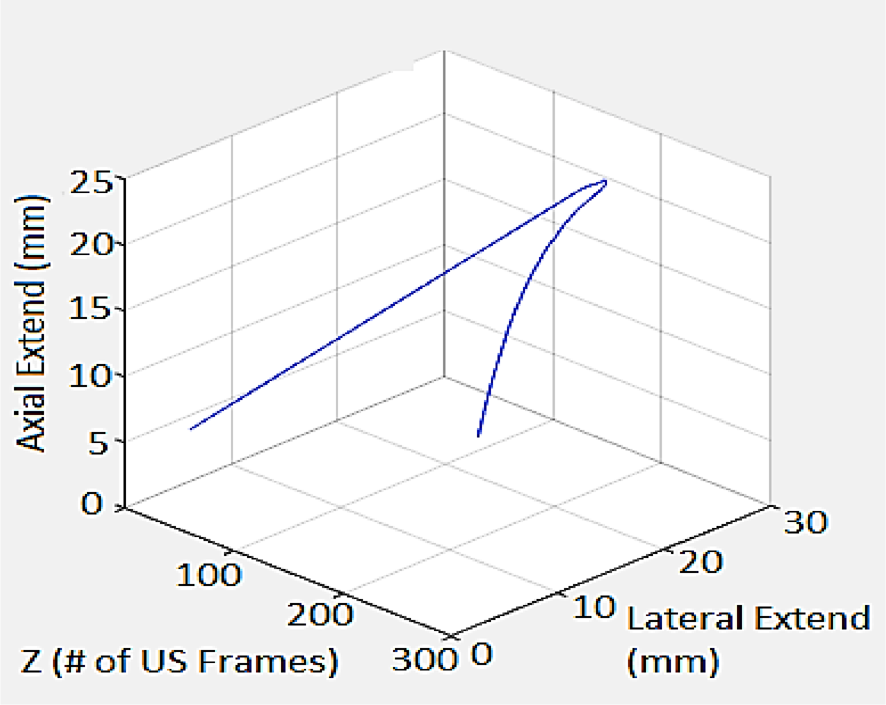


Figure 8 – Catheter 3D configuration # 3 for validation simulations

The catheter is known to have circular and consistent cross-sections along its length (e.g., Std Crv Blazer II HTD® with cross section diameter of 1 mm); therefore, the 3D catheter configurations in Figures 6, 7 and 8 only show the longitudinal axis through the center of the catheter cross sections for simplicity.

Simulating US scanning over and across the catheters depicted in Figures 6, 7 and 8, transverse and longitudinal images can be originated in MATLAB. To validate the KF, AKF and PF validation algorithms, transverse images will be used.

To be more compliant with the nature of US images, speckle and Gaussian noises should be added to distort or corrupt the simulated transverse images, making it more challenging for the KF, AKF and PF algorithms to estimate the 3D shape of the catheters. This provides a better assessment of the robustness of the proposed algorithms.

Realistic B-mode images can be simulated with scattering maps based on optical, CT, or MR images or parametric flow models [125]. Field II simulation program can be used to create realistic B-mode images [126]. The image simulation in Field II package includes 100,000-point scatterers [126]. Field II generates US speckle noise pattern as the signal from these 100,000 randomly placed point scatterers with a Gaussian amplitude [125]. In addition, the generation of US image is accomplished using linear acoustics where spatial impulse response is simulated for 50 image lines (i.e. 50 RF lines) [125]-[126]. One image line typically takes about 60 seconds to compute and the whole image can take 50 minutes to simulate [126]. Simulating 3D images and 3D flow takes even more time. Three 3D images of 50×231 lines corresponding to each catheter configuration takes about 24 days (on 32 CPU 600 MHz Pentium III PC cluster), which is not practical for iterative work [125]-[126]. In addition, developing a new fast simulation method for generating realistic B-mode US images is out of scope of this thesis since speckle noise in US emanates from signals generated by tissue cells and connective tissues. However, in the experiments conducted for this thesis, the catheter is submerged into a container filled with water, which does not emanate speckle noise when scanned as shown in Figure 9.

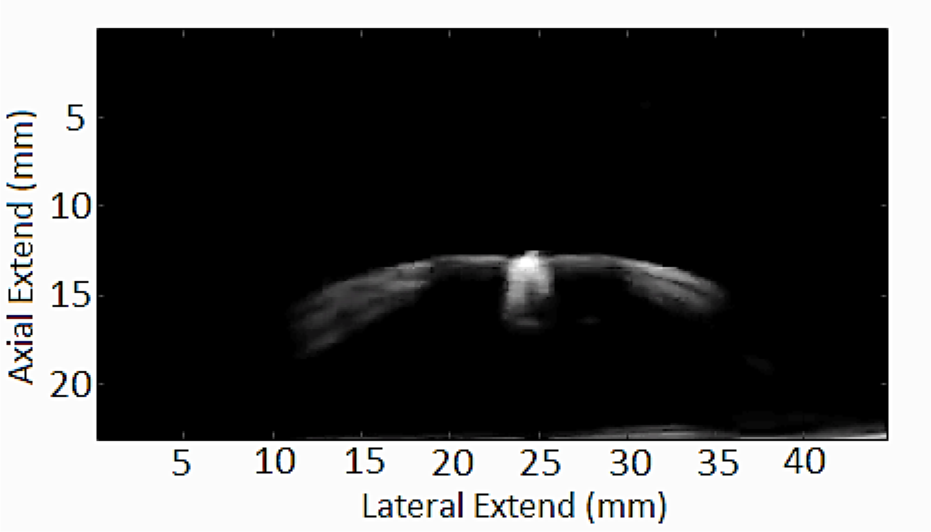


Figure 9 – A transverse US image of the cross section of catheter submerged into water-based gel (taken using SONIXTOUCH RESEARCH Q+)

As illustrated in Figure 9, the cross section of the catheter does not look circular in the US image due to reverberation artifacts (as opposed to circular cross sections programmed in US transverse images). These artifacts occur due to bouncing of the US wave between materials of different acoustic impedance such as the catheter and the surrounding water [57]. In other words, when the velocity of sound in the rounded structure (plastic 3D printed catheter) is different from that of the surrounding medium (water-based gel), a combination of reflection and refraction occur at the edge of the rounded cross section catheter structure. The resulting artifact in Figure 9 has a butterfly shaped structure, the length of which depends on the bouncing echoes that are received by the transducer [57]. This reverberation artifact is referred to as comet tail artifact (CTA) [57].

Therefore, more important than speckle noise, is the effect of the CTA on the robustness of the KF, AKF and PF catheter 3D shape estimation algorithms. The best way to validate these algorithms against CTA is by directly applying them to experimental data (US transverse images like Figure 9 of different catheter configurations submerged into water container). This experimental validation is presented in detail in chapter 5 of this thesis. However, in this section electronic additive Gaussian noise (with mean of 0.2 and variance of 0.01) and multiplicative speckle[25] noise (with variance of

[25] This multiplicative speckle noise added by MATLAB built in function is not to the same as US speckle noise.

0.5[26]) are added to the simulated transverse images using MATLAB built-in functions[27] for disturbing KF, AKF and PF algorithms in their filtering and tracking procedures. Figure 10 illustrates one of the simulated transverse images that comprise the sequences.

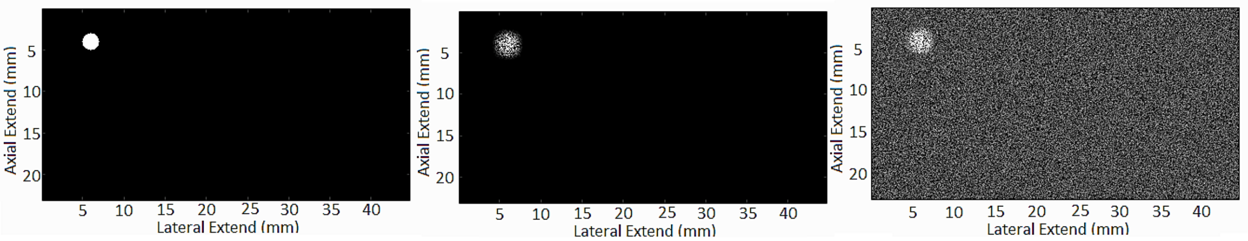


Figure 10 – (a) Original simulated image (b) Original image with speckle noise (c) Original image with speckle and Gaussian noises

In order to examine how the added noise would manifest itself in the quality of measurement states in KF, AKF and PF algorithms, the best threshold of the image in Figure 10(c) is achieved as shown in Figure 11. This thresholded image is used in KF, AKF and PF when measurement states are obtained to update or correct the predicted states.

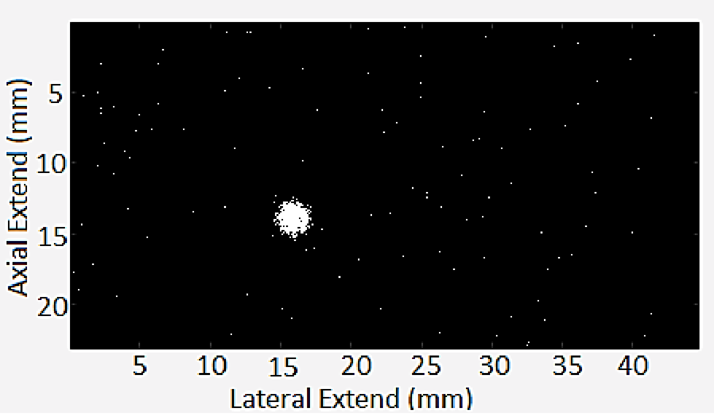


Figure 11 – Thresholded transverse image of Figure 10(c)

[26] The numbers chosen for mean and variances that define the noises are arbitrary with the purpose of distorting the US images enough to provide distraction for the tracking algorithms.

[27] In MATLAB the function: `imnoise(Original Image, type of noise (i.e. Gaussian/Speckle/…), specifications (i.e. mean/variance/…))` is used to simulate Figure 14. The code for this section is provided in Appendix B.1.

The thresholding technique used is the based on the common binary classification of pixels:

$$\mathbf{C}[u,v] = \begin{cases} 0 & \mathbf{I}[u,v] < t \\ 1 & \mathbf{I}[u,v] \geq t \end{cases} \quad \forall (u,v) \in \mathbf{I}$$

Where the decision is based simply on the value of the pixel [113]. $c[u,v]$ refers to pixel coordinates of the resulting thresholded image, $\mathrm{I}[u,v]$ refers to the pixel coordinates of the original image and t refers to the threshold. The threshold t to obtain the best threshold is 0.75 (i.e. normalized pixel intensity value) and it is the optimum number obtained by inspection using MATLAB Toolbox function ithresh.

The white pixels in Figure 11 serve as distraction to the tracking algorithms and also as the result of the added noises the cross section of the catheter is no longer circular and this deviates the measured center of the blob.

The code for this section can be found in Appendix B.1.

3.5. Kalman Filter Simulation Validation

In this section, the 3D shape estimation of the three catheters using KF is presented. The Kalman filter algorithm explained in section 3.1 and algorithm 3.1 is implemented for the catheters shown in Figures 6, 7 and 8. The distorted versions of the transverse images containing the cross section of these catheters are inputted to the KF algorithm one at the time through the KF loop and at each iteration the next state of the catheter cross section is first predicted and then corrected using the measured state and the associated Kalman gain. The corrected predicted state is then used to predict the state in the next iteration. All of the predicted states are then plotted together to give the estimate of the 3D shape of the catheters.

The flow chart of the KF algorithm provided in Figure 12 shows the inputs to the algorithm, which are discussed next.

In the absence of prior information at time $t = 1$, the prior mean $\boldsymbol{\mu}_0$ is initialized to the catheter's point of insertion or the center of the catheter cross section in the very first transverse image, as follow.

$$\boldsymbol{\mu}_0 = \begin{bmatrix} 60 \\ 40 \\ 0 \\ 0 \end{bmatrix}$$

In addition, as also declared in algorithm 3.1, at $t = 1$, the prior covariance matrix $\boldsymbol{\Sigma}_0$ is typically set to a large multiple of the identity matrix, I, of size 4×4, i.e., $\boldsymbol{\Sigma}_0 = 100\mathbf{I}_{4\times4}$.

The frame acquisition of the SONIXTOUCH RESEARCH Q+ US from BK Ultrasound (Peebody, MA) which is used for experiments in this thesis is also known to be 25 Hz. Therefore,

$$\Delta t = \frac{1}{25} sec = 0.04\, sec$$

Substituting the Δt above in equation 3.6 leads to:

$$\boldsymbol{\Psi} = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0.04 & 0 \\ 0 & 1 & 0 & 0.04 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Considering the fast US image acquisition, the mean change in the state as the consequence of external influences imposed by the dynamic model can be neglected, i.e., $\boldsymbol{\mu}_p = 0$.

The 4×4 matrix $\boldsymbol{\Sigma}_p$ representing the process noise is arbitrarily set in the KF algorithm implementation by inspection to: $\boldsymbol{\Sigma}_p = 0.01\mathbf{I}_{4\times4}$. $\boldsymbol{\Sigma}_p$ is kept constant throughout the KF algorithm simulation.

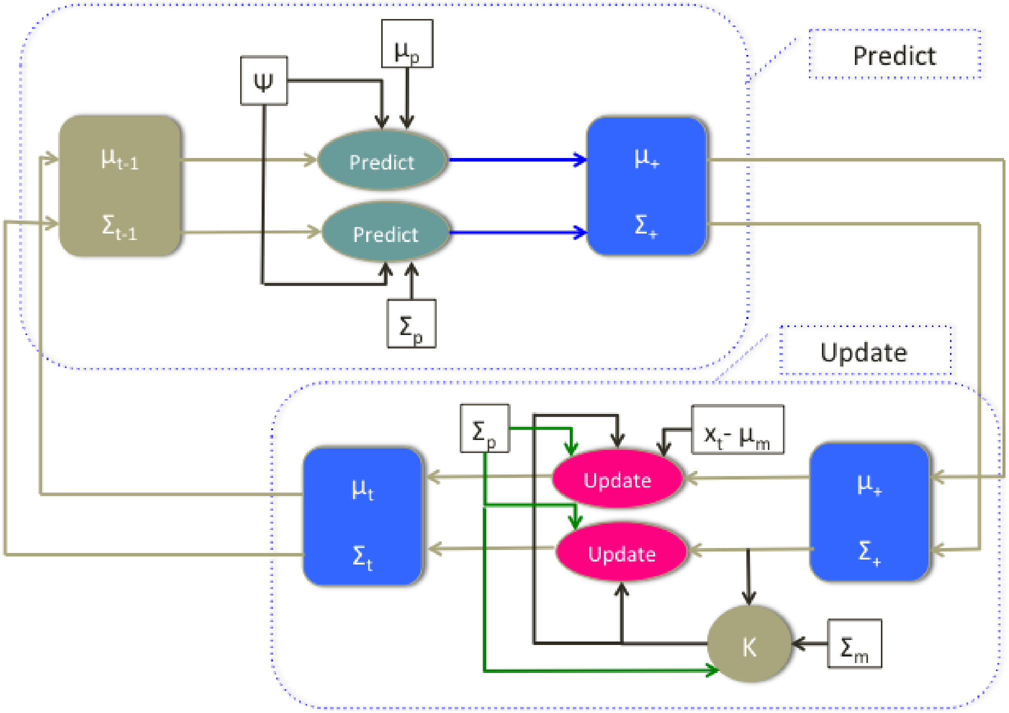


Figure 12 – KF algorithm flow chart

The 2×2 matrix, $\boldsymbol{\Sigma}_m$, quantifies additional uncertainty on the measurements and it is set by inspection until the best fit that leading to the best estimate can be found. In this thesis the first trial of $\boldsymbol{\Sigma}_m$ is set by measuring the variance along catheter cross section in the direction of u and v pixel coordinates. The variances along the pixel coordinates are $\sigma_v = 33$ and $\sigma_u = 31$ (as derived from Figure 11). The covariance given these two numbers is calculated to be 2 using the MATLAB built-in function: cov([33 31]) = 2. Therefore, $\boldsymbol{\Sigma}_m$ can be set to:

$$\boldsymbol{\Sigma}_\mathrm{m} = \begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix}$$

By far, the $\boldsymbol{\Sigma}_\mathrm{m}$ set by the method described above produces the best estimate of the catheter 3D shape compared to other trials.

Finally, $\mathbf{x}_t - \boldsymbol{\mu}_m$, is the measurement obtained from the thresholded frame which is the center of the largest identified blob representing the catheter cross section. This measurement is used with Kalman gain to adjust or correct the current prediction before using it to derive the next prediction of the catheter cross-section state.

Figure 13 illustrates the predicted state against the measured state in a single iteration of the KF loop on its respective corrupted transverse image of the catheter cross section. The red asterisk corresponds to the predicted state and the blue asterisk represents the measured state corresponding to the center of the catheter cross section.

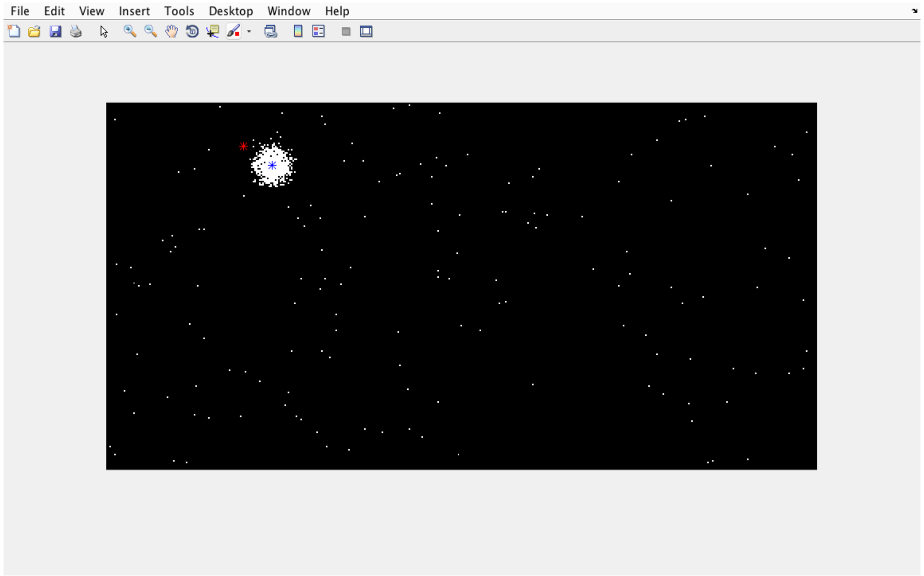


Figure 13 – KF based predicted state (red) and measured state (blue) in a single iteration

The code for KF algorithm in this section is provided in Appendix B.2. The results of KF algorithm or rather the 3D shape estimation of the catheters using KF algorithm is presented in Figures 14, 15 and 16. The blue line represents the actual configuration of the catheter as programmed in section 3.4, Figures 6, 7, and 8, while the red line represents the estimated configuration of the catheter as obtained by KF algorithm.

As shown in these figures the KF algorithm starts very well keeping a good track of the catheters while they are straight. However, KF falls short on tracking the curved segment of the catheter. Therefore one can conclude that KF delivers good tracking performance of the catheter cross-section while it is in linear motion.

The accuracy of the results in Figures 14, 15 and 16 is quantified and discussed in section 3.8. In section 3.8, the CPU run time of the KF-based catheter 3D shape estimation algorithm is also reported and discussed.

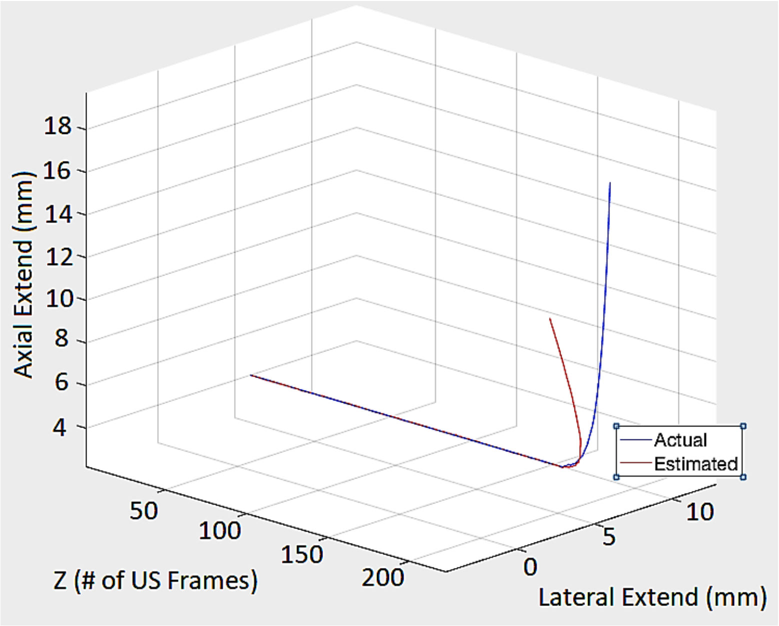


Figure 14 – 3D shape estimation of catheter configuration #1 with KF

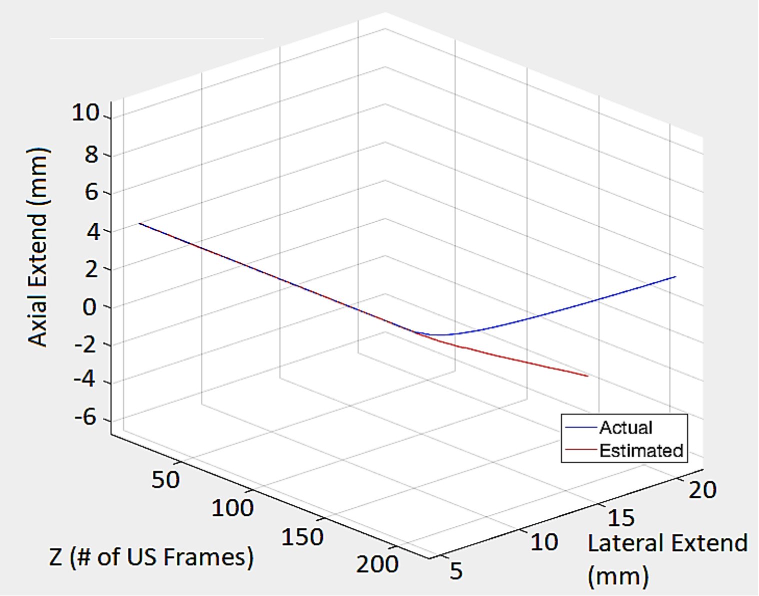


Figure 15 – 3D shape estimation of catheter configuration #2 with KF

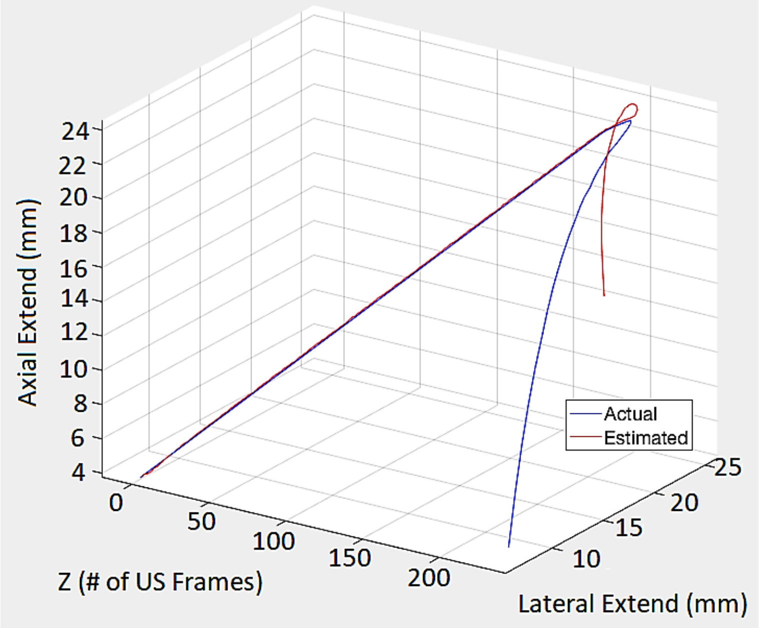


Figure 16 – 3D shape estimation of catheter configuration #2 with KF

Next section repeats the validation presented above only with AKF algorithm instead.

3.6. Adaptive Kalman Filter Simulation Validation

As explained in section 3.2, AKF algorithm is similar to KF algorithm with the exception that special attention is given to the definition of the process noise covariance matrix $\mathbf{\Sigma}_{\text{p}}$ in AKF. In the previous KF simulation, the process covariance noise was taken to be constant: $\mathbf{\Sigma}_p = 0.01\mathbf{I}_{4\times4}$. However, poor estimation of $\mathbf{\Sigma}_p$can lead to large tracking error and divergence as also evident in Figures 14, 15 and 16. On the other hand using equations 3.11-3.15 $\mathbf{\Sigma}_p$ is updated in AKF. The code of 3D catheter shape estimation using AKF is provided in Appendix B.3.

All the inputs to the AKF algorithm are the same as the ones in KF algorithm discussed in the previous section. The one new input required by AKF as also introduced in equations 3.11-3.15 is N. In this simulation program N is set by inspection. N in AKF program dictates that updating the process noise does not start until after the N^{th} transverse image in the sequence. Instead $\mathbf{q}_1$ ... $\mathbf{q}_{\text{N}}$ (calculated in the first N rounds) are used to base the estimate for the process noise starting from the

$(\mathrm{N}+1)^{\mathrm{th}}$ round according to equation 3.15 which calculates $\overline{\mathbf{q}}_{\mathrm{N}} = \left(\frac{1}{\mathrm{N}}\right)\sum_{\mathrm{i}=1}^{\mathrm{N}+1}\mathbf{q}_{\mathrm{i}}$. In these simulations N is set to 5 and $\mathbf{q}_1$ is set to $\boldsymbol{\mu}_0$ as follow:

$$\mathbf{q}_1 = \boldsymbol{\mu}_0 = \begin{bmatrix} 60 \\ 40 \\ 0 \\ 0 \end{bmatrix}.$$

In addition, in equation 3.14, given as:

$\Delta_t = \boldsymbol{\Psi}\boldsymbol{\Sigma}_{t-1}\boldsymbol{\Psi}^T - \boldsymbol{\Sigma}_t$, where $\boldsymbol{\Sigma}_t$ is updated as $\boldsymbol{\Sigma}_+$ in algorithm 3.1. Also for $t = 1$:

$\Delta_1 = -\boldsymbol{\Sigma}_1 = -100\mathrm{I}_{4\times 4}$

The results of 3D shape estimation algorithm based on AKF are presented next. As it can be seen in Figures 17, 18 and 19, updating the process noise, $\boldsymbol{\Sigma}_p$, presents significant improvements in the 3D shape estimation of the catheters especially along the curvy segments of the catheter.

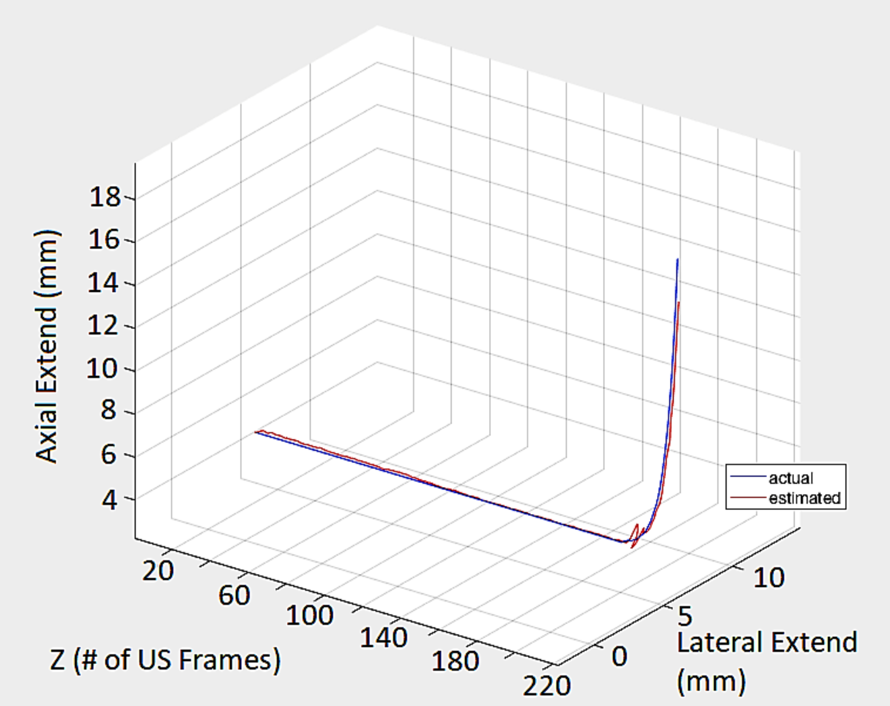


Figure 17 – 3D shape estimation of catheter configuration #1 with AKF

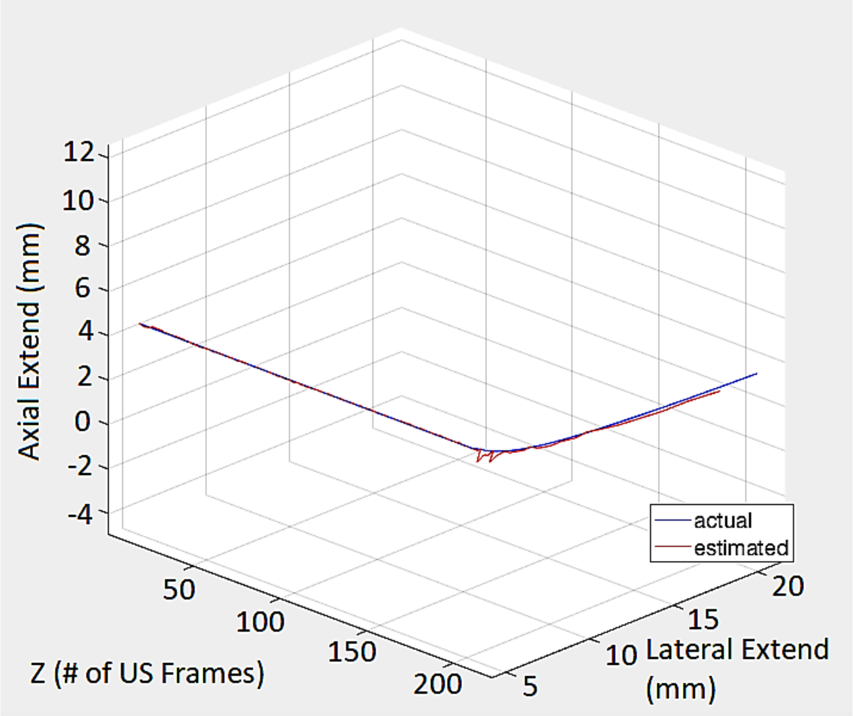


Figure 18 – 3D shape estimation of catheter configuration #2 with AKF

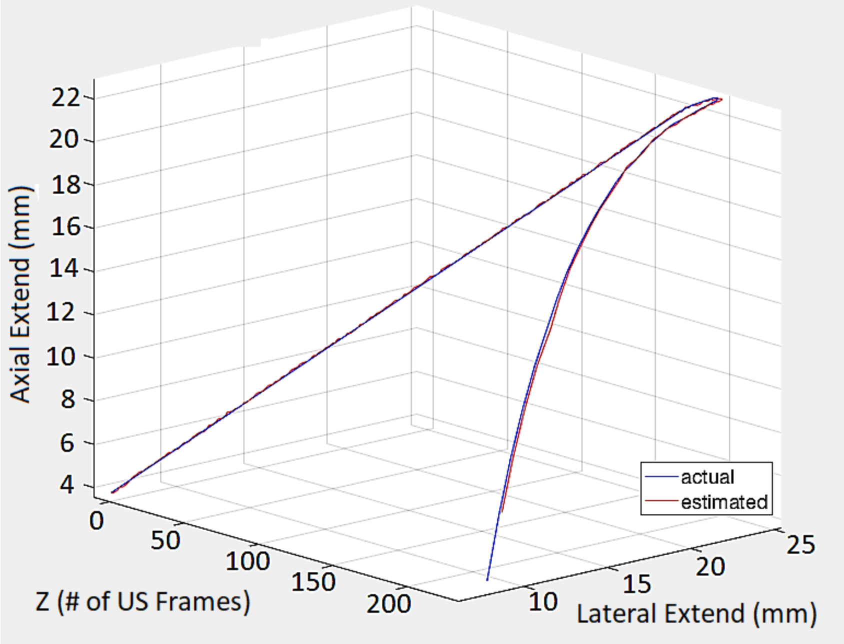


Figure 19 – 3D shape estimation of catheter configuration #3 with AKF

Even though KF failed to accurately estimate the nonlinear section of the catheters, AKF provides good estimation of the shape of the catheter in both linear and nonlinear sections.

The accuracy of the results in Figures 17, 18 and 19 is quantified and discussed in section 3.8. In section 3.8, the CPU run time of the AKF-based catheter 3D shape estimation algorithm is also reported and discussed.

3.7. Particle Filter Simulation Validation

In this section, the PF algorithm is implemented as discussed in section 3.3 to estimate the 3D shape of the catheters in Figures 6, 7, and 8. The MATLAB code for PF is provided in Appendix B.4 and the results are as follow.

PF algorithm implemented here starts with 400 particles each with equal weight of goodness of $\frac{1}{400} = 0.0025$. At time $t = 1$, these particles are randomly distributed about the catheter point of insertion, which is assumed to be known to be (60,40), representing the center of the catheter cross section at the very first transverse image of the sequence. These particles are shown in blue in Figure 20.

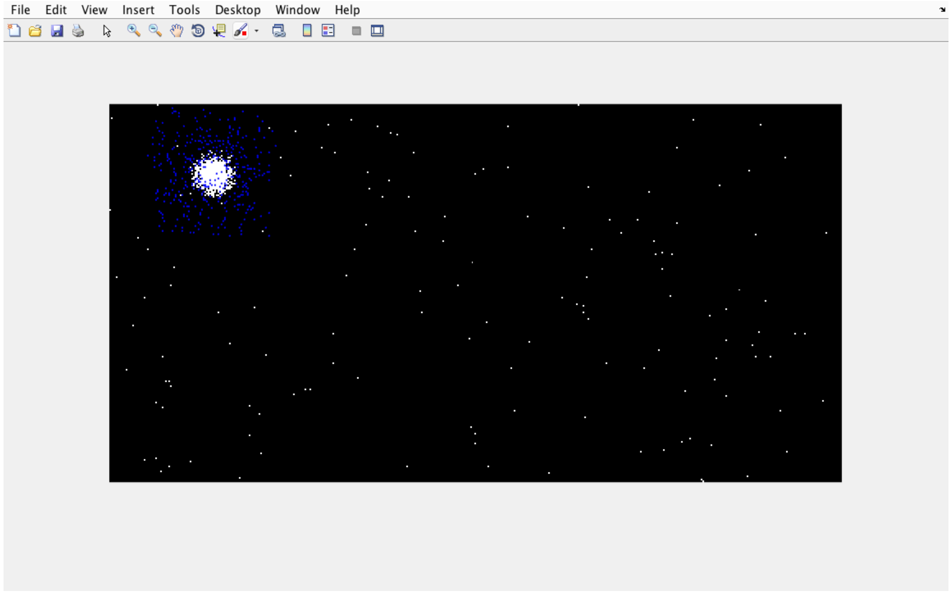


Figure 20 – Randomly distributed particles at the start of PF iteration

Next, each of the randomly distributed particles are reweighed according to how close they are to the edge of the largest blob in the transverse image which corresponds to the catheter cross section.

Out of the reweighed particles, the ones with highest weighs and therefore closest to the detected edge of the catheter cross section are sampled out through which an ellipse is fitted. The fitted ellipse represents an estimation of the contour of the catheter cross-section. Figure 21 illustrates the sampled particles, which are rated closest to the contour of the largest blob. In addition, the green ellipse is fitted through these points, which follows the equation noted on top of Figure 21.

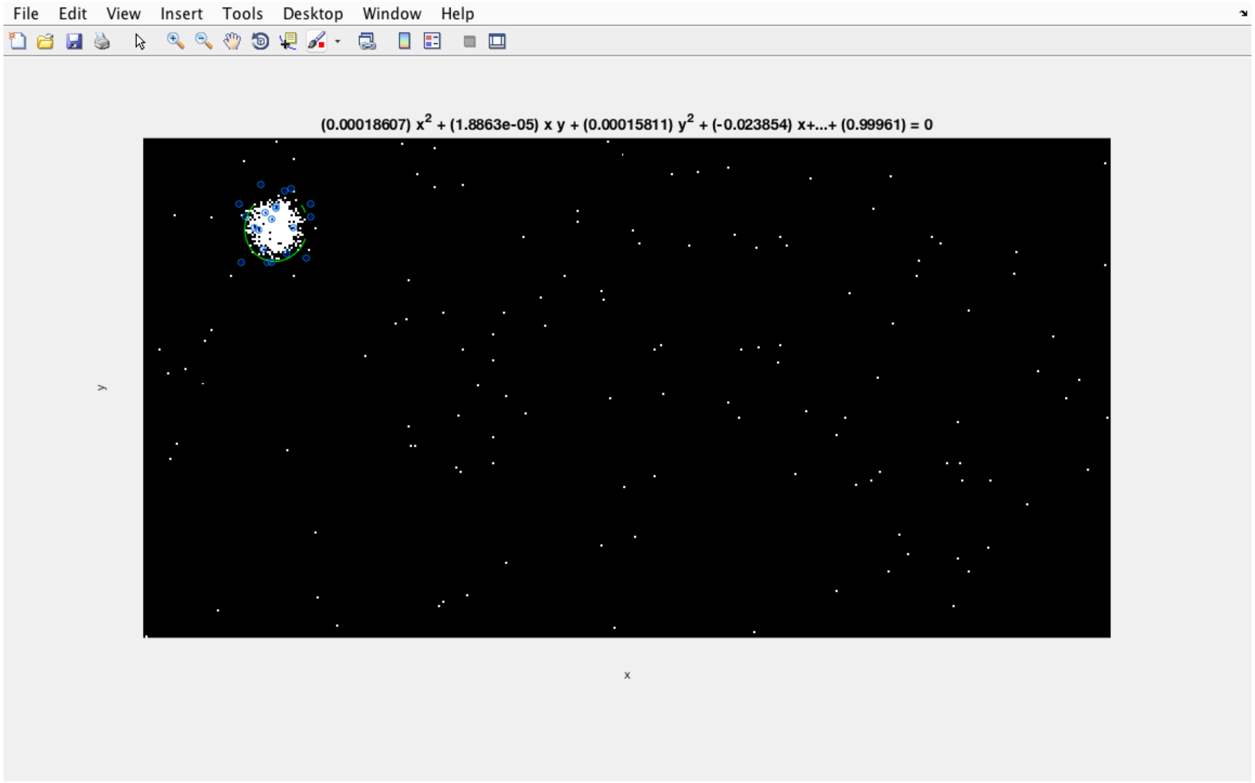

File Edit View Insert Tools Desktop Window Help
$(0.00018607)\ x^2 + (1.8863e-05)\ x\ y + (0.00015811)\ y^2 + (-0.023854)\ x + \ldots + (0.99961) = 0$
y
x

Figure 21 – Sampled particles (blue) and fitted circle (green)

The center of the fitted ellipse as shown in Figure 21 corresponds to the predicted state of the catheter cross-section center. These predicted states are put together in a 3D plot to provide a 3D estimate of the configuration of each catheter as modeled in Figures 6, 7 and 8. The results of 3D shape estimation with PF algorithm are presented in Figures 22, 23 and 24 respectively.

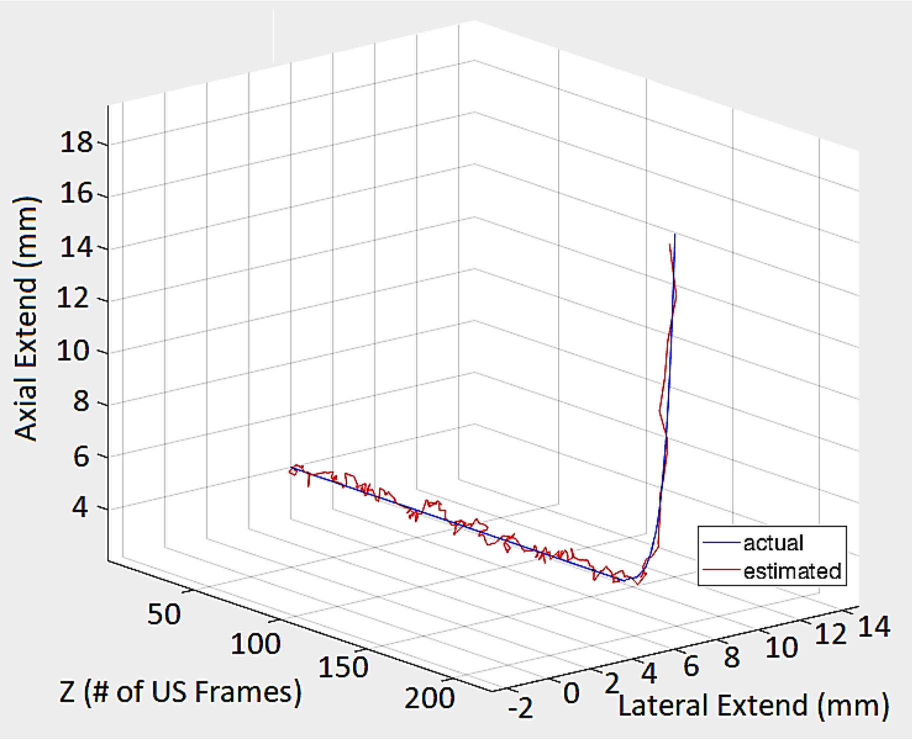


Figure 22 – 3D shape estimation of catheter configuration #1 with PF

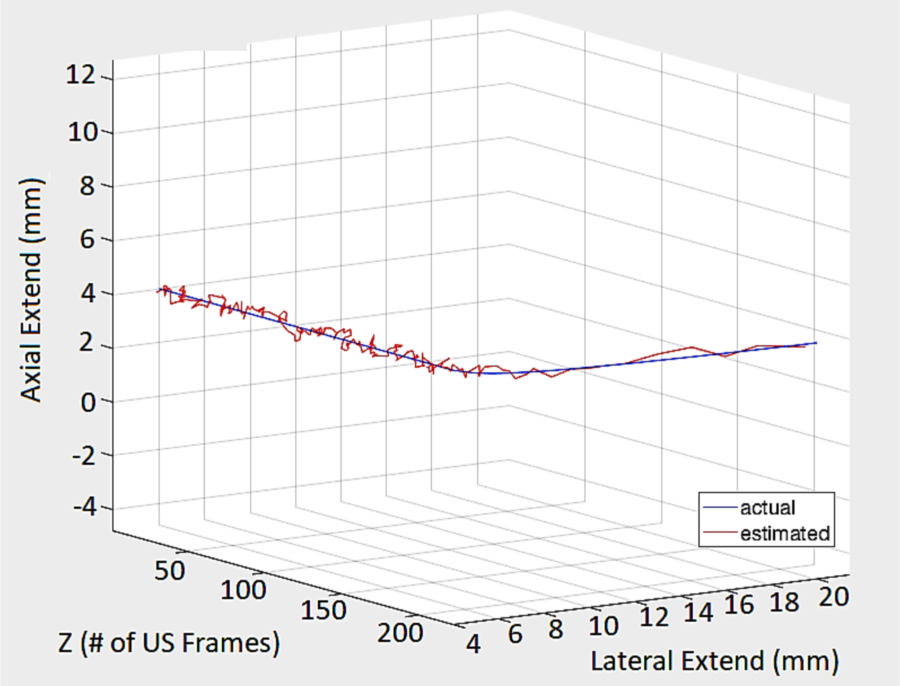


Figure 23 – 3D shape estimation of catheter configuration #2 with PF

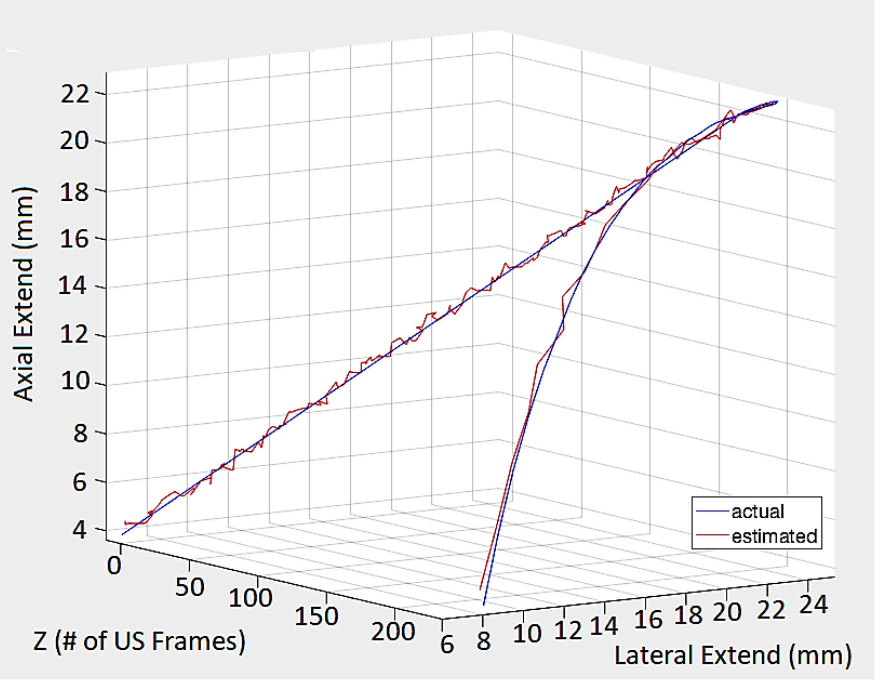


Figure 24 – 3D shape estimation of catheter configuration #3 with PF

As shown in Figures 22, 23 and 24, even though the PF estimation of the catheter maintains an oscillatory pattern of fluctuations, it tracks the catheter cross section fairly close and stable throughout both straight and curvy sections.

The accuracy of the results in Figures 22, 23 and 24 is quantified and discussed in section 3.8. In section 3.8, the CPU run time of the PF-based catheter 3D shape estimation algorithm is also reported and discussed.

3.8. Speed and Accuracy

Table 2 summarizes the CPU time each algorithm (KF, AKF and PF) takes to perform the 3D shape estimation of the catheters. These CPU times were obtained by running the algorithms on the state-of-the-art HP OMEN 3.4 GHz PC with windows 10 operating system.

As evident in Table 2, KF presents with the shortest CPU run time and PF with the longest run time with AKF in between.

Table 2

CPU times of each temporal probability algorithm

Algorithm Type	CPU time in seconds (± 0.01sec)
KF	0.24
AKF	0.46
PF	1.98

Table 3 summarizes the root mean square error (RMSE) and the Hausdorff distance between the actual catheter configurations (Figures 6, 7, 8) and the estimated 3D catheter shape

Table 3

Accuracy of 3D shape estimation algorithms

Algorithm/Catheter	Hausdorff distance (± 1 pixels)	Hausdorff distance (± 0.1 mm)[28]	RMSE (± 1 pixels)	RMSE (± 0.1 mm)
KF/Catheter #1	67	6.7	5	0.5
KF/Catheter #2	67	6.7	5	0.5
KF/Catheter #3	35	3.5	15	1.5
AKF/Catheter #1	3	0.3	2	0.2
AKF/Catheter #2	2	0.2	2	0.2
AKF/Catheter #3	5	0.5	3	0.3
PF/Catheter #1	6	0.6	3	0.3
PF/Catheter #2	8	0.8	3	0.3
PF/Catheter #3	6	0.6	3	0.3

The presented RMSEs and Hausdorff distances were calculated using MATLAB functions [130]-[131]. The RMSE is a measure of how concentrated the estimation is around the line representing the actual configuration of the catheter. Similarly, Hausdorff distance[29] represents how far the 3D estimate configurations are from the actual 3D configurations. Both of these methods may be used to assign a scalar score to the similarity between the estimates and actual models. However, in

[28] As determined in experiments using ULTROSONIX RESEARCH Q+, 1 mm is equivalent to approximately 10 pixels.

[29] A discussion of mathematical details in calculating Hausdorff distance is out of scope of this thesis; however, the reader may refer to: https://en.wikipedia.org/wiki/Hausdorff_distance, for more information

literature RMSE is usually the one used to report the accuracy of methods and algorithms in shape estimation and tracking of the catheters.

According to Table 3, the best estimate of catheter # 1, # 2 and #3 is provided by the AKF algorithm followed by PF and then KF. KF does not exhibit good tracking of catheter cross section along curvy segments of the catheter. However, along the linear segments of the catheter, KF provides the best and smoothest estimate than AKF and PF. PF provides a close track of the catheters cross section throughout their length, however, this tracking is not smooth and tends to fluctuate. On the other hand, AKF provides the closest and smoothest track of the catheters cross-section along their length.

3.9. Summary

Table 4 summarizes this chapter by putting together the RMSE scores of catheter 3D shape estimations reported in Table 3 of section 3.8.

Overall, from table 4, one may expect that the 3D shape estimation of the catheter with AKF and PF algorithms can provide best results, even compared to the methods described in literature (Table 4). This expectation is validated in Chapter 5 where the 3D shape estimation algorithms presented in this thesis are applied to the experimental data.

Table 4

Summary of simulation RMSEs

Methods	RMSE (mm)
KF/Catheter #1	0.5 ± 0.1 mm
AKF/Catheter #1	0.2 ± 0.1 mm
PF/Catheter #1	0.3 ± 0.1 mm
KF/Catheter #2	0.5 ± 0.1 mm
AKF/Catheter #2	0. 2 ± 0.1 mm
PF/Catheter #2	0.3 ± 0.1 mm
KF/Catheter #3	1.5 ± 0.1 mm
AKF/Catheter #3	0.3 ± 0.1 mm
PF/Catheter #3	0.3 ± 0.1 mm

A note on the speed of KF, AKF and PF algorithms against their corresponding estimation accuracies is in order. According to Table 2, KF has the shortest run time but according to Table 3 it provides the worst estimate of the catheter 3D configuration especially along the curvy segments. On the other hand, AKF and PF exhibit comparable estimation of the 3D catheter shape especially along the curvy segments with AKF providing much smoother estimations. However, PF takes approximately 4 times longer time than AKF for estimation. Therefore, even though PF exhibits better estimation most of the time, it comes with high computation cost.

Even though in this section the numerical simulation method was used to validate the performance of KF, AKF and PF in estimating the 3D configuration of the catheter, experimental validation is still required to evaluate these algorithms against real disturbances such as authentic US image artifacts and unstable handling of the US probe while scanning the volume of interest manually. The experimental validation is described and analyzed in chapters 4 and 5 that follow.

Chapter IV

EXPERIMENTS

KF, AKF and PF based catheter 3D shape estimation algorithms developed in the previous chapter are further validated by experiment. The aim is to evaluate the robustness of these algorithms in estimating the 3D shape of the catheters in the face of actual US artifacts and other inherent experimental discrepancies. In this chapter, the materials, methods and procedures involved in obtaining the experimental data are described.

In section 4.1, catheter modeling and their set up inside the plastic container is described. In section 4.2, the catheter's 3D configuration inside the container is measured using CMM. These measurements serve as the ground truth to validate the 3D shape estimation results from the US and Optotracked US experimental data. Section 4.3 presents un-calibrated manual US scanning over the catheters in the container to obtain a series of transverse US images. Calibration of manual transverse US scans in the CMM's frame is achieved using Optotracker sensors by first calibrating the transducer probe in the CMM's frame in section 4.4 and then calibrating the US image frame with respect to the US probe in section 4.5. The overall calibration error is considered in section 4.6. Finally in section 4.7, the appropriate syncing/matching of the data from the two machines (i.e. US system and NDI Optotracker) is discussed.

4.1. Catheter Modeling and Placement

Three different configurations of catheters were modeled in TINKERCAD, which is a 3D CAD design tool. The layouts of these catheters are illustrated in Figure 30 as they appear in the CAD design tool's work plane. The smallest grid on this work plane is 1mm adding up to 1 cm squares. A ruler in centimeters is also added to this work plane to provide a better perspective on the size and layout of the catheters.

As shown in Figure 30, catheter #1 has 3 mm diameter circular cross section, extending 95 mm in length. This catheter curves in plane for 30 mm at the last 25 mm of its length. It's important to note that the whole structure of catheter #1 lies in the work plane. Catheter #2 has square cross section with 3 mm sides and it follows an arbitrary curvy path along its 100 mm length. This catheter also lies on the work plane entirely. Catheter #3 has 3 mm diameter circular cross-section and it extends to 90 mm in length. This catheter curves out of plane for 30 mm at the last 20 mm of its length. To facilitate the out of plane curve, a support structure had to be placed under it so that the structure would not collapse while the printing is in progress.

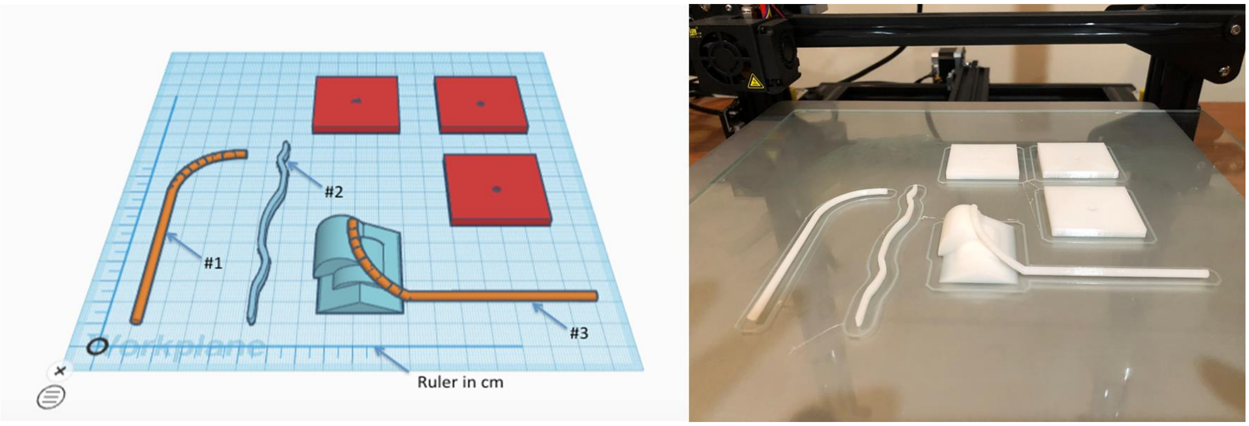


Figure 25 – (a) Catheter modeling in TINKERCAD work plane (b) 3D printed catheters with their attachment structures

The catheters #1, 2 and 3 illustrated in Figure 25(a) were modeled after observing different configurations possible by Std Crv Blazer II HTD® (110 cm in length and 2.5 mm in cross section diameter). This catheter is shown in Figure 26. It is important to note that out of the whole length of this catheter only the distal end about 7 cm from the tip of the catheter can be manipulated into different configurations. Three of the possible configurations of this distal end are shown in Figure 27.

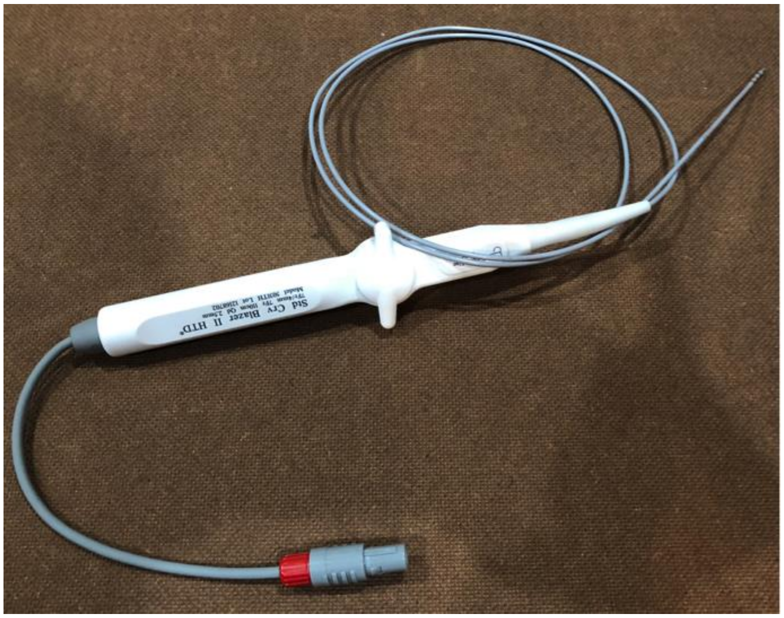


Figure 26 – Std Crv Blazer II HTD® Catheter

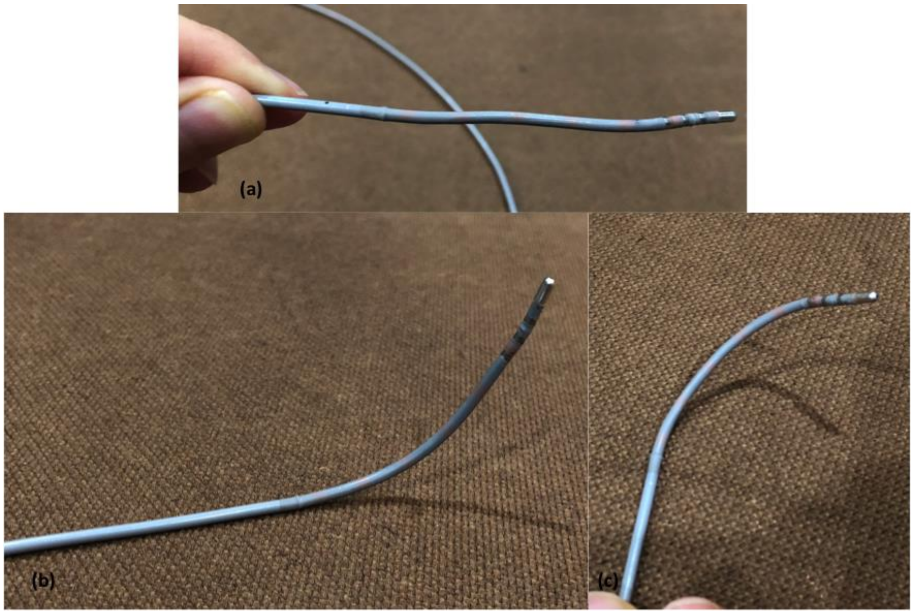


Figure 27 – (a) Arbitrary curve which inspires the 3D model, Catheter #2, in Figure 30 (b) In-plane curve which inspires 3D model of catheter # 1 and (c) Out of plane curve which inspires 3D model of catheter #3

For each of the three catheters an attachment structure is designed. These three attachment structures are shown in Figure 25 (a) as red rectangular cuboids of 40 mm × 40 mm × 4 mm. These attachment structures also include a groove in their middle that is designed to hold the catheters in place. Figure 25 (b) shows the finish print. The catheters and attachment parts were printed using a white PLA filament that went through a 0.4 mm nozzle. The 3D printer used for this job is Creality CR-10 (Creality 3D, Shenzhen, China), which is Cartesian robot machine type with rectangular-build volume of 300 mm × 300 mm × 400 mm.

The catheter attachments were glued to rectangular plastic containers as shown in Figure 28. The catheters were also glued securely in their respective grooves.

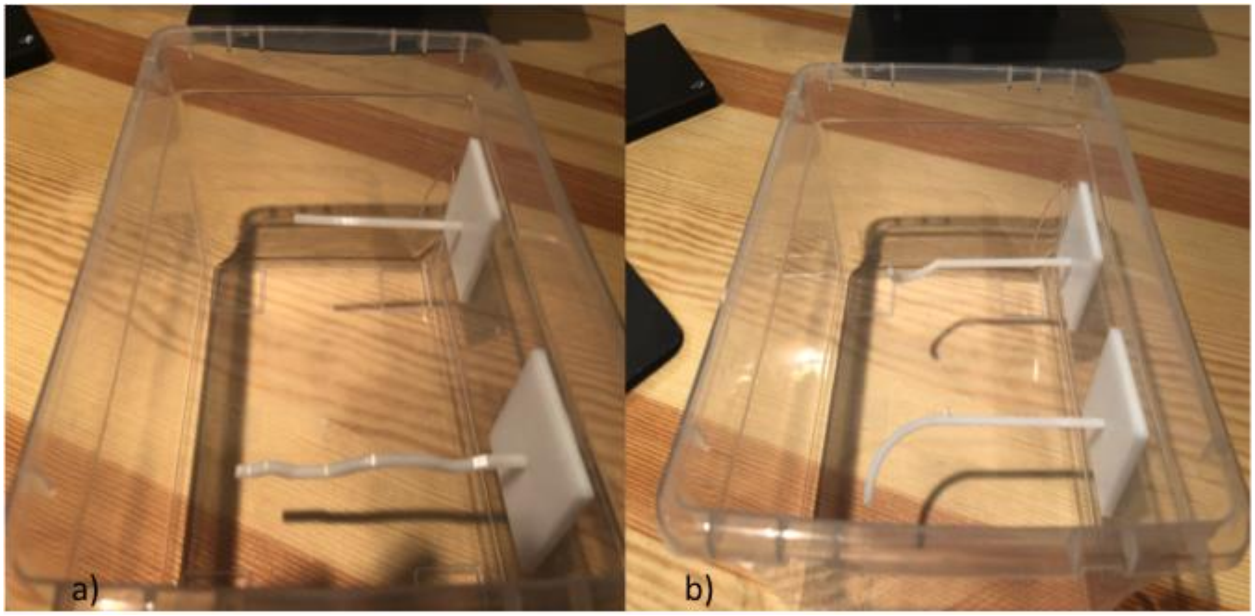


Figure 28 – Catheters' placement in containers (a) catheter #2 and a straight catheter (b) catheter #1 and #3

Water resistant and heat resistant UHU polymer super glue was used for attachments to the plastic containers. The container in Figure 28(a) holds catheter #2 and a straight plastic rod of 3 mm circular diameter and 70 mm total length. This straight catheter is separately printed by the same 3D printer and is used as a simple exercise for implementation of the shape estimation algorithms. The container in Figure 28(b) holds the planar catheter #1 and non-planer catheter #3.

The final step in setting up the experiment is to set water based gel inside the plastic containers. For each container, 1.42 litres of 60℃ water is mixed with 18.5 grams of Porcine Gelatin powder (one teaspoon gelatin powder sets one cup of liquid). The mixture is poured inside the containers holding the catheters carefully from a corner so that the liquid volume builds up slowly from underneath the catheters. The containers are then placed inside the fridge (at 6℃). It takes minimum of four hours for the gel to set. Setting the water-based gel inside the containers makes the scanning of US probe over the volume containing the catheters safer for the transducer probe and more manageable (no splashing of water) without introducing speckle noise to the US images. Figure 29 shows one of the containers after the water-based gel is set inside of it.

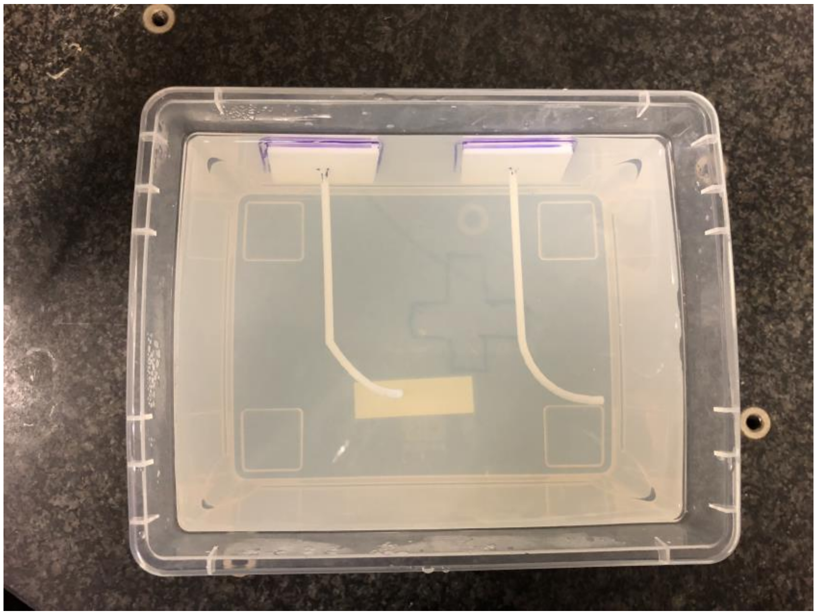


Figure 29 – water-based gel set inside the plastic container

4.2. Experimental Strategy for Validation of Results

Before setting the water-based gel inside the containers, the are measured in the 3D frame of a CMM machine. These concrete measurements are then used to build up a 3D measurement model of each of the four catheters. Each measurement model represents the ground truth of their respective catheter and is used to validate the 3D US shape estimation results. A MITUTOYO CMM (Mitutoyo, Takatsu-ku, Kawasaki) machine is used to measure the catheters in its 3D frame (Figure 30).

Before taking any measurements, the origin with respect to which the measurements are reported by the CMM machine must be defined. This is achieved by first choosing the point of origin, then positioning the measuring needle tip at this chosen point and resetting the system by pushing down the ZER button on the machine as shown in Figure 31. Now, this point will be the origin (x=0, y=0, z=0) and every other point in the CMM's space is measured with respect to this origin in units of mm.

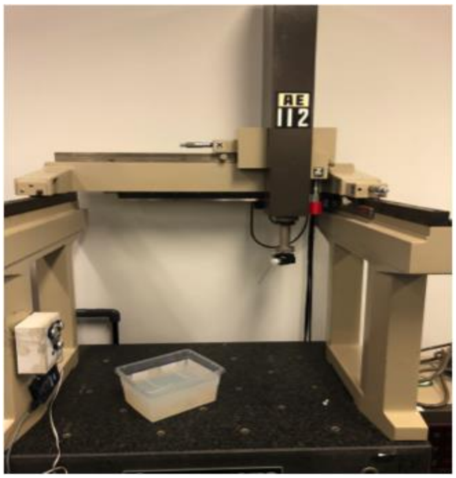


Figure 30 – The MITUTOYO CMM machine

Figure 32 illustrates the direction of positive and negative x, y and z measurements with respect to the origin. Figure 33 demonstrates the procedure of measuring the catheters along their length using the machine. As shown in Figure 33, a series of points along each of the four catheters are measured with respect to the designated origin. These points are then plotted in 3D space using MATLAB and shown in Figures 34 and 35. The configuration of the four catheters shown in these figure are considered as the ground truth representations of these catheters and are used to validate the 3D US shape estimation results as presented in chapter 5 of this thesis.

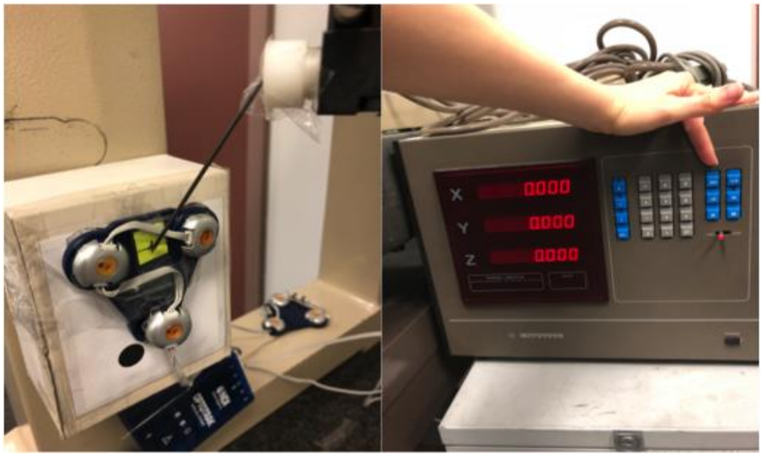


Figure 31 – Setting the origin of CMM measurements

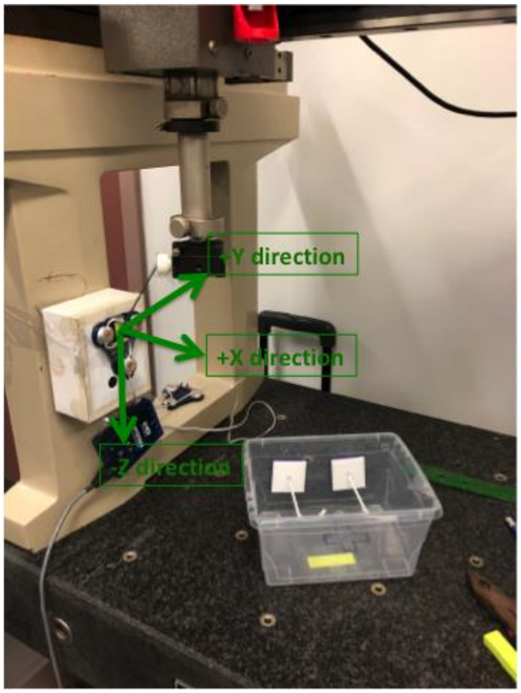


Figure 32 – Direction of x, y and z measurements with respect to the origin in CMM machine

The origin is chosen to be the mid-point between the two Optotracker sensors as shown in Figure 31. This point also marks the origin of the reference frame in the Optotracker field of view, as its set up will be explained in section 4.4. The origin of the CMM machine measurement frame and the origin of the Optotracker reference frame must be the same if their respective measurements are to be compared against one another. This comparison will be discussed in greater detail in chapter 5.

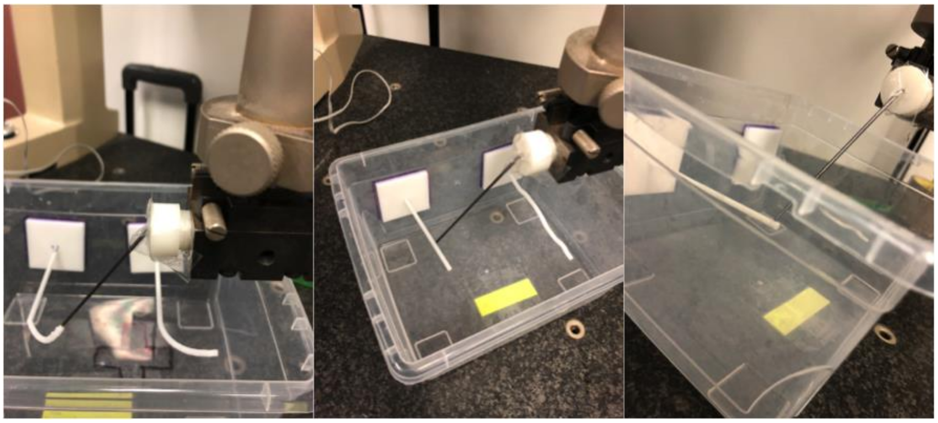


Figure 33 – Measuring points along the length of catheters

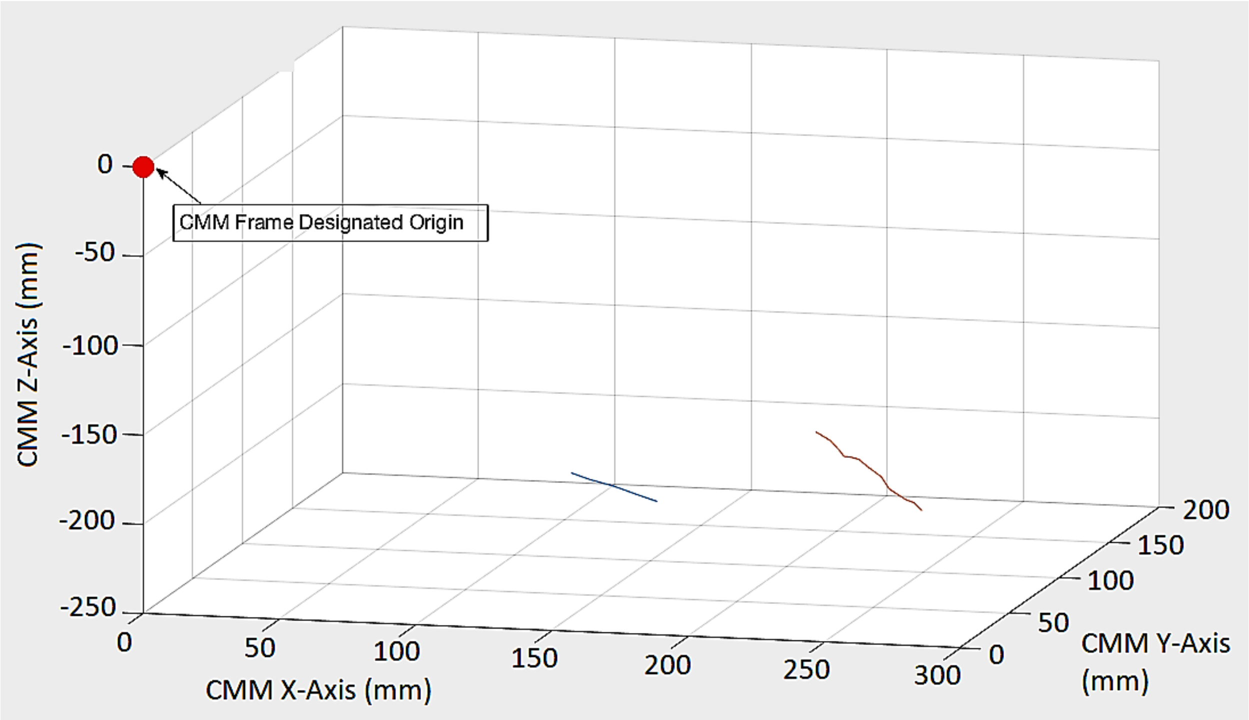


Figure 34 – CMM measurements of the container with the straight (blue) catheter and catheter #2 (red) (all measurements in mm)

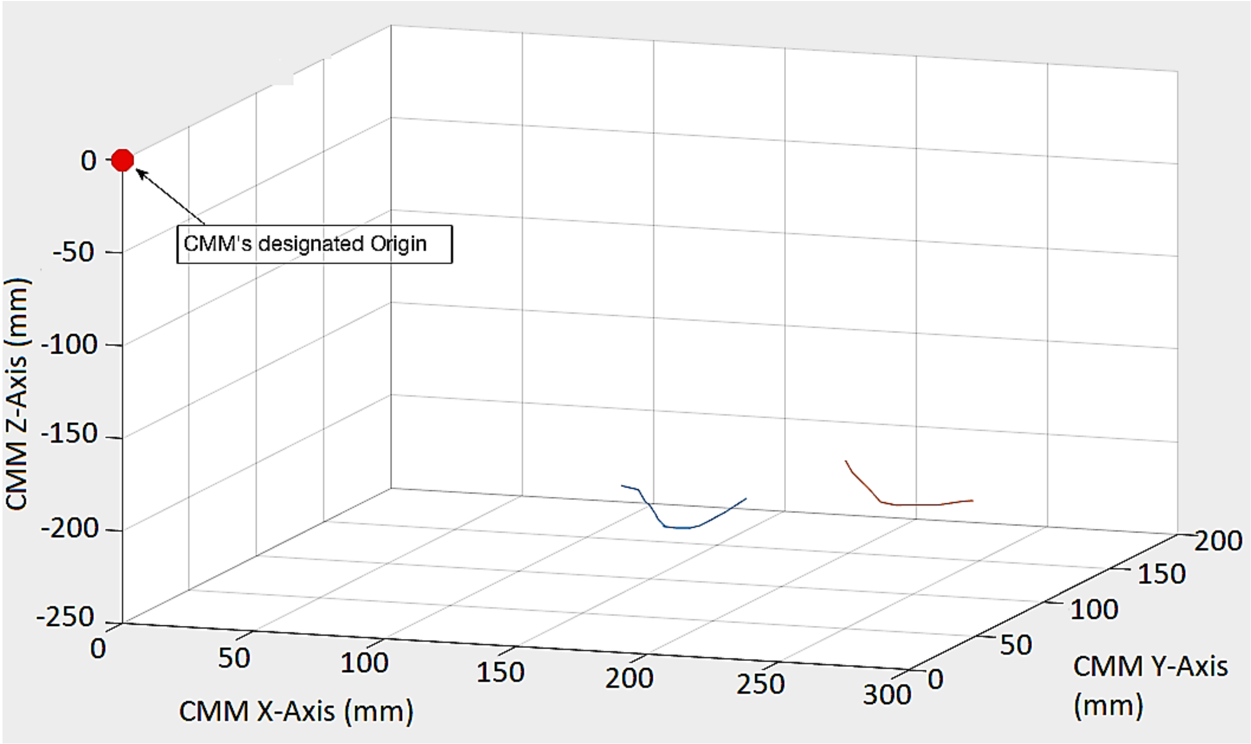


Figure 35 – CMM measurements of the container with catheter #3 (blue) and catheter #1 (red) (all measurements in mm)

As depicted in Figures 34 and 35, the four catheters are measured with respect to the origin as they are positioned and oriented in the CMM's measuring frame. The same position and orientation of the origin and catheters are maintained for experiments with Optotracker and US as discussed in section 4.4. All the measurements are in millimeters. It is also important to mention that since these catheters consist of either circular cross section of 3 mm or square cross section of 3 mm × 3 mm (only catheter #2), the measured z-axis of each point on the surface of these catheters is subtracted by $\frac{3\,mm}{2} = 1.5\,mm$, so that the measured points are representative of the center axis through the catheters.

Also, as it can be seen in Figure 34, the straight catheter tilts a bit upward and catheter # 2 tilts a bit downward. In Figure 35, catheter # 3 (plotted in blue) curves out of plane and upward while catheter # 1 (red) curves in plane and to the right. These orientations are consistent with how these catheters came out after gluing them to their attachments. Therefore, measurements depicted in Figures 34 and 35 serving as calibrated direct measurements obtained in CMM's measuring frame can be considered as ground truth to be used for validation of calibration procedures and 3D shape estimation algorithms. [30]

4.3. Un-calibrated Ultrasound Experiments

A series of US transverse images were obtained by scanning the US probe across the catheters inside the water-based gel. The probe was manually moved steadily with constant speed so to prevent any deviations in its orientation. No calibration is performed at this point.

About 5 mm of cold water was poured onto the surface of the gel inside the container to assist in smoother movement of the US transducer over the gel. If this small amount of water is not added, the US transducer would ruin the surface of the gel due to resistance during the scan and also the transducer beams would disperse.

Figure 36 shows a sample transverse US image from each of the four catheters. These images show the cross section of each catheter at a point in time. As evident, the water-based gel provides a speckle free US image and the only artifact present in these images are the occasional comet tail artifact emitted from the surface of the 3D printed plastic catheters (e.g., Figure 36(a)).

[30] The numbering of catheters mentioned in the text corresponds to the catheter numbers presented in Figure 30

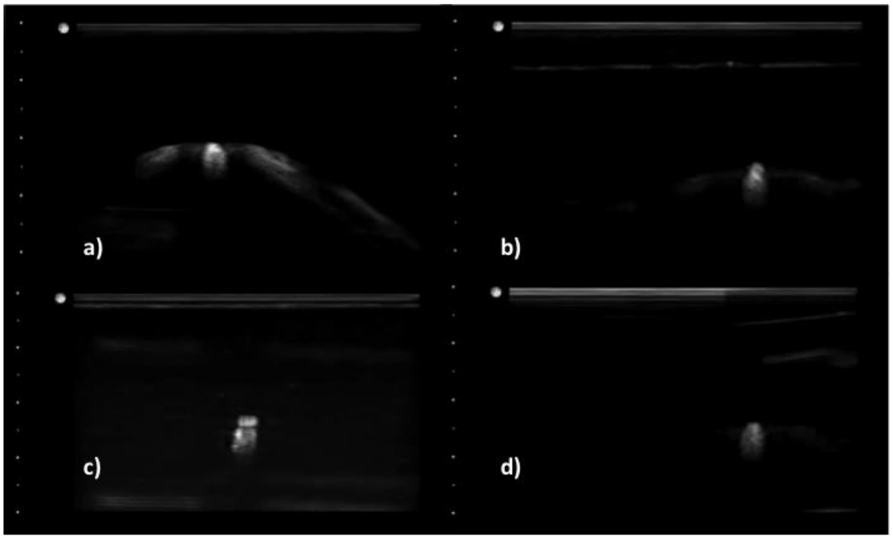


Figure 36 – sample US transverse images of (a) straight catheter (b) catheter #1 (c) catheter #2 and (d) catheter #3[31]

To obtain the US images of Figure 45, SONIXTOUCH RESEARCH Q+ US system with access to clinical grade grey-scale image and raw B mode data was utilized. The scans were performed using L14-5W/60 linear wide transducer of 5-14 MHz bandwidth and 4mm × 60.4 mm footprint. The US system and the linear transducer are shown in Figure 46.

The US transverse images were obtained by setting the transducer frequency to 14.0 MHz, transmission depth to 4 cm (as shown in Figure 38), and B-mode gain of 50%. Other settings were kept in their default values such as speed of sound (SOS) of 1540 m/s, mechanical index of less than 0.74 (MI<0.74), and thermal index of less than 0.27 (TIS<0.27). By inspection these settings produced the best and crispiest US images. In addition, the acquisition frame rate was set to high, which correspond to 33 frames per second (FPS).

[31] Catheter # 1, 2 and 3 refer to the ones modeled in Figure 30

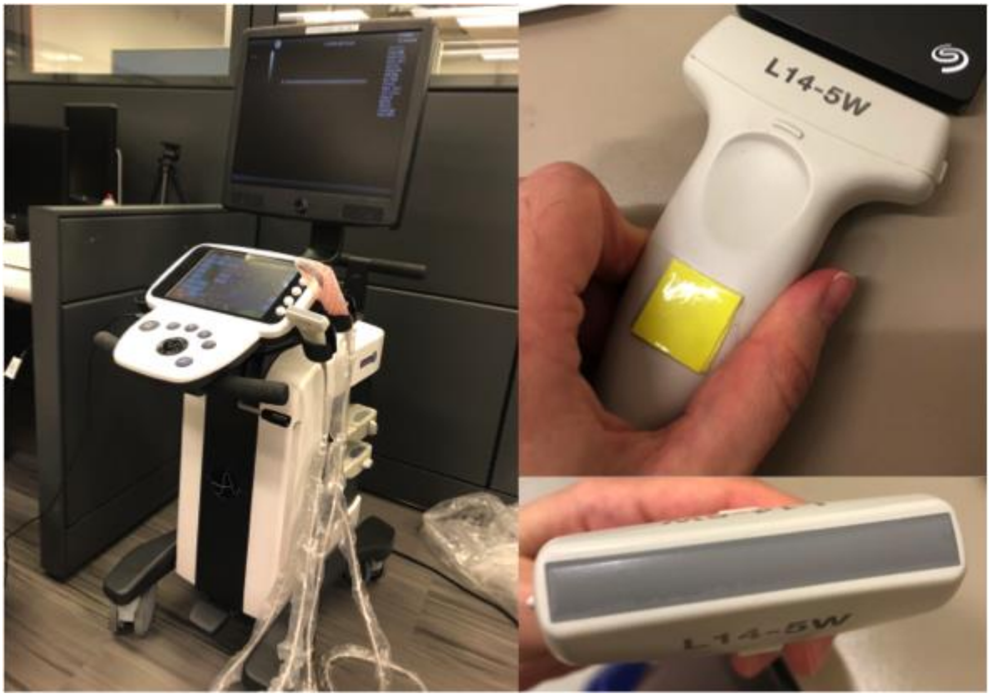


Figure 37 – ULTRASONIX RESEARCH Q+ US system along with L14-5W linear transducer

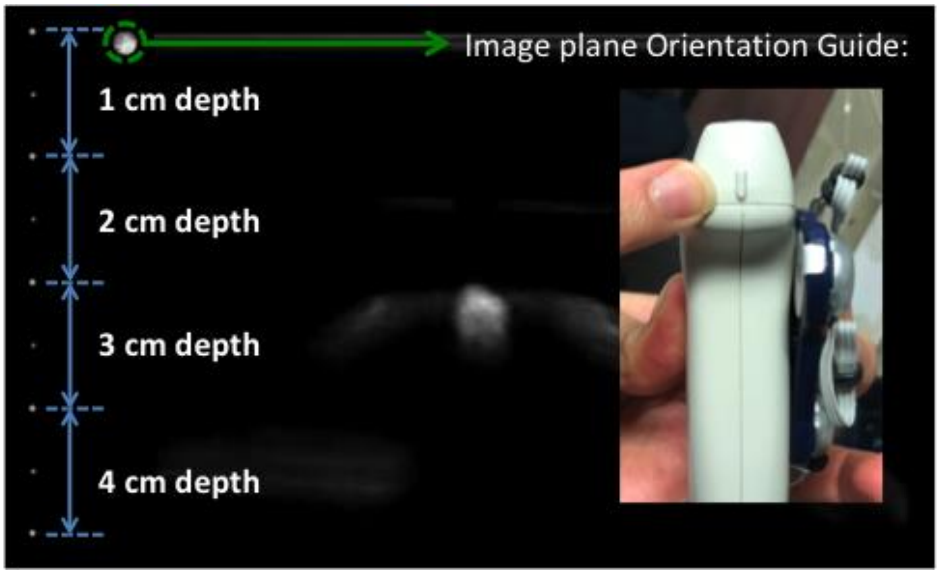


Figure 38 – US image plane specifications

As shown in Figure 38, the dots on the left side indicate the depth captured by US which in this case correspond to a total of 4 cm as set manually before. In addition, the white circle at the top left corner (marked by the dashed green circle) indicates that the top left corner of the image corresponds to the transmissions received at the notched side of the transducer. This information assists in orienting the transducer to the desired direction and provides a guide of the orientation of the image with respect to the transducer. This information will be useful in calibrating the US image plane with respect to the US probe frame as discussed in the next section.

4.4. Calibration of Ultrasound Probe using NDI Optotracker

The un-calibrated US experiments presented in the previous section lack spatial perspective and therefore the implementation of tracking algorithms cannot be validated against CMM's measurements of Figures 34 and 35. In order to calibrate the US transverse images, the first step is to calibrate the US transducer probe with respect to the designated world frame using the NDI Optotracker (Northern Digital Inc., Waterloo, Ontario) as discussed in this section. The world reference frame referred to here has the same origin and x, y, z axes orientation as the one designated for the CMM's frame within which the direct measurements were obtained. Defining the Optotracker reference frame to be the same as the CMM is intuitive in that the measurements obtained using the Optotracker can then be compared with direct measurements of the CMM.

The NDI Optotracker shown in Figure 39 provides 3DOF pose, (x, y, z), for any number of sensors attached to it with respect to the internally defined global reference frame. The sensors of the NDI Optotracker are shown in Figure 40. A triangular frame (Figure 40) supplied by the NDI Optotracker manufacturer can hold three individual sensors, which together can be used to define a coordinate frame. In this section one triangular frame is used to designate the reference coordinate frame (i.e. the previously mentioned world frame) and another triangular frame is used to designate the US transducer probe coordinate frame. The goal here is to set the NDI Optotracker to measure (i.e. by detecting the triangularly framed sensors) the position and orientation of the US transducer probe frame with respect to the reference frame.

The reference coordinate frame is stationary as it is secured in position to the frame of the CMM's machine given the whole experiment is set up in the CMM's base. However, the US transducer probe coordinate frame moves as the probe scans across the catheters over the water-based gel (container of which is securely fixed in position on the CMM's base). The NDI Optotracker is configured to provide the position and orientation (i.e. pose, ξ, defined by 6DOF $(x, y, z, \theta_r, \theta_p, \theta_y)$[32]) of the US probe's frame with respect to the pose of the reference frame. This configuration is set using NDI Optotracker's computer software package (NDI First Principle and NDI Architecture). The Optotracker sensor placement on the CMM's frame and US probe are demonstrated in Figures 41 and 42 respectively. These figures also demonstrate how each coordinate frame was configured in the NDI software package in terms of the location of its origin and direction of axes. This configuration is important in that it designates the increasing or decreasing directions of sensors reading especially as they move through space against one another. For the reference frame to be consistent with the CMM's reference frame (both being the same), sensor #1 is set to be on the -z axis, sensor #2 on -y axis and sensor #3 on +y axis. Therefore, the origin of the reference frame is set to be exactly in the midpoint between sensor #2 and sensor #3 and the +x axis is perpendicular to the triangular frame as shown in Figure 41. For simplicity, the US probe coordinate frame is configured in a similar fashion to the reference frame coordinate frame with the origin in the middle of sensor #7 and #8, +y axis passing through sensor #7, -y axis passing through sensor #8, -z axis passing through sensor #9 and +x axis perpendicular to the triangular frame as shown in Figure 42.

[32] (x in mm, y in mm, z in mm, roll angle in degrees, pitch angle in degrees, yaw angle in degrees)

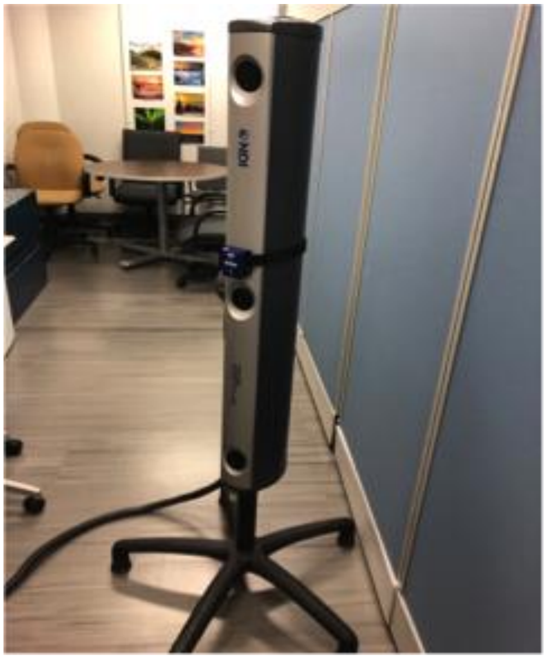


Figure 39 – The NDI Optotracker

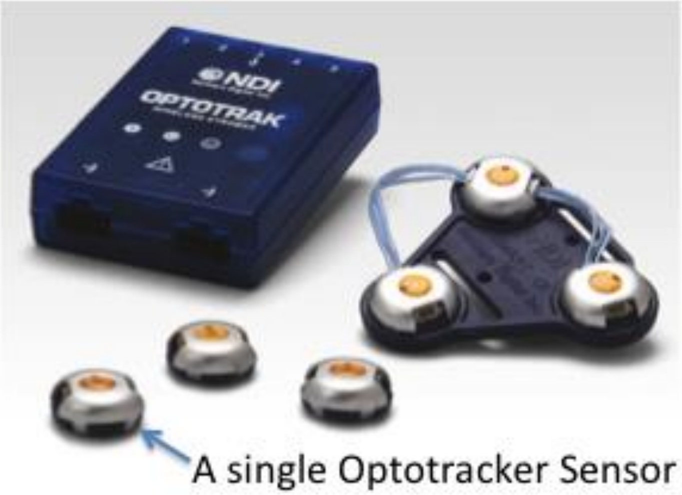
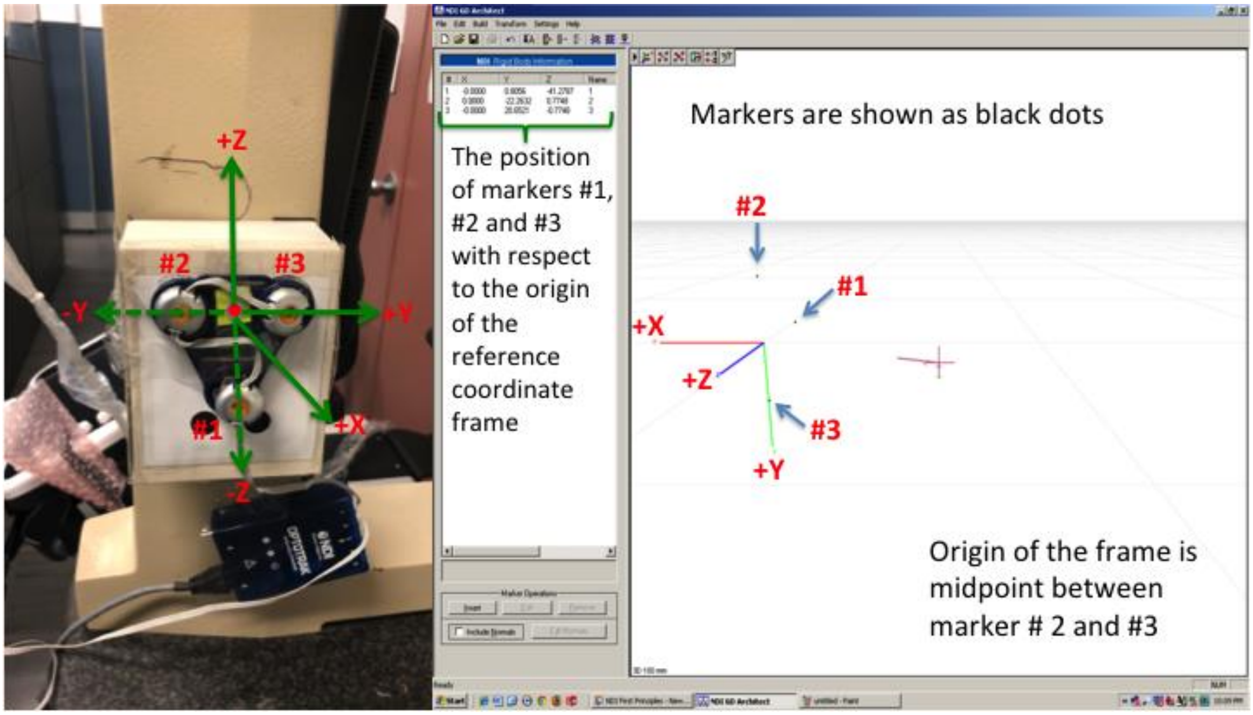


Figure 40 – Sensors of the NDI Optotracker


+Z
#2
#3
-Y
+Y
#1
+X
-Z
NDI 6D Architect
File Edit Build Transform Settings Help
NDI Rigid Body Information
#2
#1
+X
+Z
#3
+Y

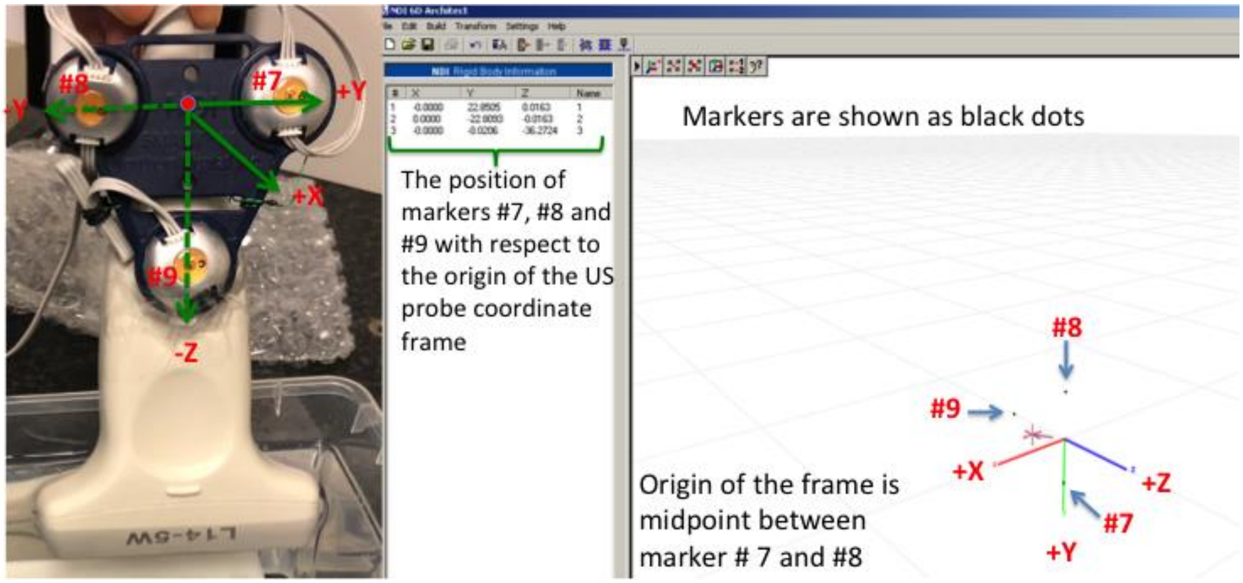
#	X	Y	Z	Name
1	-0.0000	0.8056	-41.2787	1
2	0.0000	-22.2632	0.7748	2
3	-0.0000	20.0521	-0.7748	3

The position of markers #1, #2 and #3 with respect to the origin of the reference coordinate frame

Markers are shown as black dots

Origin of the frame is midpoint between marker # 2 and #3

Figure 41 – Configuring world reference coordinate frame


#8
#7
-Y
+Y
+X
#9
-Z
L14-5W
NDI 6D Architect
File Edit Build Transform Settings Help
NDI Rigid Body Information
#8
#9
+X
+Z
#7
+Y

#	X	Y	Z	Name
1	-0.0000	22.0505	0.0163	1
2	0.0000	-22.0090	-0.0163	2
3	-0.0000	-0.0206	-36.2724	3

The position of markers #7, #8 and #9 with respect to the origin of the US probe coordinate frame

Markers are shown as black dots

Origin of the frame is midpoint between marker # 7 and #8

Figure 42 – Configuring US transducer probe coordinate frame

Summing up, the concept of the intended calibration is illustrated in Figure 43. The global reference frame, {G}, is internally defined by the NDI Optotracker manufacturer. The reference frame, {R}, is configured so that the Optotracker provides its pose with respect to {G}. Since the reference frame is secured fixed on the CMM's frame, the pose ξ_R^G (i.e. the position and orientation of the frame {R} with respect to the frame {G}) is constant and never changes through time.

On the other hand, the US transducer probe is not stationary as it scans across the volume of interest to accumulate a series of US transverse image frames. Therefore, the pose of the US probe changes over time and is not constant. The NDI Optotracker is configured to provide the position and orientation of the US probe frame, $\{P\}$, with respect to the reference frame, $\{R\}$ over time. Figure 44 illustrates a sample reading of the position and orientation of the US probe frame with respect to the reference frame over 25 time-steps provided by the NDI Optotracker.

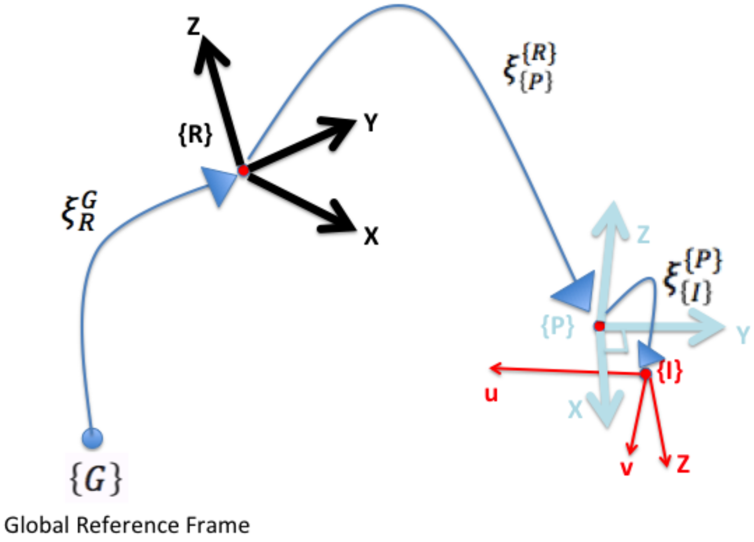
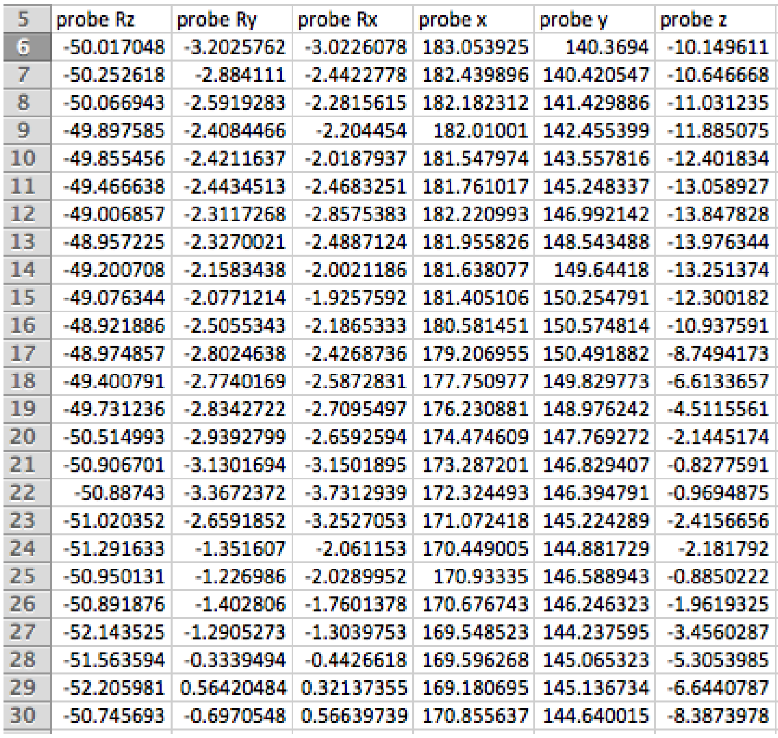


Figure 43 – Experimental Calibration setting



5	probe Rz	probe Ry	probe Rx	probe x	probe y	probe z
6	-50.017048	-3.2025762	-3.0226078	183.053925	140.3694	-10.149611
7	-50.252618	-2.884111	-2.4422778	182.439896	140.420547	-10.646668
8	-50.066943	-2.5919283	-2.2815615	182.182312	141.429886	-11.031235
9	-49.897585	-2.4084466	-2.204454	182.01001	142.455399	-11.885075
10	-49.855456	-2.4211637	-2.0187937	181.547974	143.557816	-12.401834
11	-49.466638	-2.4434513	-2.4683251	181.761017	145.248337	-13.058927
12	-49.006857	-2.3117268	-2.8575383	182.220993	146.992142	-13.847828
13	-48.957225	-2.3270021	-2.4887124	181.955826	148.543488	-13.976344
14	-49.200708	-2.1583438	-2.0021186	181.638077	149.64418	-13.251374
15	-49.076344	-2.0771214	-1.9257592	181.405106	150.254791	-12.300182
16	-48.921886	-2.5055343	-2.1865333	180.581451	150.574814	-10.937591
17	-48.974857	-2.8024638	-2.4268736	179.206955	150.491882	-8.7494173
18	-49.400791	-2.7740169	-2.5872831	177.750977	149.829773	-6.6133657
19	-49.731236	-2.8342722	-2.7095497	176.230881	148.976242	-4.5115561
20	-50.514993	-2.9392799	-2.6592594	174.474609	147.769272	-2.1445174
21	-50.906701	-3.1301694	-3.1501895	173.287201	146.829407	-0.8277591
22	-50.88743	-3.3672372	-3.7312939	172.324493	146.394791	-0.9694875
23	-51.020352	-2.6591852	-3.2527053	171.072418	145.224289	-2.4156656
24	-51.291633	-1.351607	-2.061153	170.449005	144.881729	-2.181792
25	-50.950131	-1.226986	-2.0289952	170.93335	146.588943	-0.8850222
26	-50.891876	-1.402806	-1.7601378	170.676743	146.246323	-1.9619325
27	-52.143525	-1.2905273	-1.3039753	169.548523	144.237595	-3.4560287
28	-51.563594	-0.3339494	-0.4426618	169.596268	145.065323	-5.3053985
29	-52.205981	0.56420484	0.32137355	169.180695	145.136734	-6.6440787
30	-50.745693	-0.6970548	0.56639739	170.855637	144.640015	-8.3873978

Figure 44- Screen shot of Optotracker reading providing the position and orientation of the US probe frame with respect to the reference frame over 25 steps of time

Rx corresponds to the roll angle, Ry corresponds to the pitch angle, and Rz corresponds to the yaw angle.

The pose of {P} with respect to {R}, $\xi_{\{P\}}^{\{R\}} \sim \mathbf{T}_{\{P\}}^{\{R\}} \in SE\,(3)$ can be obtained using the following equation[33]:

$$\mathbf{T} = \begin{bmatrix} c_\gamma c_\beta & -s_\gamma c_\alpha + c_\gamma s_\beta s_\alpha & s_\gamma s_\alpha + c_\gamma s_\beta c_\alpha & x \\ s_\gamma c_\beta & c_\gamma c_\alpha + s_\gamma s_\beta s_\alpha & -c_\gamma s_\alpha + s_\gamma s_\beta c_\alpha & y \\ -s_\beta & c_\beta s_\alpha & c_\beta c_\alpha & z \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{4.1}$$

where $\alpha = R_x, \beta = R_y, \gamma = R_z, x = x_{\{P\}}^{\{R\}}, y = y_{\{P\}}^{\{R\}}$ and $z = z_{\{P\}}^{\{R\}}$.

[33] Whenever convenint, the shorthand notations $c_\theta = \cos(\theta)$ and $s_\theta = \sin(\theta)$ are used.

Finally, to calibrate the US image frame, $\{I\}$, with respect the reference frame, $\{R\}$, the homogeneous transformation denoted by $\xi_{\{I\}}^{\{P\}}$ must be derived. Here, $\xi_{\{I\}}^{\{R\}}$ is also constant as the position and orientation of the US image frame does not change with respect to the probe's frame over time given they are both part of the probe structure which does not deviate through time. The derivation of $\xi_{\{I\}}^{\{R\}}$ requires additional experiments and procedures, which are explained in detail in the next section.

4.5. Calibration of Ultrasound Image Frame using Calibration Phantom

To calibrate the US image coordinate frame in the world reference frame, in addition to $\xi_{\{R\}}^{\{G\}}$ and $\xi_{\{P\}}^{\{R\}}$, $\xi_{\{I\}}^{\{R\}}$ (i.e., the pose of the US image frame with respect to the probe frame), which is constant in time throughout the whole experiment, must be derived. This is accomplished by applying the N-wire fast calibration method proposed by Pagoulatos *et al.* [136].

A special calibration phantom was printed using Creality CR-10 3D printer and 1.75 mm white PLA filament. The STL design file[34] and the 3D printed phantom are shown in Figures 45.

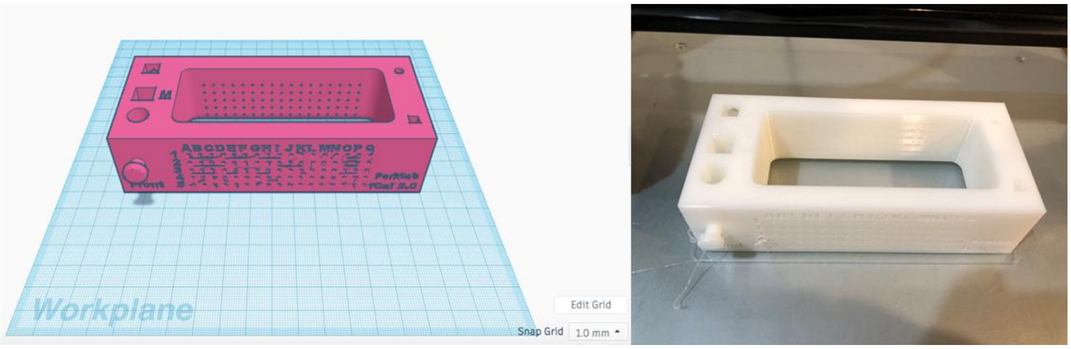


Figure 45- STL design file of the calibration phantom (fCal2.0) used to calibrate the US image frame

[34] The phantom model in STL format is downloadable from the following link: https://subversion.assembla.com/svn/plus/trunk/PlusLib/data/CADModels/fCalPhantom/

The fCla2.0 phantom s recommended for 30-90 mm calibration depth, which is applicable to the purposes of this thesis.

Galvanized steel wires of about 0.2-0.3 mm were passed through the phantom holes to form fiducial lines. The wires were placed as to form a set of 6 fiducials, each with the shape of the letter ‘N’, hence called N-fiducials. The N-fiducials are grouped in five planes, each containing six N-fiducials. Figure 46 draws the top view of the five planes showing the arrangement of the six N-fiducials in each plane. Figure 47 shows a picture of the wired fCal phantom.

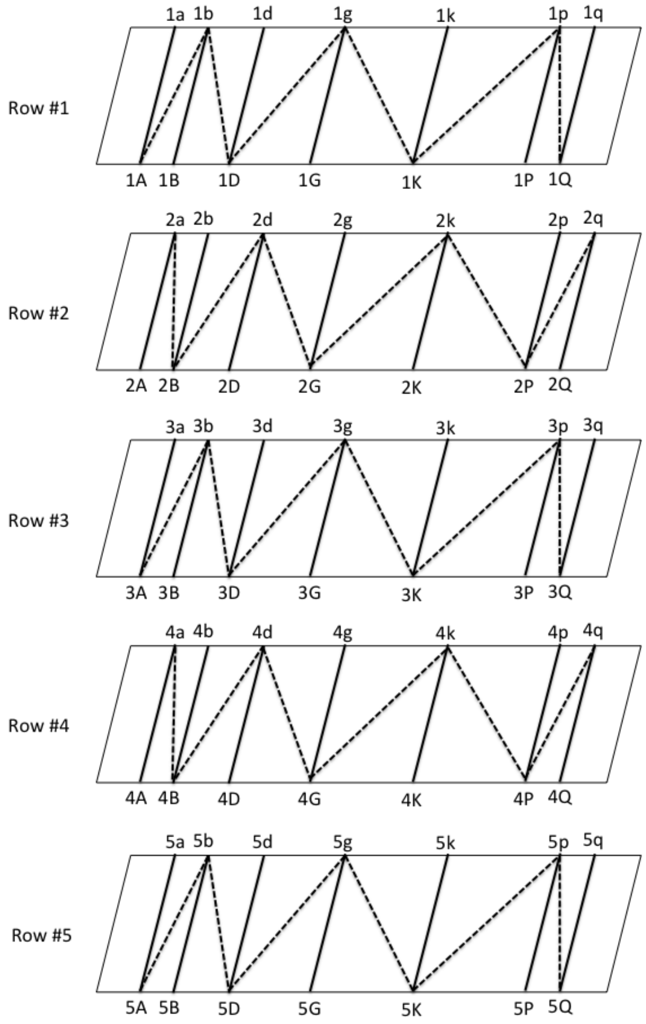


Figure 46- The width and direction of the N-fiducials for the five rows of the fCal2.0 phantom. Solid lines denote the parallel wires (perpendicular to the phantom walls) and dashed lines denote the oblique wires of the N-fiducials

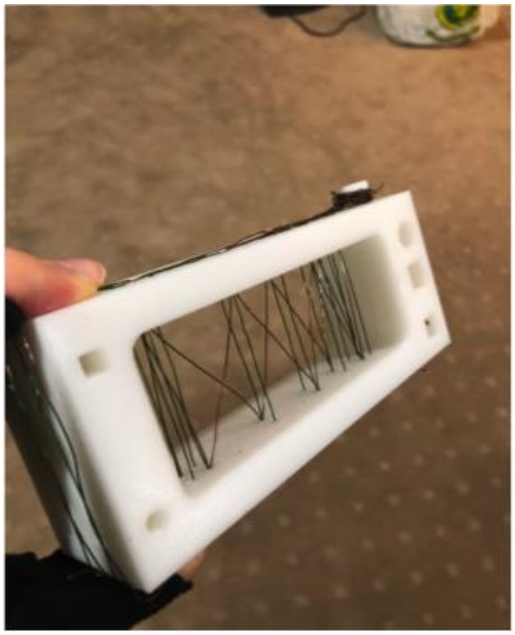
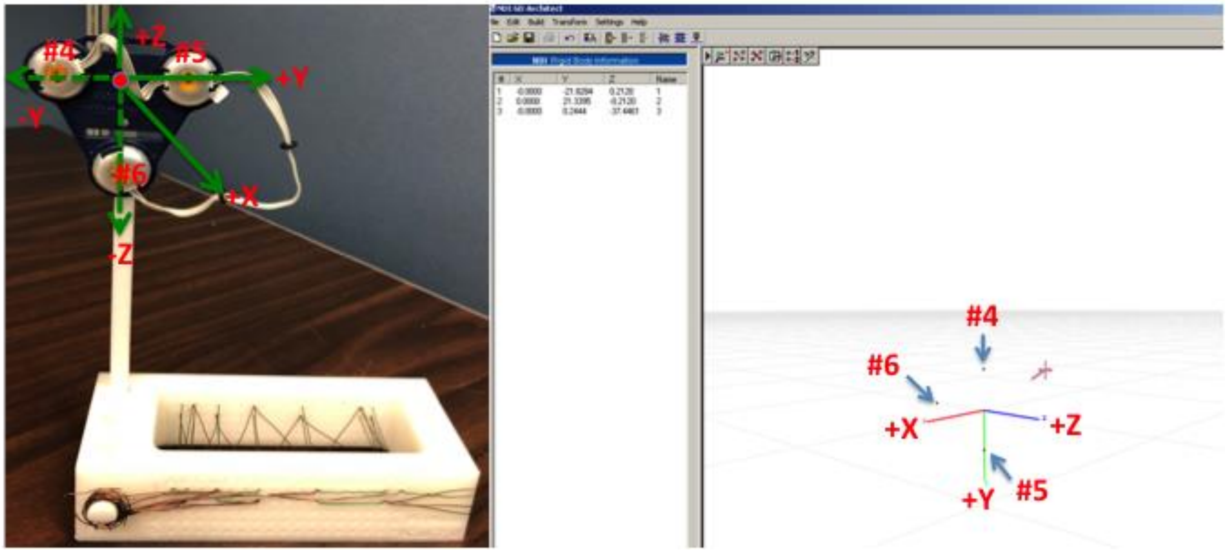


Figure 47- Wired fCal2.0 phantom

Originally, in [136]-[137], the fCal2.0 phantom is designed in its structure to be calibrated using an EM tracker or SonixGPS (BK Ultrasound, Richmond, BC) by plugging the sensors into the cubic (with 5mm/10mm cross section side length) or cylindrical (with 5mm/10mm cross section diameter) holes provided on each side of the fCal2.0 phantom as shown in Figure 47. However, in this thesis since the US transducer probe is tracked using the Optotracker, it is preferable to calibrate the phantom also using the Optotracker with respect to the world reference frame, which was configured using Optotracker in the previous section. The Optotracker sensors however cannot be attached to the body of the phantom since the phantom is going to be submerged into the water, which can defect the sensors. To remedy this limitation, a rectangular cube of 150 mm in length and $10\ \mathrm{mm} \times 10\ \mathrm{mm}$ square cross section was fabricated using the 3D printer to be fitted onto the top left corner cubic hole of 10 mm sides square cross section and it was glued securely using UHU Polymer super glue. Finally, a triangular frame of three Optotracker sensors was secured on the rectangular cube attachment as shown in Figure 48. A phantom coordinate frame using these three sensors is configured using the Optotracker's NDI First Principle and NDI Architecture software packages, where –y axis passes through sensor #4, +y axis passes through sensor #5, and -z axis passes through sensor #6. The origin of the phantom coordinate frame lays midway between sensors

#4 and #5 on the triangular frame and the +x axis is perpendicular to this triangular frame as shown in Figure 48.



NDI Rigid Body Information

#	X	Y	Z	Name
1	-0.0000	-21.8294	0.2120	1
2	0.0000	21.3395	-0.2120	2
3	-0.0000	0.2444	-37.4461	3

Figure 48- Configuring the phantom coordinate frame

The five rows (i.e. planes) of six N-fiducials are therefore parallel to the x-y plane of the phantom as shown in Figure 49.

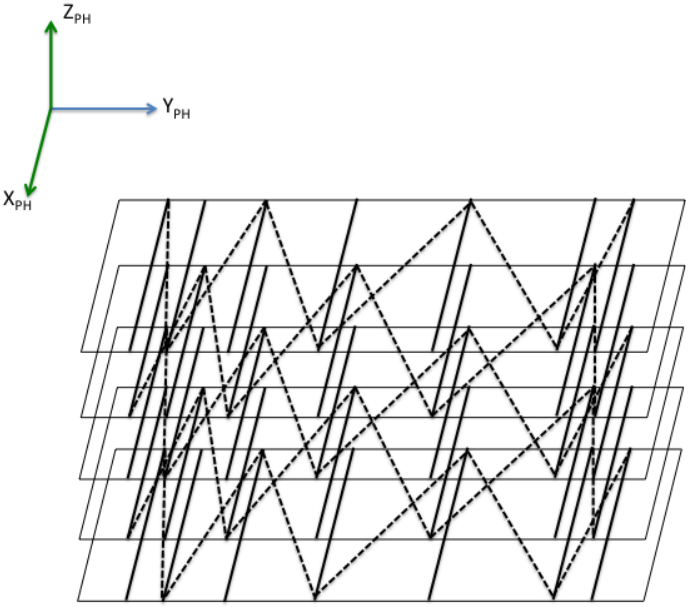


Figure 49- N-fiducials are parallel to the phantom's coordinate frame *x-y* plane

In an US image oriented approximately parallel to the z-axis of the phantom (Figure 50), an N-fiducial appears as a set of three points (ellipses), as illustrated in Figure 51, which shows the phantom as imaged through its upper face. Figure 52 illustrates the actual US image obtained from this experiment.

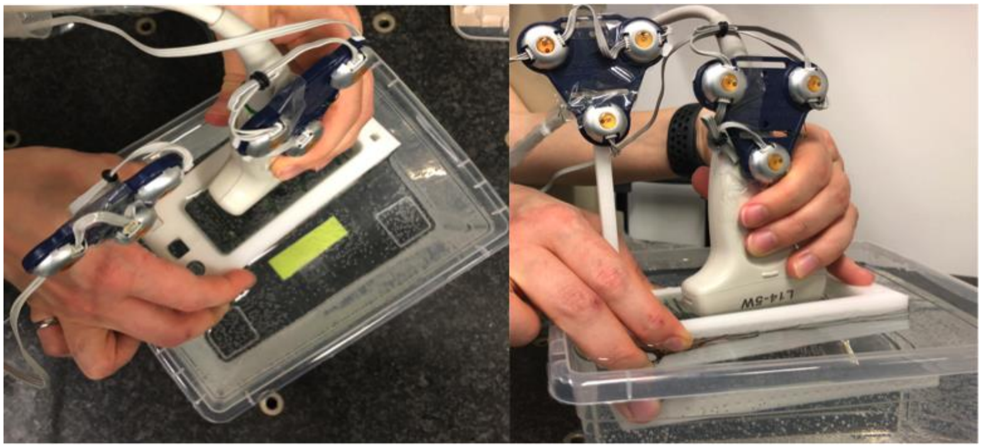


Figure 50- Calibration phantom experiments

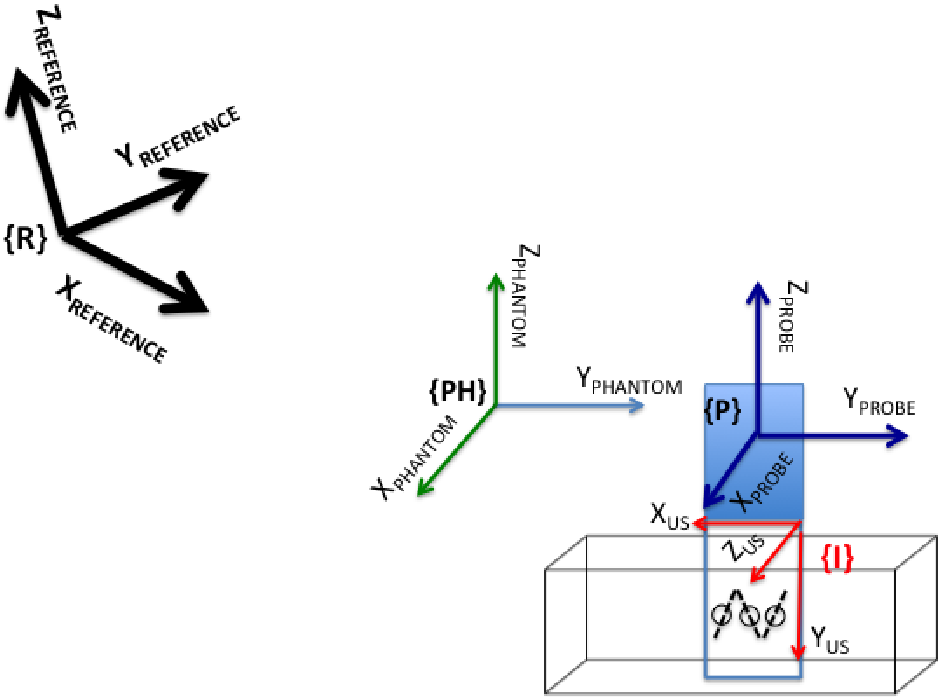
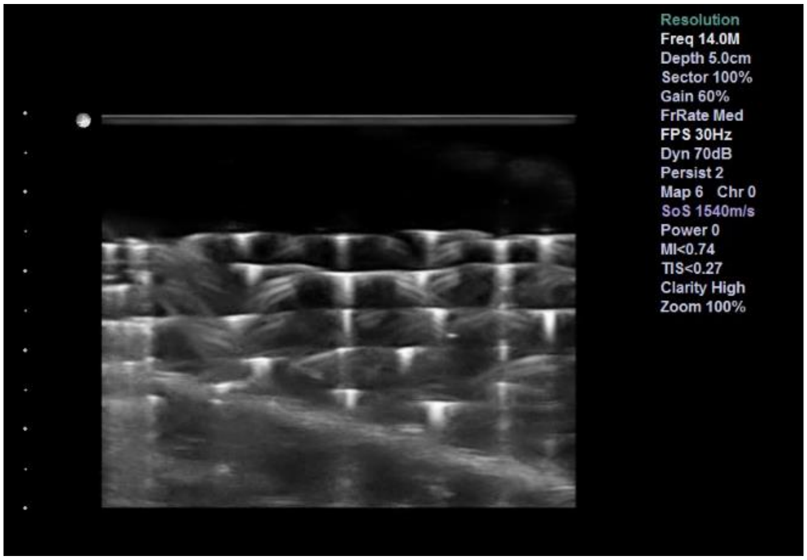


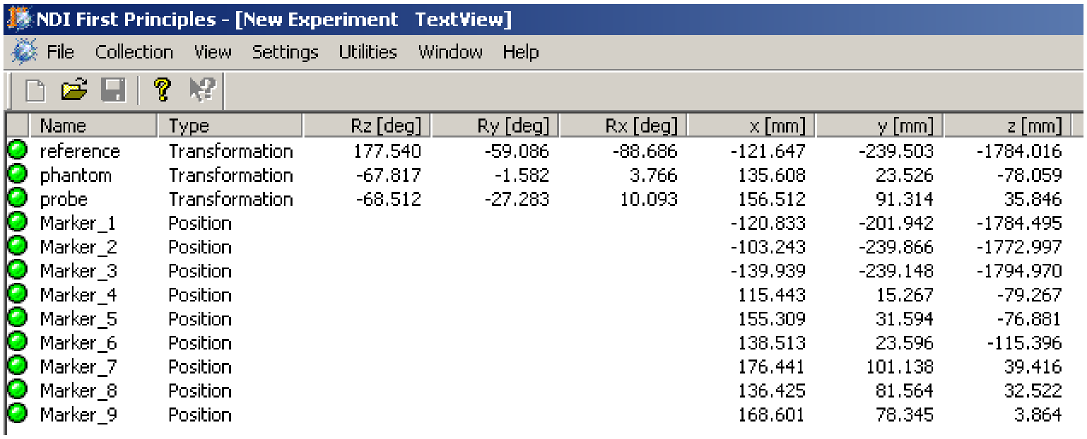
Figure 51- the geometric configuration of the four coordinate systems involved in the calibration procedure. The intersection of the US plane with one N-fiducial is illustrated as a set of three ellipses



Resolution
Freq 14.0M
Depth 5.0cm
Sector 100%
Gain 60%
FrRate Med
FPS 30Hz
Dyn 70dB
Persist 2
Map 6 Chr 0
SoS 1540m/s
Power 0
MI<0.74
TIS<0.27
Clarity High
Zoom 100%

Figure 52- The US image obtained from the wired fCal2.0 phantom and used to calibrate the US image frame

The NDI Optotracker is set up so that it provides the position and orientation of the US probe with respect to the world reference frame, $\xi_{\{P\}}^{\{R\}}$, and the phantom with respect to the world reference, $\xi_{\{PH\}}^{\{R\}}$. Also, as shown in Figure 67, the NDI Optotracker also provide the 3D position, (x, y, z), of each of the sensors (markers #1-9) and the position and orientation of the world reference frame with respect to the internally defined global frame.



	Name	Type	Rz [deg]	Ry [deg]	Rx [deg]	x [mm]	y [mm]	z [mm]
●	reference	Transformation	177.540	-59.086	-88.686	-121.647	-239.503	-1784.016
●	phantom	Transformation	-67.817	-1.582	3.766	135.608	23.526	-78.059
●	probe	Transformation	-68.512	-27.283	10.093	156.512	91.314	35.846
●	Marker_1	Position				-120.833	-201.942	-1784.495
●	Marker_2	Position				-103.243	-239.866	-1772.997
●	Marker_3	Position				-139.939	-239.148	-1794.970
●	Marker_4	Position				115.443	15.267	-79.267
●	Marker_5	Position				155.309	31.594	-76.881
●	Marker_6	Position				138.513	23.596	-115.396
●	Marker_7	Position				176.441	101.138	39.416
●	Marker_8	Position				136.425	81.564	32.522
●	Marker_9	Position				168.601	78.345	3.864

Figure 53- Screen shot of Optotracker readings used to calibrate the US frame

As shown in Figure 51, four coordinate systems are relevant to the description of the calibration method. These coordinate systems are illustrated in Figure 54 along with the various transformations between them.

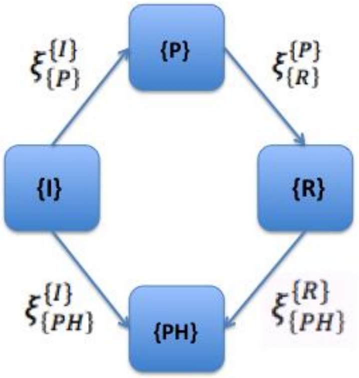


Figure 54- Coordinate transformations between various coordinate systems

The objective of the calibration is to determine $\xi_{\{P\}}^{\{I\}}$, which is the transformation from the US image-based coordinate systems to the US probe coordinate system. Spatial relationships can be written by composing relative poses and $\oplus/\ominus$ (pose composition operator/inverse of a pose) operators as follow:

$$\xi_{\{PH\}}^{\{I\}} = \xi_{\{P\}}^{\{I\}} \oplus \xi_{\{R\}}^{\{P\}} \oplus \xi_{\{PH\}}^{\{R\}}. \qquad (4.2)$$

A concrete representation of relative pose ξ is $\xi \sim \mathbf{T} \in SE\ (3)$. Also given properties,

$$\mathbf{T}_1 \oplus \mathbf{T}_2 \rightarrow \mathbf{T}_1\mathbf{T}_2, \qquad (4.3)$$

which is a standard matrix multiplication and

$$\ominus\, \mathbf{T} \rightarrow \mathbf{T}^{-1} \qquad (4.4)$$

Equation 4.2 can be rewritten as:

$$\mathbf{T}_{\{PH\}}^{\{I\}} = \mathbf{T}_{\{P\}}^{\{I\}}\mathbf{T}_{\{R\}}^{\{P\}}\mathbf{T}_{\{PH\}}^{\{R\}}. \tag{4.5}$$

$\mathbf{T}_{\{\mathrm{P}\}}^{\{\mathrm{I}\}}$ is extracted out of equation 4.5 since it corresponds to $\xi_{\{P\}}^{\{I\}}$ which is the transformation of interest in this section:

$$\mathbf{T}_{\{\mathrm{PH}\}}^{\{\mathrm{I}\}}\left(\mathbf{T}_{\{\mathrm{R}\}}^{\{\mathrm{P}\}}\mathbf{T}_{\{\mathrm{PH}\}}^{\{\mathrm{R}\}}\right)^{-1} = \mathbf{T}_{\{\mathrm{P}\}}^{\{\mathrm{I}\}}\left(\mathbf{T}_{\{\mathrm{R}\}}^{\{\mathrm{P}\}}\mathbf{T}_{\{\mathrm{PH}\}}^{\{\mathrm{R}\}}\right)\left(\mathbf{T}_{\{\mathrm{R}\}}^{\{\mathrm{P}\}}\mathbf{T}_{\{\mathrm{PH}\}}^{\{\mathrm{R}\}}\right)^{-1} \text{ Leading to:}$$

$$\mathbf{T}_{\{\mathrm{P}\}}^{\{\mathrm{I}\}} = \mathbf{T}_{\{\mathrm{PH}\}}^{\{\mathrm{I}\}}\left(\mathbf{T}_{\{\mathrm{R}\}}^{\{\mathrm{P}\}}\mathbf{T}_{\{\mathrm{PH}\}}^{\{\mathrm{R}\}}\right)^{-1} \tag{4.6}$$

Here $\mathbf{T}_{\{\mathrm{PH}\}}^{\{\mathrm{R}\}}$, which corresponds to the pose of the phantom coordinate frame with respect to the world reference frame (i.e. $\xi_{\{PH\}}^{\{R\}}$), can be derived by the 6DOF pose components (x, y, z, Rx, Ry, Rz) provided by the output of the NDI Optotracker as shown in Figure 53 in front of the word "phantom". For the specific experiment shown in Figure 50, which resulted in the US image of Figure 52, the $T_{\{PH\}}^{\{R\}}$ can be obtained.

$$\mathbf{T}_{\{\mathrm{PH}\}}^{\{\mathrm{R}\}} = \begin{bmatrix} 0.5526 & 0.8335 & -0.0031 & 142.2180 \\ -0.8142 & 0.5406 & 0.2116 & 115.8248 \\ 0.1781 & -0.1144 & 0.9773 & -35.9345 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Similarly, $\mathbf{T}_{\{R\}}^{\{P\}}$ noted equation 4.6 can be calculated using 6DOF readings of the position and orientation of the probe frame with respect to the world reference frame provided by the Optotracker. Using equation 4.1 and considering the relation given by:

$$\left(\mathbf{T}_{\{P\}}^{\{R\}}\right)^{-1} = \mathbf{T}_{\{R\}}^{\{P\}} \tag{4.7}$$

$$\mathbf{T}_{\{R\}}^{\{P\}} = \begin{bmatrix} 0.5153 & -0.8403 & 0.1686 & 1.0532 \\ 0.8488 & 0.4732 & -0.2361 & -258.9608 \\ 0.1186 & 0.2647 & 0.9570 & 5.8358 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

For the transformation $\mathbf{T}_{\{PH\}}^{\{I\}}$ (i.e. the transformation of the phantom coordinate frame, {PH}, with respect to the US image coordinate frame, {I}), homologous points provided by the N-fiducials can be used [136]. In an US image approximately transverse to a fiducial, the "N" will be visualized as a set of three ellipses whose centers are collinear (Figure 51). Because each N-fiducial is parallel to the x-y plane of the phantom coordinate system, all three ellipse centers must have the same z coordinate in the phantom space. For the middle ellipse, the x and y phantom space coordinates of its centers can be determined based on the similar triangles BEF and FGD in Figure 55. The phantom coordinates of the points of the intersection of the US imaging plane with an N-fiducial can then be determined by

$$x_F^{\{PH\}} = x_B^{\{PH\}} + a\left(x_C^{\{PH\}} - x_B^{\{PH\}}\right) \tag{4.8}$$

$$y_F^{\{PH\}} = y_B^{\{PH\}} + a\left(y_C^{\{PH\}} - y_B^{\{PH\}}\right) \tag{4.9}$$

The ratio $a = \frac{FE}{EG}$ can be measured using the locations of the three ellipses in the US image (Figure 56), and the coordinates of the vertices (i.e. $x_B^{\{PH\}}$, $x_C^{\{PH\}}$, $y_B^{\{PH\}}$, $y_C^{\{PH\}}$) in the phantom space {PH} are known based on the phantom design and measured using CMM as shown in Figure 58. Therefore, for each N-fiducial, the middle ellipse provides a pair of homologous points with known coordinates in the US ($z_{US} = 0$ for all points in the US image) and phantom coordinate frames.

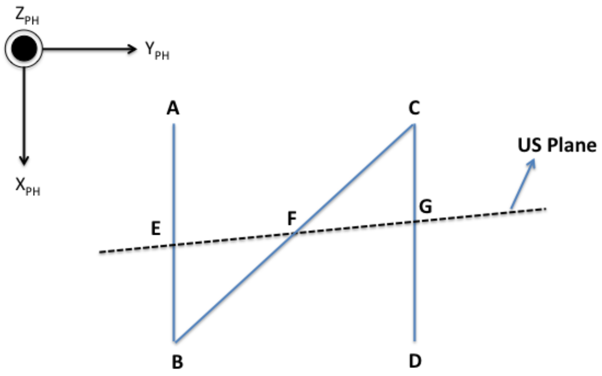


Figure 55- A top view of an US plane intersecting a single N-fiducial. The dashed line represents the US imaging plane

The US plane that is transverse to the N fiducial and shown in a dashed line in Figure 55 is shown in Figure 56. In this figure the US transverse image is shown in appropriate units of millimeters where the points E_1, F_1 and G_1 (Corresponding to points E, F, G in Figure 69) are measured.

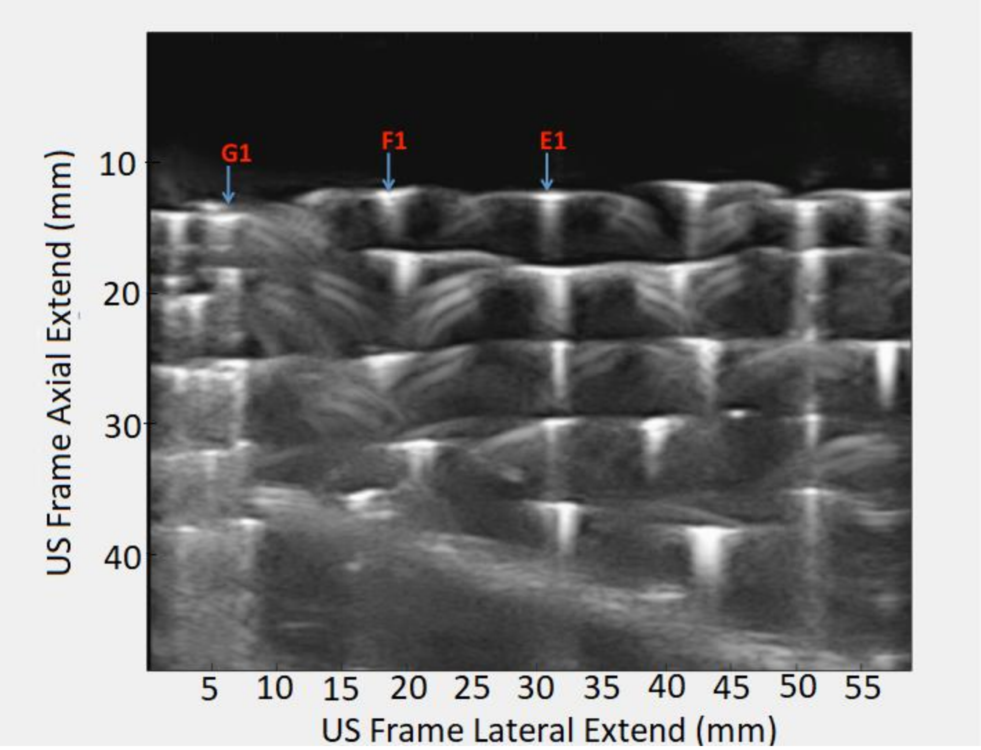
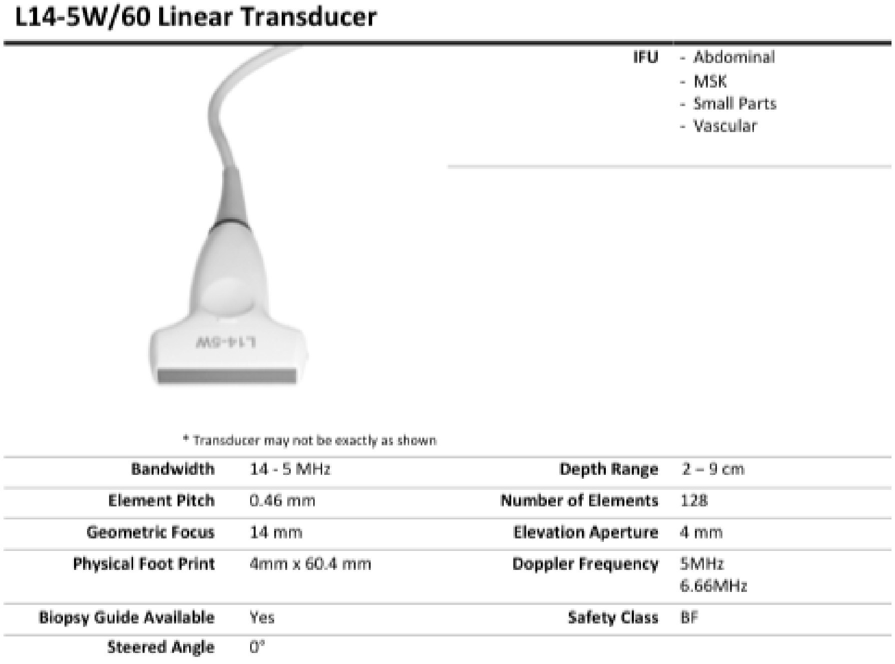


Figure 56- US transverse image of the N-fiducial phantom at 5 cm imaging depth

The code to convert and resize the pixel-based image to millimeter-based image is provided in Appendix B.5. To appropriately produce the conversion from pixels to millimeters, the information of the specific transducer used must be retrieved specifically for “Element Pitch” and “Element Number”. In this thesis, the linear transducer L14-5W/60 is used to carry out the experiments and Figure 57 summarizes this transducer’s specifications. The element pitch of this transducer is 0.46 mm and this transducer has 128 elements in total. The Lateral length of the US image produced by the L14-5W/60 transducer can therefore calculated to be:

$$\text{Total Lateral Length (in mm)} = \text{Total \# of Elements} \times \text{Pitch per Element} = 128 \times 0.46 = 58.8800 \text{ mm}$$

Dividing the calculated total lateral distance (i.e. $58.8800\ \mathrm{mm}$) by the total number of pixels in the lateral direction of transverse US image provides the amount in millimeters each pixel represents. To accurately retrieve the total number of lateral pixels in the saved US image, the default header file must be retrieved.



L14-5W/60 Linear Transducer

IFU
- Abdominal
- MSK
- Small Parts
- Vascular

* Transducer may not be exactly as shown

Bandwidth	14 - 5 MHz	**Depth Range**	2 – 9 cm
Element Pitch	0.46 mm	**Number of Elements**	128
Geometric Focus	14 mm	**Elevation Aperture**	4 mm
Physical Foot Print	4mm x 60.4 mm	**Doppler Frequency**	5MHz 6.66MHz
Biopsy Guide Available	Yes	**Safety Class**	BF
Steered Angle	0°		

Figure 57- L14-5W/60 transducer specifications

The total number of pixels in the lateral direction of the US image is 497, which leads to:

$$\frac{\text{Total Lateral Length}}{\text{Total Number of Lateral Pixels}} = \frac{58.8800\ \text{mm}}{497\ \text{pixels}} = 0.1185\ \text{mm/pixel}$$

Given a pixel to be square we can derive the axial length of the US image to be:

$$0.1185\,\frac{\text{mm}}{\text{pixel}} \times\ 416\ \text{total number of axial pixels} =\ \ 49.2839\ \text{mm axial length of US image}$$

Given the total lateral and axial lengths of the transverse US image (i.e. $58.8800\ \mathrm{mm}$ and $49.2839\ \mathrm{mm}$ respectively), the original pixel-based US image can be interpolated to the millimeter-based image of Figure 56.

The lateral distances $\mathrm{F_1E_1}$ and $\mathrm{E_1G_1}$ can be read off Figure 56 to be 12.47 mm and 24.83 mm respectively, which leads to $a_1 = \frac{\mathrm{F_1E_1}}{\mathrm{E_1G_1}} = 0.5022$.

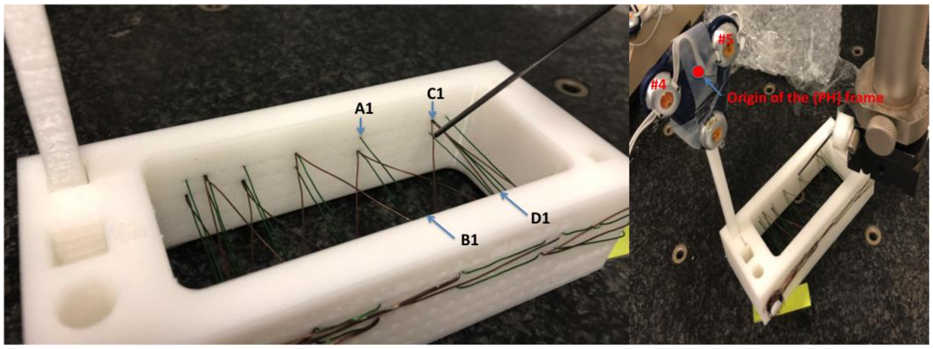


Figure 58 – Measuring points A and B in {PH} frame referring to Figure 69 and required by equations 4.8 and 4.9

In addition, using CMM, the following measurements (in mm) were obtained:

$$\mathrm{A1}^{\{\mathrm{PH}\}} = \begin{bmatrix} 18.631 \\ 72.492 \\ -142.212 \end{bmatrix} \mathrm{mm}\ ,\ \mathrm{B1}^{\{\mathrm{PH}\}} = \begin{bmatrix} 54.976 \\ 57.046 \\ -142.212 \end{bmatrix} \mathrm{mm}\ ,\ \mathrm{C1}^{\{\mathrm{PH}\}} = \begin{bmatrix} 28.988 \\ 95.621 \\ -142.212 \end{bmatrix} \mathrm{mm}\ ,\ \mathrm{D1}^{\{\mathrm{PH}\}} = \begin{bmatrix} 63.850 \\ 79.813 \\ -142.212 \end{bmatrix} \mathrm{mm}$$

By substituting these measurements into equations 4.8 and 4.9, $x_{F1}^{\{PH\}} = 41.9248\mathrm{mm}$ and $y_{F1}^{\{PH\}} = 76.4183\mathrm{mm}$ are obtained. Therefore, the point $\mathrm{F_1}$ in the phantom coordinate {PH} is given by:

$$\mathrm{F_1}^{\{\mathrm{PH}\}} = \begin{bmatrix} 41.9248 \\ 76.4183 \\ -142.212 \end{bmatrix} \mathrm{mm}$$

On the other hand, as measured in Figure 56 and considering that $\mathrm{z_{US}} = 0$, the point $\mathrm{F_1}$ in the US image frame coordinate is given by:

$$\mathrm{F}_1^{\{\mathrm{I}\}} = \begin{bmatrix} 18.94 \\ 12.62 \\ 0 \end{bmatrix} \mathrm{mm}$$

The points $\mathrm{F}^{\{\mathrm{PH}\}}$ and $\mathrm{F}^{\{\mathrm{I}\}}$ should be turned into their homogeneous forms by appending a one and the following will hold:

$$\tilde{\mathrm{F}}_1{}^{\{\mathrm{I}\}} = \mathbf{T}^{\{\mathrm{I}\}}_{\{\mathrm{PH}\}} \tilde{\mathrm{F}}_1{}^{\{\mathrm{PH}\}} \tag{4.10}$$

According to equation 4.10, $\mathrm{T}^{\{\mathrm{I}\}}_{\{\mathrm{PH}\}}$ is a homogeneous transformation transferring the point F1 from the calibration phantom coordinate, {PH}, to the US image coordinate frame {I}. Substituting the known homogeneous vectors into this equation:

$$\begin{bmatrix} 18.94 \\ 12.62 \\ 0 \\ 1 \end{bmatrix} = \mathbf{T}^{\{\mathrm{I}\}}_{\{\mathrm{PH}\}} \begin{bmatrix} 41.9248 \\ 76.4183 \\ -142.212 \\ 1 \end{bmatrix} \rightarrow \mathbf{T}^{\{\mathrm{I}\}}_{\{\mathrm{PH}\}} = \begin{bmatrix} \mathrm{m}_{11} & \mathrm{m}_{12} & \mathrm{m}_{13} & \mathrm{m}_{14} \\ \mathrm{m}_{21} & \mathrm{m}_{22} & \mathrm{m}_{23} & \mathrm{m}_{24} \\ \mathrm{m}_{31} & \mathrm{m}_{32} & \mathrm{m}_{33} & \mathrm{m}_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.11}$$

As shown in matrix 4.11, the 12 unknowns of the transformation matrix $\mathbf{T}^{\{I\}}_{\{PH\}}$ cannot be solved only with one point, F_1. To solve for this transformation matrix, nine more points (for the total of ten points) are measured experimentally in both {PH} and {I}, in the same manner as explained for measuring F_1. These ten points, $\mathrm{F}_1 \ldots \mathrm{F}_{10}$, are shown in the US frame, {I}, in Figure 59 where they can be directly measured. These ten points should also be measured in the phantom frame, {PH}, Table 5 summarizes these points measured in both frames. No more than ten points can be measured in the fCal 2.0 phantom using L14-5W/60 linear probe, given that fCal 2.0 only includes five rows as shown in Figure 57 and the ~60 mm lateral length of the linear probe can only capture 2 N-fiducials leading to an F point. Hence 5 rows x 2 F points = 10 points in total.

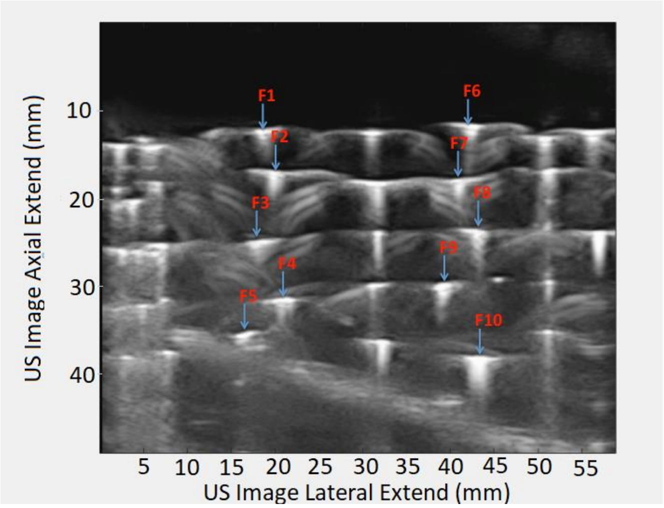


Figure 59- points $\mathrm{F}^{\{\mathrm{I}\}}_1 \ldots \mathrm{F}^{\{\mathrm{I}\}}_{10}$

Table 5
Measuring ten points in both {PH} and {I}

Point #	Measured in {I}	Ratio *a* (mm/mm)	Measured in {PH}
F_1	$\begin{bmatrix}18.94\\12.62\\0\end{bmatrix}$	0.5022	$\begin{bmatrix}41.9248\\76.4183\\-142.212\end{bmatrix}$
F_2	$\begin{bmatrix}19.95\\17.45\\0\end{bmatrix}$	0.4755	$\begin{bmatrix}40.1326\\75.9731\\-147.212\end{bmatrix}$
F_3	$\begin{bmatrix}18.15\\25.09\\0\end{bmatrix}$	0.5637	$\begin{bmatrix}40.3266\\78.7907\\-152.212\end{bmatrix}$
F_4	$\begin{bmatrix}21.41\\31.72\\0\end{bmatrix}$	0.4244	$\begin{bmatrix}37.8219\\75.5990\\-157.212\end{bmatrix}$
F_5	$\begin{bmatrix}16.80\\35.88\\0\end{bmatrix}$	0.6453	$\begin{bmatrix}38.2059\\81.9384\\-162.212\end{bmatrix}$
F_6	$\begin{bmatrix}42.65\\11.95\\0\end{bmatrix}$	0.4218	$\begin{bmatrix}29.8694\\55.0558\\-142.212\end{bmatrix}$
F_7	$\begin{bmatrix}41.19\\18.35\\0\end{bmatrix}$	0.5284	$\begin{bmatrix}32.0980\\56.4807\\-147.212\end{bmatrix}$
F_8	$\begin{bmatrix}43.43\\23.86\\0\end{bmatrix}$	0.2955	$\begin{bmatrix}24.3852\\54.6211\\-152.212\end{bmatrix}$
F_9	$\begin{bmatrix}39.39\\30.15\\0\end{bmatrix}$	0.6131	$\begin{bmatrix}29.6793\\59.3563\\-157.212\end{bmatrix}$
F_{10}	$\begin{bmatrix}43.32\\38.47\\0\end{bmatrix}$	0.5131	$\begin{bmatrix}33.8338\\55.3700\\-162.212\end{bmatrix}$

Finding the optimal rotation and translation between two sets of 3D points (i.e. one set belonging to the US frame, {I}, and the other set belonging to the phantom frame, {PH}) is the problem of finding the Euclidean or Rigid transformation between these two sets of points. In this thesis the method, "Corresponding Point Set Registration", discussed by Besl *et al.* [138] is followed to find the rigid transformation between $F^{\{I\}}$ and $F^{\{PH\}}$. In what follows the notations and equations from this paper are used to present the procedure involved.

Denoting $\mathbf{q}_\mathrm{R} = [q_0\ q_1\ q_2\ q_3]^T$ to be a four-vector unit quaternion representing the rotation between the two dataset, where $q_0 \geq 0$ and $q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1$, the 3x3 rotation matrix that maps $\mathrm{F}^{\{\mathrm{PH}\}}$ to $\mathrm{F}^{\{\mathrm{I}\}}$ can be derived as follow.

$$\mathbf{R}(\mathbf{q}_\mathrm{R}) = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1 q_2 - q_0 q_3) & 2(q_1 q_3 + q_0 q_2) \\ 2(q_1 q_2 + q_0 q_3) & q_0^2 + q_2^2 - q_1^2 - q_3^2 & 2(q_2 q_3 - q_0 q_1) \\ 2(q_1 q_3 - q_0 q_2) & 2(q_2 q_3 + q_0 q_1) & q_0^2 + q_3^2 - q_1^2 - q_2^2 \end{bmatrix} \tag{4.12}$$

Denoting $\mathbf{q}_\mathrm{T} = [q_4\ q_5\ q_6]^T$ to be the translation vector, the complete registration state vector $\mathbf{q}$ (which maps the set of points in {PH} onto the ones in {I}) is denoted $\mathbf{q} = [\mathbf{q}_\mathrm{R} | \mathbf{q}_\mathrm{T}]^T$. Let $\mathrm{F}^{\{\mathrm{PH}\}} = \{\mathrm{F}_\mathrm{i}^{\{\mathrm{PH}\}}\}$ be the measured data point in the phantom frame, {PH}, set to be aligned with point set $\mathrm{F}^{\{\mathrm{I}\}} = \{\mathrm{F}_\mathrm{i}^{\{\mathrm{I}\}}\}$ measured in the US image frame, {I}, where $\mathrm{N} = \mathrm{N}_{\mathrm{F}^{\{\mathrm{I}\}}} = \mathrm{N}_{\mathrm{F}^{\{\mathrm{PH}\}}} = 10$ and where each point $\mathrm{F}_\mathrm{i}^{\{\mathrm{PH}\}}$ corresponds to the point $\mathrm{F}_\mathrm{i}^{\{\mathrm{I}\}}$ with the same index as shown in Table 8. The mean square objective function to be minimized is given by the following equation.

$$f(\vec{q}) = \frac{1}{\mathrm{N}} \sum_{\mathrm{i}=1}^{\mathrm{N}} \left|\left| \mathrm{F}_\mathrm{i}^{\{\mathrm{I}\}} - \left(\mathbf{R}(\mathbf{q}_\mathrm{R}) \mathrm{F}_\mathrm{i}^{\{\mathrm{PH}\}} + \mathbf{q}_\mathrm{T} \right) \right|\right|^2 \tag{4.13}$$

In equation 4.13, $\left(\mathbf{R}(\mathbf{q}_\mathrm{R})\mathrm{F}_\mathrm{i}^{\{\mathrm{PH}\}} + \mathbf{q}_\mathrm{T}\right)$ maps the point $\mathrm{F}_\mathrm{i}^{\{\mathrm{PH}\}}$ from frame {PH} to the frame {I} using optimal rotation $\mathbf{R}(\mathbf{q}_\mathrm{R})$ and translation $\mathbf{q}_T$. After this mapping, the Euclidean distance between the two points is calculated[35]. In other words, equation 4.13 is nothing more than root mean square error between the points measure in {I} and the points mapped onto {I}.

The center of mass $\boldsymbol{\mu}_{\mathrm{F}^{\{\mathrm{PH}\}}}$ of the measured point set $\mathrm{F}^{\{\mathrm{PH}\}}$ and the center of mass $\boldsymbol{\mu}_{\mathrm{F}^{\{\mathrm{I}\}}}$ of the measured point set $\mathrm{F}^{\{\mathrm{I}\}}$ are given by (using the data in Table 8):

[35] The Euclidean distance between the two points $\mathbf{r}_1 = (x_1, y_1, z_1)$ and $\mathbf{r}_2 = (x_2, y_2, z_2)$ is $d(\mathbf{r}_1, \mathbf{r}_2) = ||\mathbf{r}_1 - \mathbf{r}_2|| = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$

$$\boldsymbol{\mu}_{F^{\{PH\}}} = \frac{1}{N}\sum_{i=1}^{N} F_i^{\{PH\}} = \begin{bmatrix} 34.8277 \\ 66.9603 \\ -152.2120 \end{bmatrix} \text{mm} \tag{4.14}$$

$$\boldsymbol{\mu}_{F^{\{I\}}} = \frac{1}{N}\sum_{i=1}^{N} F_i^{\{I\}} = \begin{bmatrix} 30.5230 \\ 24.5540 \\ 0 \end{bmatrix} \text{mm} \tag{4.15}$$

The cross-covariance matrix $\boldsymbol{\Sigma}_{F^{\{PH\}}F^{\{I\}}}$ of the sets $F^{\{PH\}}$ and $F^{\{I\}}$ is given by:

$$\boldsymbol{\Sigma}_{F^{\{PH\}}F^{\{I\}}} = \frac{1}{N}\sum_{i=1}^{N}[\left(F_i^{\{PH\}} - \boldsymbol{\mu}_{F^{\{PH\}}}\right)\left(F_i^{\{I\}} - \boldsymbol{\mu}_{F^{\{I\}}}\right)^T] \tag{4.16}$$

Using equations 4.14, 4.15 and Table 8, $\boldsymbol{\Sigma}_{F^{\{PH\}}F^{\{I\}}}$ is calculated to be:

$$\boldsymbol{\Sigma}_{F^{\{PH\}}F^{\{I\}}} = \begin{bmatrix} -56.2376 & -1.5682 & 0 \\ -126.6650 & 8.0770 & 0 \\ 1.6400 & -62.8150 & 0 \end{bmatrix}$$

The cyclic components of the anti-symmetric matrix,

$$\mathbf{A}_{ij} = \left(\boldsymbol{\Sigma}_{F^{\{PH\}}F^{\{I\}}} - \boldsymbol{\Sigma}_{F^{\{PH\}}F^{\{I\}}}^T\right)_{ij} \tag{4.17}$$

are used to form the column vector,

$$\Delta = [A_{23}\ A_{31}\ A_{12}]^T = [62.8150 \quad 1.6400 \quad 125.0968]^T. \tag{4.18}$$

Where the anti-symmetric matrix is calculated to be:

$$\mathbf{A} = \begin{bmatrix} 0 & 125.0968 & -1.6400 \\ -125.0968 & 0 & 62.8150 \\ 1.6400 & -62.8150 & 0 \end{bmatrix}$$

Subsequently, the symmetric 4x4 matrix $\mathbf{Q}(\boldsymbol{\Sigma}_{F^{\{PH\}}F^{\{I\}}})$ can be formed as follow.

$$\mathbf{Q}\left(\boldsymbol{\Sigma}_{F^{\{PH\}}F^{\{I\}}}\right) = \begin{bmatrix} tr\left(\boldsymbol{\Sigma}_{F^{\{PH\}}F^{\{I\}}}\right) & \Delta^T \\ \Delta & \boldsymbol{\Sigma}_{F^{\{PH\}}F^{\{I\}}} + \boldsymbol{\Sigma}_{F^{\{PH\}}F^{\{I\}}}^T - tr\left(\boldsymbol{\Sigma}_{F^{\{PH\}}F^{\{I\}}}\right)\mathbf{I_3} \end{bmatrix} \tag{4.19}$$

where $\mathbf{I_3}$ is the 3x3 identity matrix and $tr\left(\mathbf{\Sigma}_{\mathrm{F}^{\{\mathrm{PH}\}}\mathrm{F}^{\{\mathrm{I}\}}}\right)$ refers to the trace of the square matrix $\mathbf{\Sigma}_{\mathrm{F}^{\{\mathrm{PH}\}}\mathrm{F}^{\{\mathrm{I}\}}}$.

The calculated matrix $\mathbf{Q}\left(\mathbf{\Sigma}_{\mathrm{F}^{\{\mathrm{PH}\}}\mathrm{F}^{\{\mathrm{I}\}}}\right)$ is given by:

$$\mathbf{Q}\left(\mathbf{\Sigma}_{\mathrm{F}^{\{\mathrm{PH}\}}\mathrm{F}^{\{\mathrm{I}\}}}\right) = \begin{bmatrix} -48.1606 & 62.8150 & 1.6400 & 125.0968 \\ 62.8150 & -64.3145 & -128.2333 & 1.6400 \\ 1.6400 & -128.2333 & 64.3145 & -62.8150 \\ 125.0968 & 1.6400 & -62.8150 & 48.1606 \end{bmatrix}$$

The unit eigenvector $\mathbf{q}_{\mathrm{R}} = [q_0\ q_1\ q_2\ q_3]^T$ corresponding to the maximum eigenvalue of the matrix $\mathbf{Q}\left(\mathbf{\Sigma}_{\mathrm{F}^{\{\mathrm{PH}\}}\mathrm{F}^{\{\mathrm{I}\}}}\right)$ is selected as the optimal rotation.

Using MATLAB, the eigenvector corresponding to the biggest eigenvalue of the matrix $\mathbf{Q}\left(\mathbf{\Sigma}_{\mathrm{F}^{\{\mathrm{PH}\}}\mathrm{F}^{\{\mathrm{I}\}}}\right)$ is:

$$\mathbf{q}_{\mathrm{R}} = [q_0\ q_1\ q_2\ q_3]^T = [0.3776 \quad 0.3915 \quad -0.6198 \quad 0.5656\ ]^T$$

Substituting the unit quaternion rotation vector presented above into the matrix, $\mathbf{R}(\mathbf{q}_R)$, in 4.12, the 3x3 rotation matrix that maps the measured points in {PH} onto {I} is given by:

$$\mathbf{R}_{\{\mathrm{PH}\}}^{\{\mathrm{I}\}} = \mathbf{R}(\mathbf{q}_{\mathrm{R}}) = \begin{bmatrix} -0.4082 & -0.9125 & -0.0252 \\ -0.0581 & 0.0535 & -0.9969 \\ 0.9110 & -0.4055 & -0.0749 \end{bmatrix}$$

Finally, the optimal translation vector can be obtained by the following equation:

$$\mathbf{t}_{\{\mathrm{PH}\}}^{\{\mathrm{I}\}} = \mathbf{q}_{\mathrm{T}} = \boldsymbol{\mu}_{\mathrm{F}^{\{\mathrm{I}\}}} - \mathbf{R}(\mathbf{q}_{\mathrm{R}})\boldsymbol{\mu}_{\mathrm{F}^{\{\mathrm{PH}\}}} = \begin{bmatrix} 102.0074 \\ -128.7426 \\ -15.9775 \end{bmatrix} \tag{4.20}$$

In addition, using the calculated rotation matrix and translation vector, the 4x4 transformation matrix in equation 4.10 can be obtained to be:

$$\mathbf{T}_{\{PH\}}^{\{I\}} = \begin{bmatrix} \mathbf{R}_{\{PH\}}^{\{I\}} & \mathbf{t}_{\{PH\}}^{\{I\}} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} -0.4082 & -0.9125 & -0.0252 & 102.0074 \\ -0.0581 & 0.0535 & -0.9969 & -128.7426 \\ 0.9110 & -0.4055 & -0.0749 & -15.9775 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.21)$$

The last step is to find the root mean square error of the transformation matrix presented in 4.21 using the mean square objective function in equation 4.13.

$$\mathrm{RMSE} = \frac{1}{\mathrm{N}}\sum_{\mathrm{i}=1}^{\mathrm{N}} \left\| \mathrm{F}_{\mathrm{i}}^{\{\mathrm{I}\}} - \left(\mathbf{R}(\mathbf{q}_{\mathrm{R}})\mathrm{F}_{\mathrm{i}}^{\{\mathrm{PH}\}} + \mathbf{q}_{\mathrm{T}}\right) \right\|^2 = 7.6223 \text{ mm}$$

To estimate a more accurate rigid transformation matrix, the goal must be to minimize the RMSE value obtained through equation 4.13. The more points measured in both frames {PH} and {I}, the more accurate the estimation of 3D rigid transformation will be and hence the lower RMSE value can be achieved. However, as explained before in experiments conducted using fCal2.0 and L14-5W/60 linear US probe, no more than ten points are measurable given the small size of fCal2.0 phantom. To be able to measure more points using N-fiducials, the use of fCal3.1 phantom is recommended[36]. This phantom is bigger in size than fCal2.0 phantom (76 mm in width, 220 mm in length and 150 mm in height) and incorporates a total of 238 holes in 14 rows and 17 columns, which can be used to incorporate more N-fiducials.

The MATLAB code to calculate the optimal 3D rigid transformation between $\mathrm{F}^{\{\mathrm{I}\}}$ and $\mathrm{F}^{\{\mathrm{PH}\}}$ is provided in Appendix B.5.

After estimating $\mathbf{T}_{\{PH\}}^{\{I\}}$ in equation 4.21, this matrix can be substituted in equation 4.6, along with calculation of $\mathbf{T}_{\{PH\}}^{\{R\}}$ and $\mathbf{T}_{\{R\}}^{\{P\}}$ from Tables 6 and 7 respectively, to obtain the transformation matrix that relates the US probe frame to the US image frame as follow.

[36] The .stl file of fCal3.1 can be downloaded from: http://perk-software.cs.queensu.ca/plus/doc/nightly/modelcatalog/printable/fCal_3.1.stl

$$\mathbf{T}_{\{P\}}^{\{I\}} = \mathbf{T}_{\{PH\}}^{\{I\}}\left(\mathbf{T}_{\{R\}}^{\{P\}}\mathbf{T}_{\{PH\}}^{\{R\}}\right)^{-1} = \begin{bmatrix} -0.3672 & -0.9172 & -0.1542 & 25.5202 \\ -0.0457 & 0.1834 & -0.9820 & -97.6940 \\ 0.9290 & -0.3535 & -0.1094 & -13.4112 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.22}$$

4.6. Calibration Error

In addition to measuring the pose of the frames against one another, The NDI Optotracker also reports the estimated error in its relative pose readings. In the calibration experiments of this thesis, the NDI Optotracker First Principle software console reports 0.0767 mm estimated error in sensing the pose of the US probe frame, {P}, with respect to the reference frame, {R}. In addition, the estimated error in sensing the pose of the fCal2.0 phantom frame, {PH}, with respect to {R} is reported to be 0.3245 mm. The reference frame, {R}, itself is reported to have 0.4134 mm estimated error in its sensed pose with respect to the internal global frame.

On the other hand, the estimated transformation between the US probe frame and US image frame ($\mathbf{T}_{\{P\}}^{\{I\}}$) was calculated to have RMSE error of 7.6223 mm. To elaborate on this inherent error, it should be noted that the corresponding point set registration estimation technique proposed by Best et al. [138] is not expected to produce an accurate transformation matrix. The source of this inaccuracy, according to [136], is the challenge inherent in making sure that features used for calibration (point targets) are centered with respect to the thickness (elevational resolution) of the US image. This centering is very important, especially in the near-field zone where the US beam thickness is large. Noise and outliers are also other sources of inaccuracy. For example, inaccuracies in phantom construction (affecting the localization of the N-fiducials in the phantom coordinate space) and identification of the N-fiducial ellipses in the US images (affecting the coordinates of the middle ellipse of each N-fiducial in the US and phantom coordinate space). In addition, the inherent deficiency of this method is that it attempts to estimate the unknown non-rigid spatial transformation in the context of rigid transformation by considering the alignment of two point-sets as a probability density estimation problem [142]. That is why this technique comes with a cost function. This cost function, f(q) in eq 4.13, is used to find the RMSE associated with the estimated transformation. The goal of optimization must be to reduce the RMSE error and this is accomplished by accommodating more point measurements. In other words, more accurate and precise calibrations can be achieved provided that a large number of N-fiducials are included in one US image. This can be achieved by either constructing a separate phantom (e.g. fCal3.1) or providing multiple sections with different density of N-fiducials in a single phantom. The exact number of points required to make an accurate estimation can only be determined by conducting more experiments within the calibration experimental protocols discussed in this thesis. For example, in [136], through acquiring a large (30-40) number of images it was determined that calibration matrix obtained for US imaging depth of 16 cm (12 feature points) provides both more precision and accurate calibrations than calibration matrix obtained for US imaging depth of 9 cm (8 points) simply due to the fact that at larger imaging depth settings, a larger US field of view is acquired and thus more feature points can be contained in the US image; therefore, a more representative sampling of the US plane is possible.

The aforementioned calibration errors may offset the 3D shape estimation results of the catheters, which will be presented in chapter 5 in comparison with the direct CMM measurements of the same catheters presented as ground truth in Figures 42 and 43.

4.7. Calibration of Ultrasound Image Frame using Direct Structural Measurements

In section 4.5, the transformation matrix between the US probe and US frame was estimated using N-wire fast calibration method proposed by Pagoulatos *et al.* [136] and quaternion-based point set registration algorithm discussed by Besl *et al.* [138]. In this section this transformation is calculated directly by measuring the US image frame position and orientation with respect to the US probe reference frame position and orientation. This is illustrated in Figure 60.

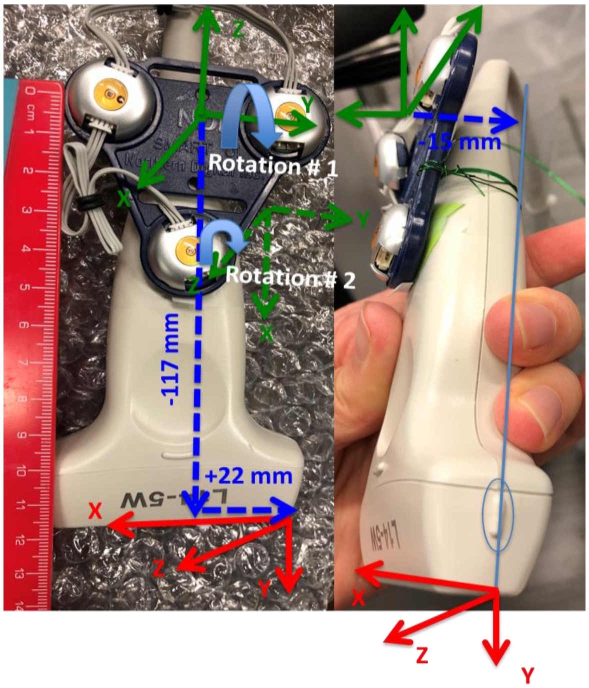


Figure 60 – Steps involved in manual calibration of US image frame with respect to the US probe frame

As shown in Figure 60, by having an intuitive anticipation of the location and orientation of the US image, one can estimate the transition matrix using direct measurements. In other words, the US probe coordinate frame can match the position and orientation of the US image frame by translating it by -117 mm in the z-direction, +22 mm in the y-direction and -15 mm in the x-direction. In terms of orientation, the US probe frame will match the US image frame by first rotating it along the y-axis by +90° and second rotating the resulting frame about its z-axis by -90°. This translation is therefore derived to be:

$$\mathbf{T}_{\{P\}}^{\{I\}} = \left(\mathbf{T}_{\{I\}}^{\{P\}}\right)^{-1} = \begin{bmatrix} 0 & -1 & 0 & 22 \\ 0 & 0 & -1 & -117 \\ 1 & 0 & 0 & 15 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.23}$$

Figure 61 shows this transformation (equation 4.23) in MATLAB against the one obtained in section 4.5 (equation 4.22).

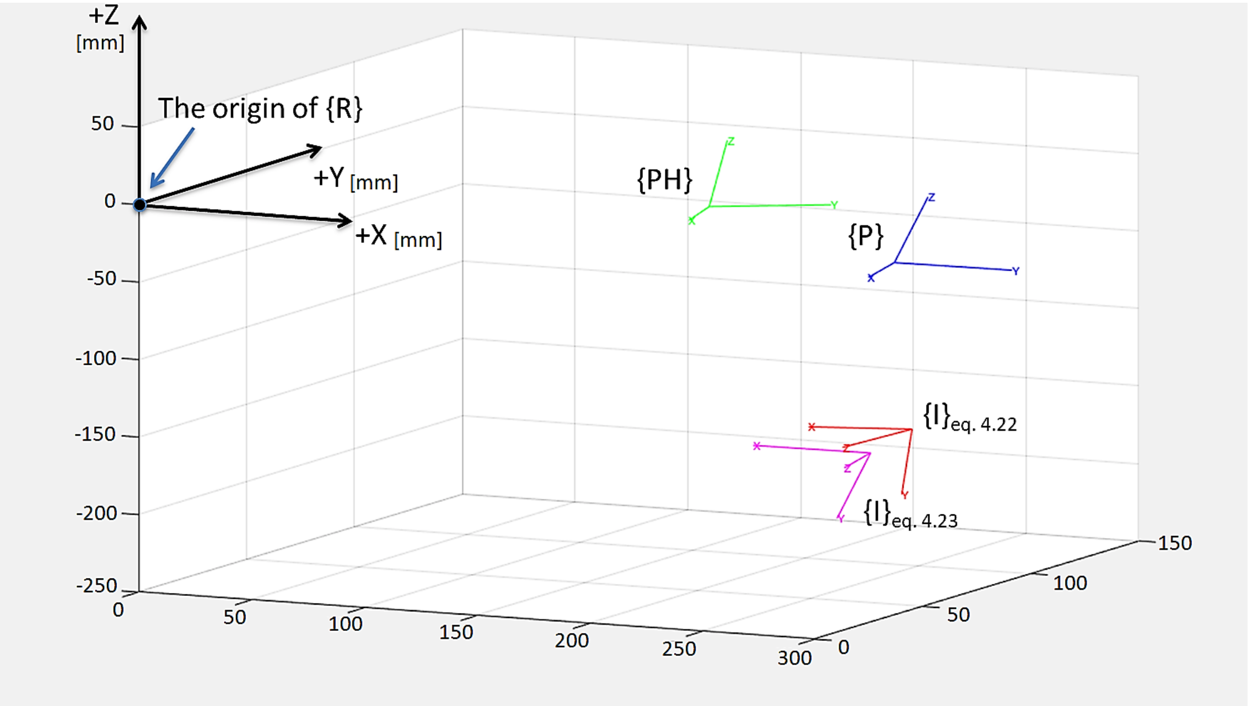
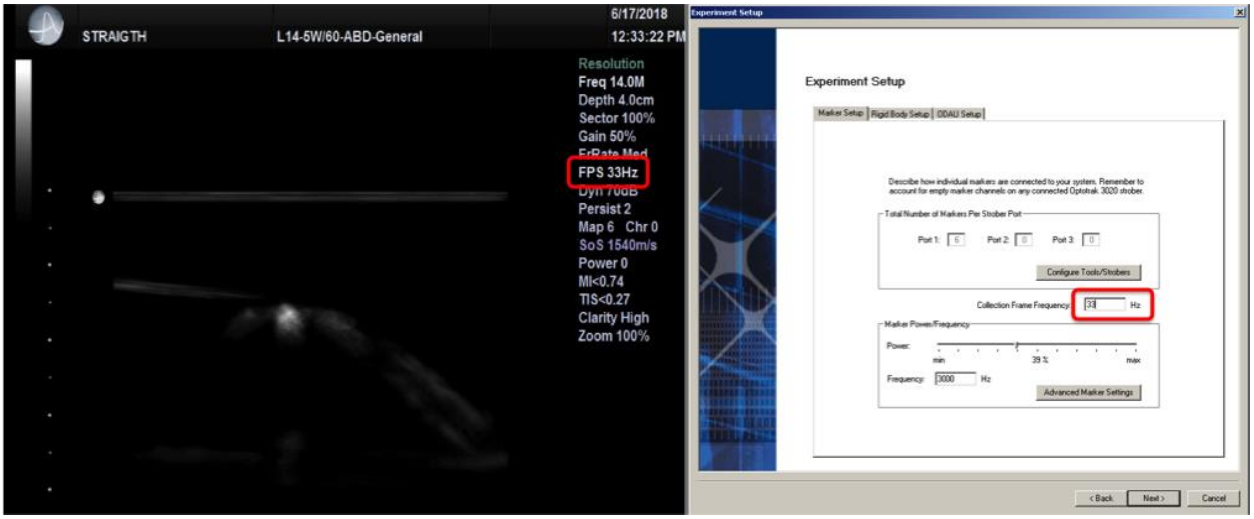


Figure 61- Manually calibrated US image frame {I} (in magenta) vs. estimated {I} (in red)

The problem with determining the transition matrix between the US image frame and probe frame using manual measurements as described in this section is that its correctness cannot be validated, and its accuracy cannot be quantified. However, it is still valuable to compare the calibration results using $\mathbf{T}_{\{I\}}^{\{P\}}$ obtained in this section with the ones obtained using $\mathbf{T}_{\{I\}}^{\{P\}}$ obtained in section 4.5 as it provides an observation of the reliability and feasibility of manual calibration. This issue is further discussed in chapter 5 where 3D shape estimation results of KF, AKF and PF are calibrated within the workspace using both versions of $\mathbf{T}_{\{I\}}^{\{P\}}$.

4.8. Syncing of Ultrasound Data with Optotracker Data

The purpose of using the NDI Optotracker with US experiments is to record the position of the US transducer probe in space at each point in time the US frames are acquired. This is to estimate the 3D shape of the catheter within a calibrated space that can be validated by Figures 34 and 35. Therefore, after the aforementioned calibration procedures, one can properly set the acquisition rate of the US frames and Optotracker readings so that they can be synced together correctly. For this purpose, the acquisition rate of both the US system and the NDI Optotracker are set to 33 Hz as shown in Figure 62. In addition while performing experiments special care was given to starting both machines at exactly the same time so that the data obtained at the same points in time can be matched and synced together. US machine stores the frames at 33 Hz in the .b8 file while the NDI Optotracker stores the pose readings at 33 Hz in an excel file. These data are matched and synced together in chapter 5 and the shape estimation algorithms of KF, AKF and PF are applied to them accordingly.



STRAIGTH
L14-5W/60-ABD-General
6/17/2018
12:33:22 PM

Resolution
Freq 14.0M
Depth 4.0cm
Sector 100%
Gain 50%
FrRate Med
FPS 33Hz
Dyn 70dB
Persist 2
Map 6 Chr 0
SoS 1540m/s
Power 0
MI<0.74
TIS<0.27
Clarity High
Zoom 100%

Experiment Setup

Experiment Setup

Describe how individual markers are connected to your system. Remember to account for empty marker channels on any connected Optotrak 3020 strober.

Total Number of Markers Per Strober Port
Port 1: 6 Port 2: 0 Port 3: 0

Collection Frame Frequency: 33 Hz

Marker Power/Frequency
Power: 39 %
Frequency: 3000 Hz

Figure 62 – Setting acquisition rate for both US system and NDI Optotracker to 33 FPS

In an experiment it was verified that in five seconds, the US system used for the experiments of this thesis obtains and records exactly 149 US image frames at 33 Hz. On the other hand, the NDI Optotracker obtain and records exactly 150 frames at 33 Hz. Therefore, the US lags behind the Optotracker in frame acquisition by an offset of 0.03 seconds. This offset is taken to account while syncing the two datasets together.

4.9. Summary

Summing up, four catheters with different configurations were 3D printed and secured within containers. The configuration of each of these catheters were directly measured using CMM as they were placed inside the CMM's workspace. These directly measured configurations as shown in Figures 34 and 35 are used as basis to validate the calibrated estimation results of KF, AKF, and PF. Therefore, the experiments were all carried out within the CMM workspace while keeping the containers at the same spot CMM measurements were obtained. Next the containers were filled with water-based gel and US transverse images were obtained by scanning across each of the four catheters. The US probe was tracked using NDI Optotracker with respect to the reference frame, that was set to be the same as the origin of the reference frame within which the CMM measurements of the catheters were obtained. Calibration procedures were carried out to find the transformation matrices that relate the US image frame, $\{I\}$, to the US probe frame, $\{P\}$, and to the reference frame, $\{R\}$. A calibration phantom was used to estimate the transformation of US probe frame, $\{P\}$, to the US image frame, $\{I\}$, $\mathbf{T}_{\{P\}}^{\{I\}}$, with RMSE of 7.6223 mm, by implementing N-wire fast calibration method proposed by Pagoulatos et al. [136] and quaternion-based point set registration algorithm discussed by Besl et al. [138]. $\mathbf{T}_{\{P\}}^{\{I\}}$ was also calculated by direct measurements between the US probe frame and image frame. However, the accuracy of $\mathbf{T}_{\{P\}}^{\{I\}}$ obtained manually in this way cannot be determined until the calibrated 3D shape estimation results of KF, AKF and PF using both versions of $\mathbf{T}_{\{P\}}^{\{I\}}$ is validated against the CMM measurements of Figures 34 and 35. Finally, the frame acquisition of both the NDI Optotracker and US system are set to be 33 Hz. This allows the both systems to produce same number of frames/sensor readings within a certain time with the US system lagging behind the NDI Optotracker by 0.03 seconds.

Chapter V

RESULTS AND VALIDATION

In this chapter, the KF, AKF and PF algorithms for 3D shape estimation of tendon-driven catheters are applied to the experimentally obtained US transverse images that are calibrated within CMM's workspace. At 33 Hz acquisition rate, the spatial pose data obtained by the NDI Optotracker is synced with frames obtained by the US system. The purpose of this chapter is to validate the calibrated 3D shape estimation results of the KF, AKF and PF against CMM direct measurements presented in Figures 34 and 35 and quantify their accuracy. In addition, the catheter's 3D shape estimation results are calibrated using both $\mathbf{T}_{\{P\}}^{\{I\}}$, one obtained by the calibration phantom and one by manual measurements. The two sets of results are then compared against one another with CMM's measurements of Figures 42 and 43 to evaluate the feasibility of manual calibration procedure of the US image frame with respect to the US probe frame.

In section 5.1, the US transverse B-mode images are processed to remove any artifacts and emphasize the cross section of the catheters within them. Section 5.2 demonstrates the position and orientation within which each of the US transverse images were obtained in the calibrated CMM workspace. In sections 5.3, 5.5, and 5.7 the cross section of the catheters is tracked within the experimentally obtained US transverse images using KF, AKF and PF algorithms respectively. Subsequently, in sections 5.4, 5.6 and 5.8, the KF, AKF and PF results are calibrated within the CMM workspace and validated against the direct measurements of figures 42 and 43. Finally in section 5.9, the accuracy of the shape estimation algorithms before and after the calibration are discussed. In addition, in this section the feasibility of manual calibration of the US image frame with respect to the US probe frame is evaluated by comparing the accuracy of calibrated 3D shape estimation results using the two methods (i.e., calibration by manual measurement vs. calibration by calibration phantom).

5.1. Ultrasound Image Processing

Before implementing any of the tracking algorithms (i.e., KF, AKF and PF), the grey-scale US images should be processed removing any artifacts and segmenting the region of interest containing the cross section of the 3D-printed catheters. The image processing procedure serving this purpose is explained in this section.

The water-based gel surrounding the 3D-printed catheters as shown in Figure 36, does not produce visible speckle noise in the US images, which simplifies the image processing and catheter cross section segmentation. The only US artifact present is the comet tail artifact (CTA), which occurs due to bouncing of the US wave between materials of different acoustic impedance such as the PLA catheter and the surrounding water [57]. This reverberation artifact is shown in Figures 13 and 45 to have a symmetrical tail shaped structure, the length of which depends on the bouncing echoes that are received by the transducer [57]. Fortunately the pixel grey-level intensity of the CTA in the US images is lower than that of catheter cross section and therefore it can be thresholded out of the image.

Figures 63 and 64 summarizes the segmentation of each of the four PLA catheters' cross-section using thresholding and blob analysis in MATLAB.

As shown in Figure 63, each US frame is first thresholded at grayscale[37] intensity level range of 120-130 resulting in a binary image[38] that contains a number of blobs. Each of these blobs is measured in MATLAB for its centroid, area, major axis length and minor axis length.

[37] For a grayscale images, the pixel value is a single number that represents the brightness of the pixel. Possible values range from 0 to 255, where zero is taken to be black, and 255 is taken to be white.

[38] Whereas in grayscale image the intensity values of the pixels vary from 0 to 255 (or 0 to 1), in Binary image the pixel value is either 0 or 1.

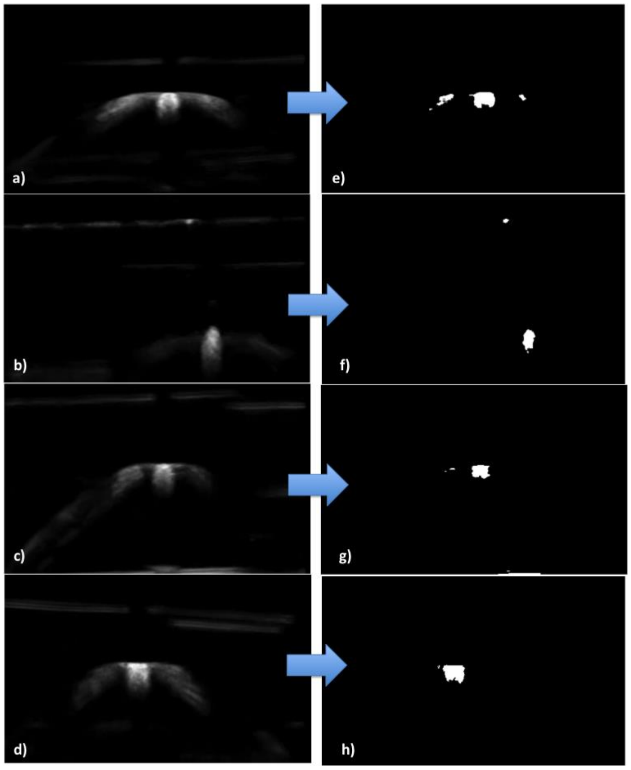


Figure 63 – US transverse image of a) straight catheter (thresholded at gray level intensity of 130 (e)), b) catheter #1 (thresholded at gray level intensity of 120 (f)), c) catheter #3 (thresholded at gray level intensity of 120 (g)), d) catheter #2 (thresholded at gray level intensity of 125 (h))

As shown in Figure 64, the blob with largest area represents the catheter cross section which is masked out of the image. Also, its major and minor axis lengths are measured and used as diameters of an equivalent ellipse for catheters with circular cross section (Figures 63a, 63b, 63c, 64a, 64b and 64c) and as sides of an equivalent rectangle for the catheter with square cross section (Figures 63d and 84d). Figures 64e, 64f, 64g and 64h demonstrate the detected catheter cross-section using an equivalent ellipse for the straight catheter, catheter #1 and #3[39] which have circular cross-section and using an equivalent rectangle for catheter #2 which has the square cross section.

[39] Catheter #1, #2, and #3 refer to the ones indicated in Figure 30

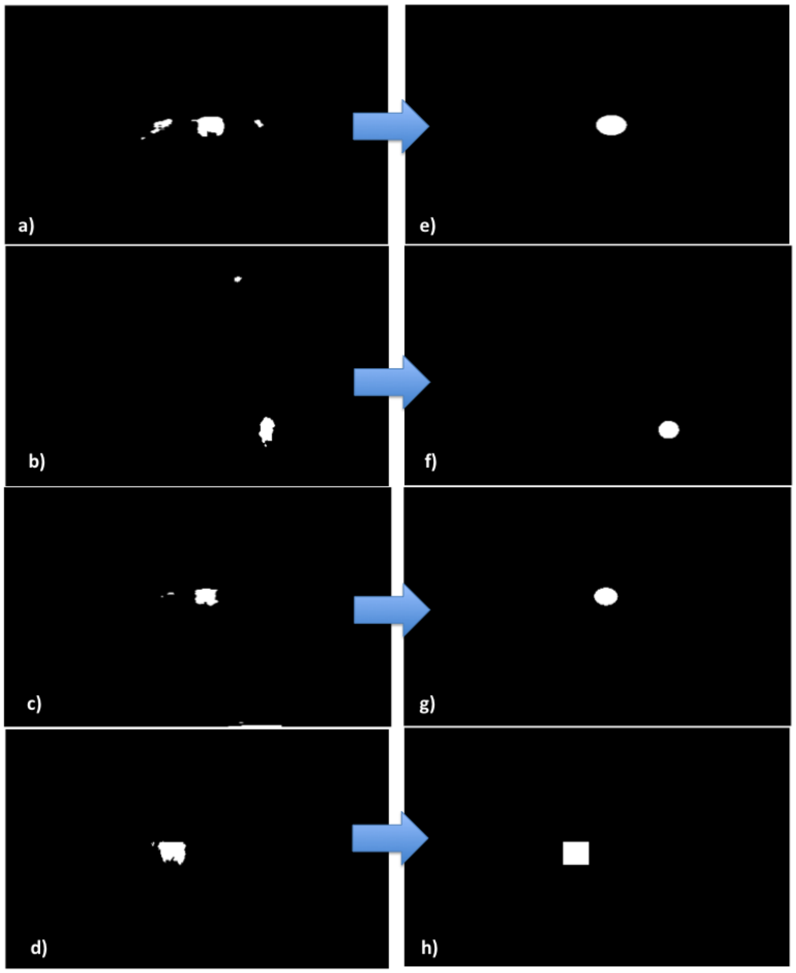


Figure 64 – a) Straight catheter cross section (along with its equivalent ellipse (e)), b) Catheter #1 cross section (along with its equivalent ellipse (f)), c) Catheter #3 cross section (along with its equivalent ellipse (g)), d) Catheter #2 cross section (along with its equivalent rectangle (h))

All of US transverse images taken from the four PLA 3D-printed catheters are processed into frames that look like Figures 64e, 64f, 64g and 64h upon which the tracking algorithms (i.e. KF, AKF and PF) are executed.

All the code for this section is provided in appendix B.6.

5.2. Ultrasound Probe Scanning Direction

To estimate the 3D shape of the catheters, US transverse images are obtained by scanning across the four catheters as shown in Figure 65. The position and orientation, (x, y, z, Rx, Ry, Rz), at which the US probe scans across the containers with respect to the predefined reference frame is recorded using NDI Optotracker from the three sensors attached to the US probe as shown. Using the NDI Optotracker sensor readings, the transformation between the US probe frame, {P}, and the reference frame, {R}, ($\mathbf{T}_{\{P\}}^{\{R\}}$ in Figure 66) can be calculated. In addition, the transformation between the US image frame, {I}, and the reference frame, {R}, ($\mathbf{T}_{\{I\}}^{\{R\}}$ in Figure 66) can be obtained using the following equation.

$$\mathbf{T}_{\{\mathbf{I}\}}^{\{R\}} = \mathbf{T}_{\{P\}}^{\{R\}}\mathbf{T}_{\{I\}}^{\{P\}}, \qquad (5.1)$$

where $\mathbf{T}_{\{I\}}^{\{P\}}$ can be obtained using equation 4.22 or 4.23.

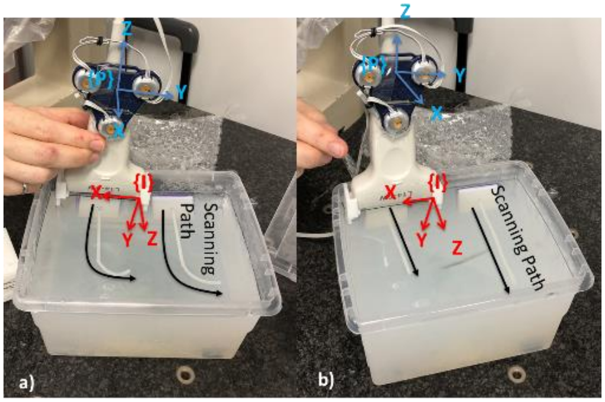


Figure 65 – The direction of US probe scan across the four catheters: a) un-planar (left), planar (right), b) straight (left) and curvy (right)

Using $\mathbf{T}_{\{I\}}^{\{R\}}$ and $\mathbf{T}_{\{P\}}^{\{R\}}$, the orientation and position of the US probe frame ({P}) and US image frame ({I}) while scanning across the four catheters can be plotted in MATLAB with respect to the reference frame ({R}) as shown in Figures 85 and 86.

The US probe scans across the planar and un-planar catheters with distal curves in a non-straight path. As the result, in the obtained sequence of US images, the catheter cross section appears to be in the middle of the US image frame throughout. In other words, when these transverse US images are stacked together in a cuboid (without incorporating their true position an orientation in space), they form a straight catheter. Therefore, to obtain the correct perspective of the KF, AKF and PF results, they should be calibrated within the workspace.

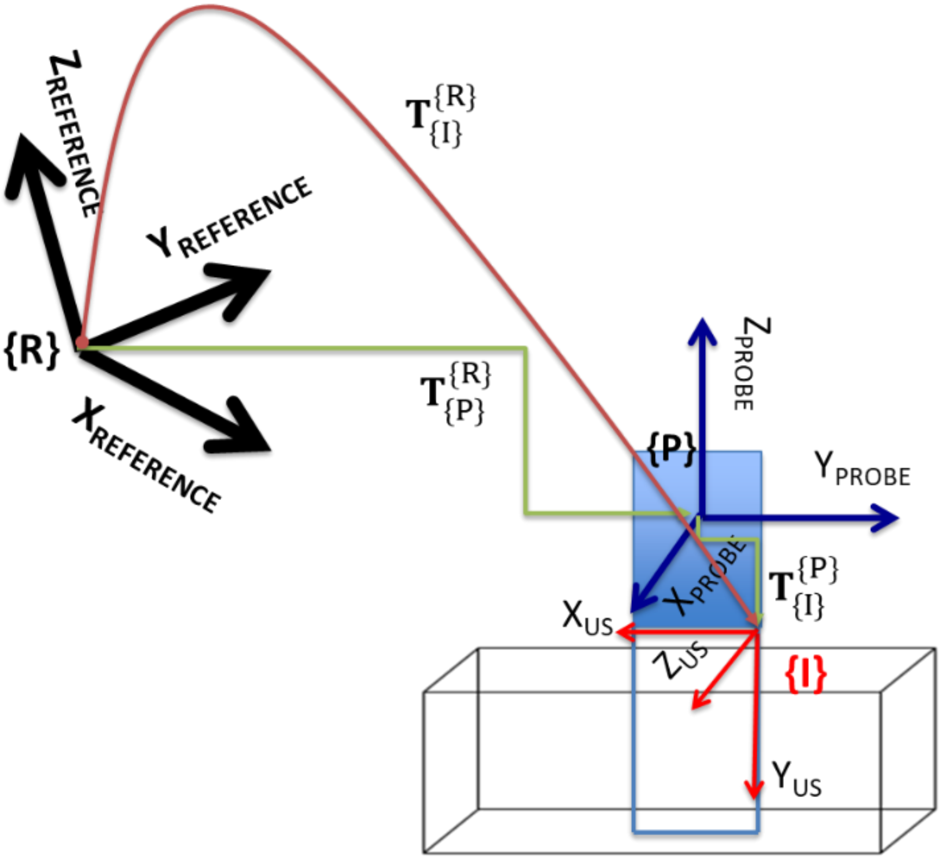


Figure 66 – US probe and US image frame calibration with respect to the reference frame

In Figures 67 and 68, all frames are illustrated with respect to the reference frame, {R}. The blue frames represent the US probe ({P}), the red frames represent the US image, {I}, as obtained in equation 4.22 and the magenta frames represent {I}, as obtained in equation 4.23. In addition, the catheters as measured using CMM machine and previously presented in Figures 34 and 35 are shown again in Figures 67 and 68 in black.

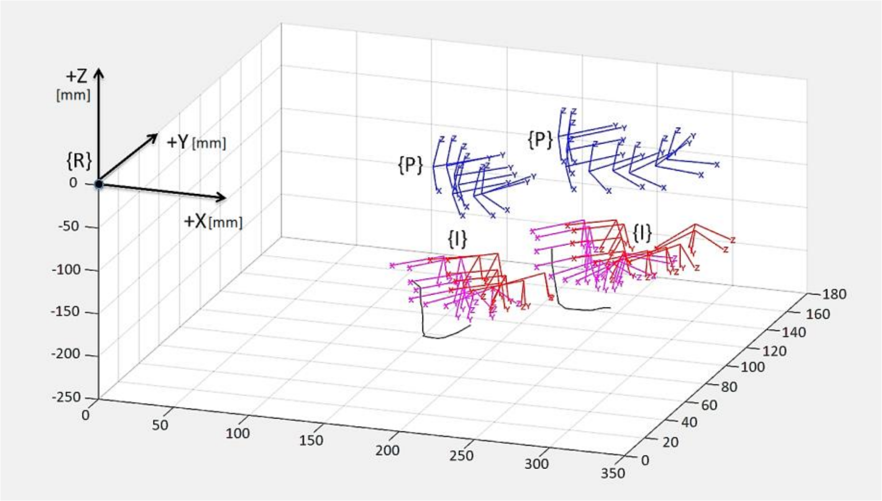


Figure 67 – US probe and US image frames shown with respect to the reference frame during scanning of planar and unplanar catheters

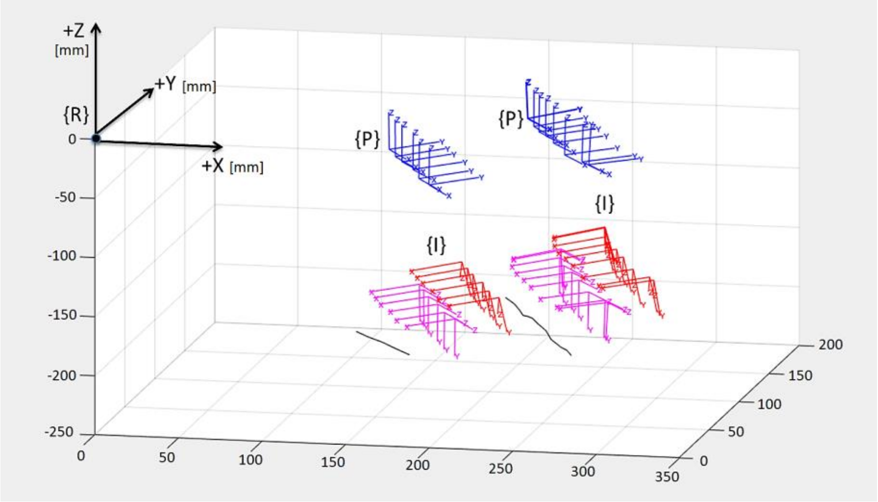


Figure 68 – US probe and US image frames shown with respect to the reference frame during scanning of straight and curvy catheters

The MATLAB code used to obtain the Figure in this section can be found in Appendix B.7.

5.3. Uncalibrated Results of Catheter 3D Shape Estimation using Kalman Filter Algorithm

Transverse US images were obtained from four differently configured 3D printed catheters. In this section Kalman filter algorithm is applied to a series of US transverse images (i.e., one for each of the straight, curvy, planar and un-planar catheters) to estimate the 3D shape of the each of the catheters by tracking their cross section over the sequence of US transverse images. The results for the straight, curvy, planar and un-planar catheters are presented in Figures 69.

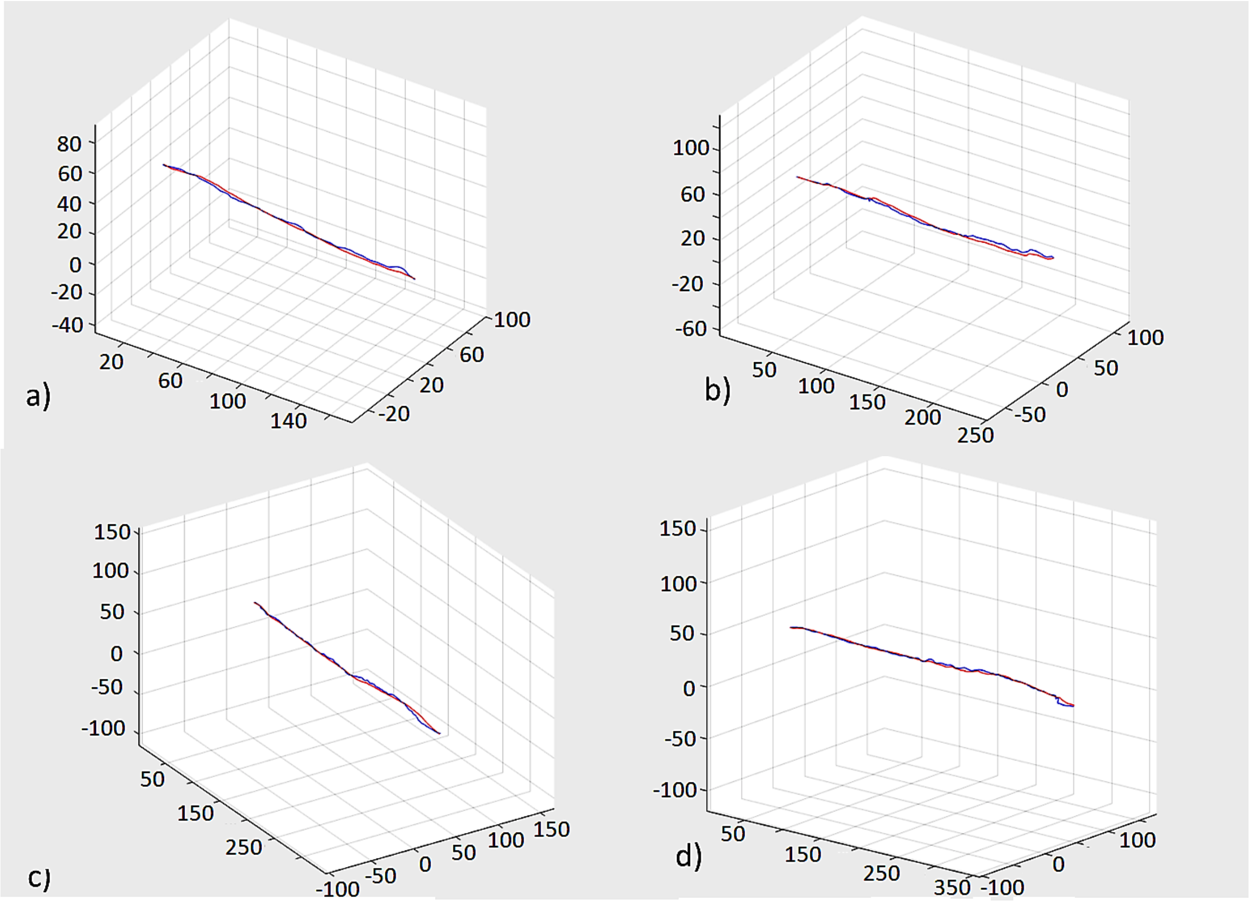


Figure 69 – Uncalibrated KF algorithm results for the (a) straight, (b) curvy, (c) planar, and (d) unplanar catheters

In Figures above, the red curve represents the KF filter tracking results (i.e., the estimated states) and the blue curve represents the measured states. Here, “state” refers to the position of the center of the catheter cross section estimated or measured in US transverse images.

The KF tracking results demonstrated above are within US image frame, {I}, and are not calibrated within the CMM workspace. Therefore, 3D estimation of the shapes is not shown within the correct perspective. In other words, the KF tracking of the cross section of the catheter is performed through a sequence of US images that are stacked together in a cuboid. This, however, does not represent the correct position and orientation (i.e., of the US probe) at which these transverse images were obtained. Consequently, the correctness of these results cannot be validated against the direct CMM measurements in Figures 34 and 35 until the next section where calibration transformations are applied to the tracking results.

The accuracy of the estimated states (red) against the measured states (blue) are quantified using both RMSE and Haussdorff distance in Table 9 of section 5.8

The MATLAB code used to obtain the Figure in this section can be found in Appendix B.7.

5.4. Calibrated Results of Catheter 3D Shape Estimation using Kalman Filter Algorithm

In this section, the KF tracking results presented in section 5.2 are calibrated in the CMM frame so that they can be comparable to the direct measurements of catheter shapes shown in Figures 34 and 35. NDI Optotracker provides sensor readings, (i.e., *x, y, z, Rx, Ry, and Rz*) at 33 Hz, of the frame {P}, US probe frame, with respect to the reference frame {R} as shown in Figure 70. In addition, through calibration procedures the transformation matrix between the US image frame, {I}, and {P} was calculated. Therefore, the points estimated in {I} using KF (i.e., $\mathbf{p}^{\{\mathrm{I}\}}$) can be mapped into {R} (i.e., $\mathbf{p}^{\{\mathrm{R}\}}$) through the following calibration transformations.

$$\mathbf{p}^{\{\mathrm{R}\}} = \mathbf{T}_{\{\mathrm{P}\}}^{\{\mathrm{R}\}}\mathbf{T}_{\{\mathrm{I}\}}^{\{\mathrm{P}\}}\mathbf{p}^{\{\mathrm{I}\}} \tag{5.2}$$

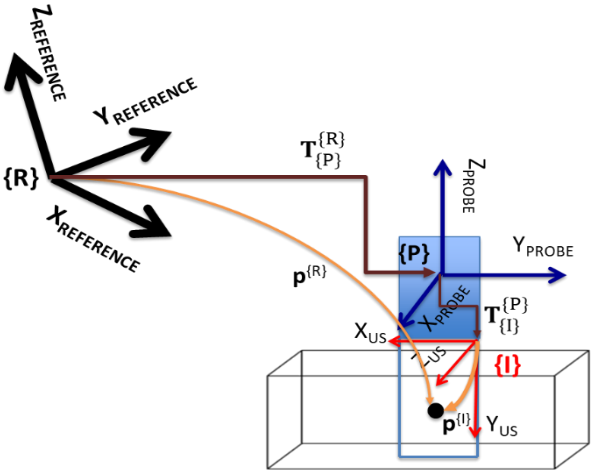


Figure 70 – Calibrating a point in {I} into a point in {R}

As discussed in section 4.7, there are two methods by which the transformation $\mathbf{T}_{\{I\}}^{\{P\}}$ can be calculated. One by using N-fiducials in calibration phantom (equation 4.22) and one by manual measurements (equation 4.23). The transformation process in equation 5.1 is carried out with both versions of $\mathbf{T}_{\{I\}}^{\{P\}}$. As shown in Figure 71, the transformation in equation 4.22 provides the shape estimation with an offset pertaining to the calculated RMSE error of 7.6223. Therefore, the results of shape estimation obtained through the transformation in equation 4.23 will be presented henceforth.

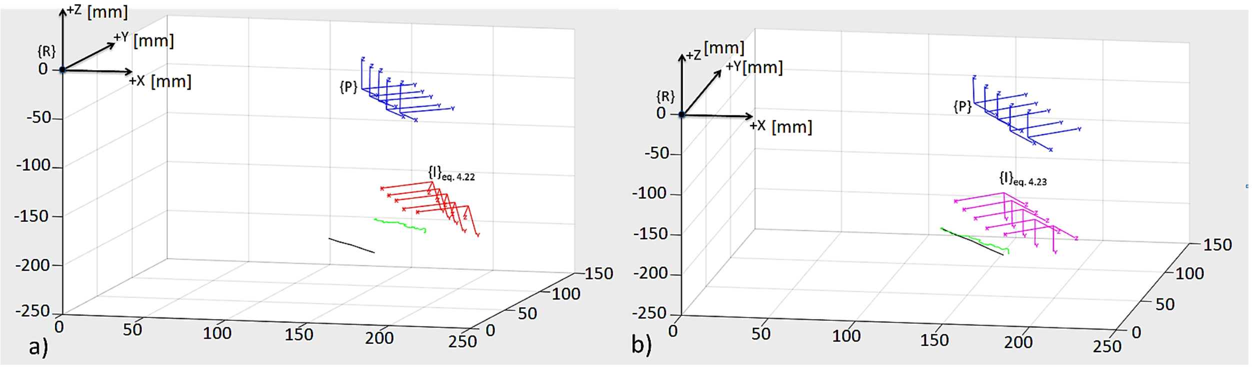


Figure 71 – Calibrated KF (a) using $\mathbf{T}_{\{I\}}^{\{P\}}$of eq.4.22, (b) $\mathbf{T}_{\{I\}}^{\{P\}}$of eq.4.23 for 3D shape estimation of the straight catheter

Figures 72 demonstrate the calibrated KF 3D shape estimation results overlaid on the respective CMM measurements (Figures 34 and 35) for each of the straight, curvy, planar and un-planar catheters.

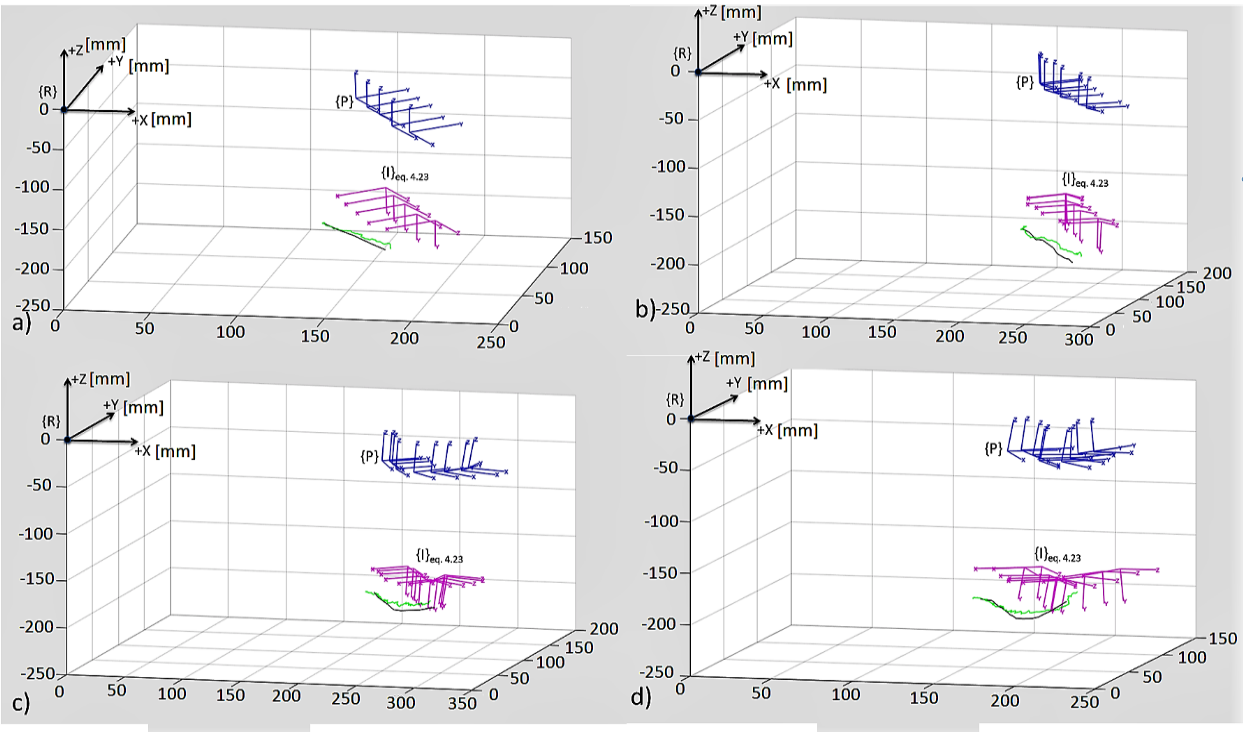


Figure 72 – Calibrated KF 3D shape estimation of the (a) straight, (b) curvy, (c) planar and (d) nonplanar catheters

In Figures above the blue frames represent $\mathbf{T}_{\{P\}}^{\{R\}}$, and the magenta frames represent $\mathbf{T}_{\{I\}}^{\{P\}}$ through equation 4.23. Also, for clearer presentation out of thirty consecutive frames only one is plotted. The black curves represent the CMM measurements of the catheter and the green curves represent the calibrated KF estimations of the 3D shape of the corresponding catheter.

The Haussdorff distance between the KF 3D shape estimations and CMM measurements for each of the figures above is summarized in Table 10 of section 5.8.

The MATLAB code used to obtain the Figure in this section can be found in Appendix B.7.

5.5. Uncalibrated Results of Catheter 3D Shape Estimation using Adaptive Kalman Filter Algorithm

In this section, AKF tracking results of the cross section of the catheters are demonstrated (Figure 73). The red curve represents the estimated states and the blue curve represents the measured states. These results are not calibrated within the CMM workspace; therefore, the figures below are not in a correct spatial perspective to be validated against direct CMM measurements of Figures 34 and 35.

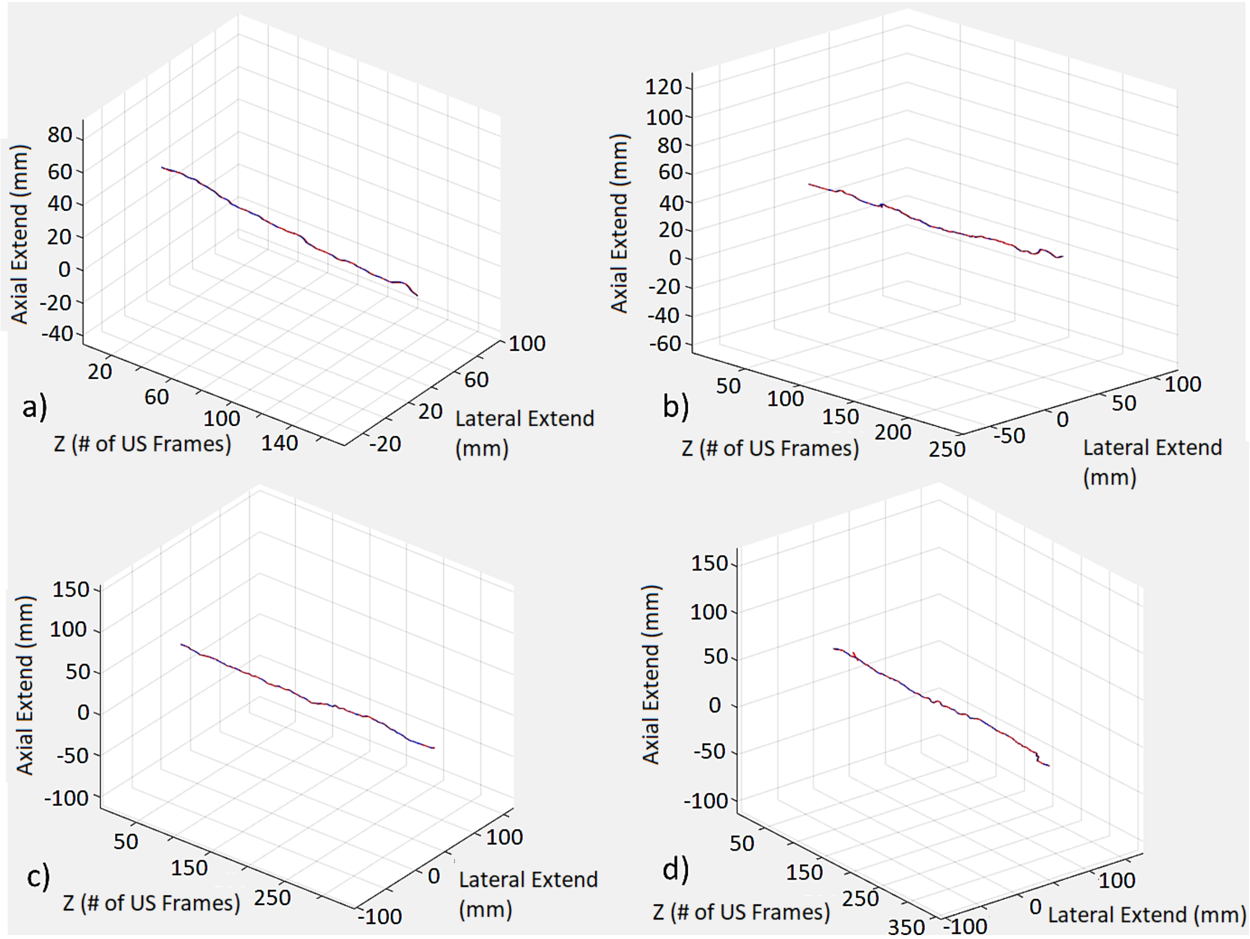


Figure 73 – Uncalibrated AKF algorithm results for the (a) straight, (b) curvy, (c) planar, and (d) unplanar catheters

The accuracy of the estimated states (red) against the measured states (blue) are quantified using both RMSE and Haussdorff distance in Table 9 of section 5.8. The MATLAB code used to obtain the Figure in this section can be found in Appendix B.7.

5.6. Calibrated Results of Catheter 3D Shape Estimation using Adaptive Kalman Filter Algorithm

The AKF tracking results of section 5.4 are calibrated within the CMM workspace frame using equation 5.1 and 5.2. The calibrated AKF 3D shape estimations of the four catheters are therefore obtained as shown in Figure 74 which are comparable to the direct CMM measurements of Figures 34 and 35 and can be validated accordingly. The AKF results calibrated using $\mathbf{T}_{\{I\}}^{\{P\}}$ from equation 4.23 is the one without an offset error and therefore the only one presented.

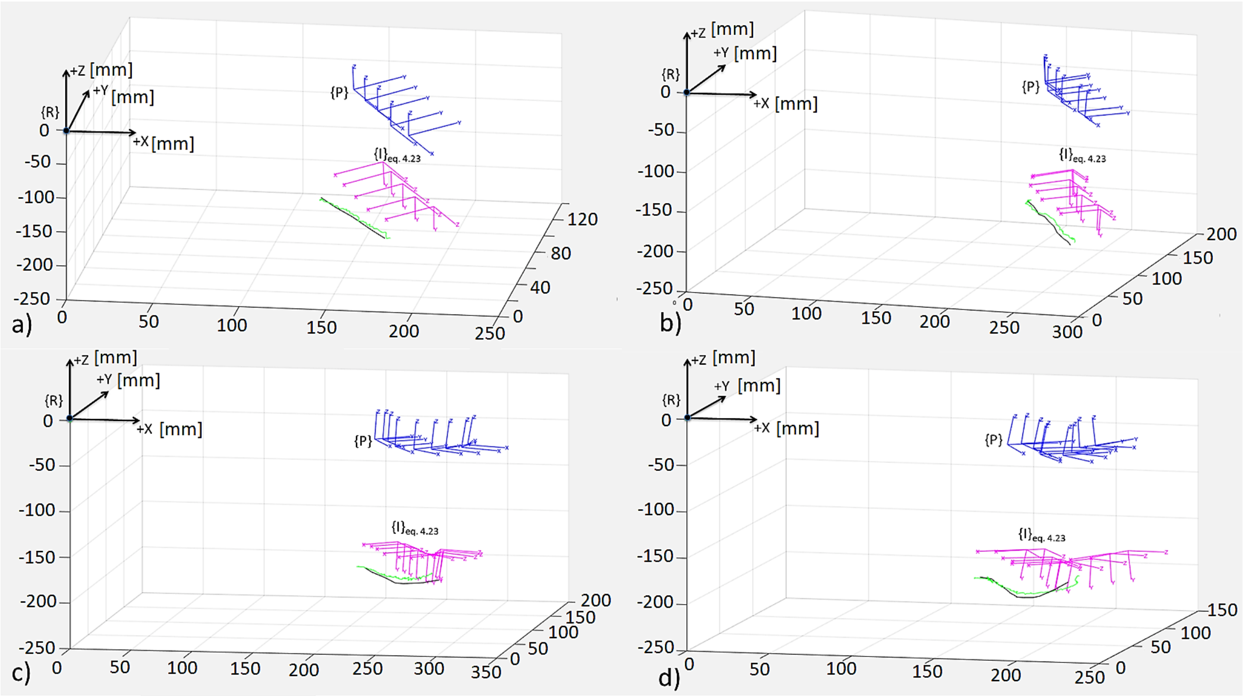


Figure 74 – Calibrated AKF 3D shape estimation of the (a) straight, (b) curvy, (c) planar and (d) nonplanar catheters

The Haussdorff distance between the AKF 3D shape estimations and CMM measurements for each of the figures above is summarized in Table 10 of section 5.8.

5.7. Uncalibrated Results of Catheter 3D Shape Estimation using Particle Filter Algorithm

In this section, PF tracking results of the cross section of the catheters are demonstrated in Figure 75. The red curve represents the estimated states and the blue curve represents the measured states. These results are not calibrated within the CMM workspace; therefore, the figures below are not in a correct spatial.

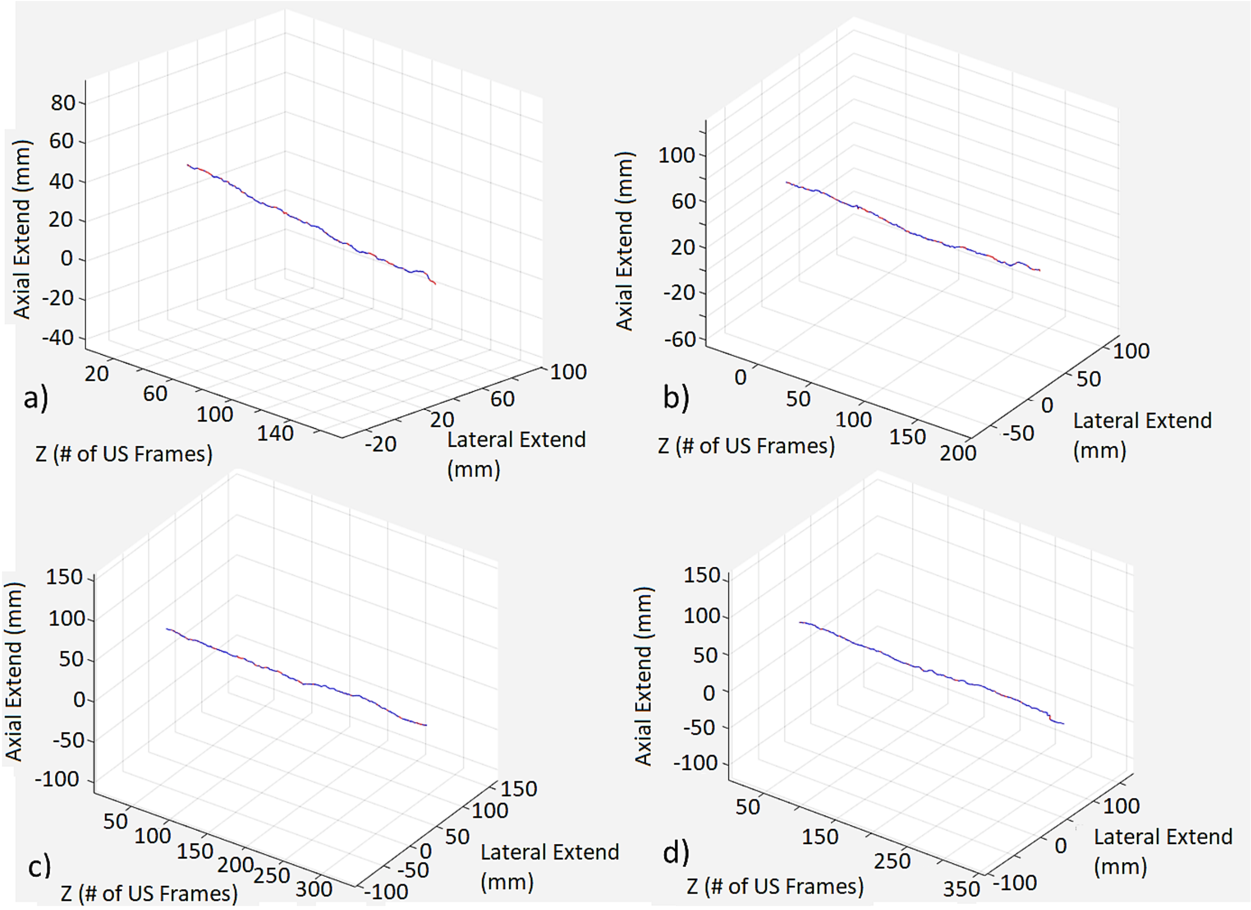


Figure 75 – Uncalibrated PF algorithm results for the (a) straight, (b) curvy, (c) planar, and (d) unplanar catheters

The accuracy of the estimated states (red) against the measured states (blue) are quantified using both RMSE and Haussdorff distance in Table 9 of section 5.8

The MATLAB code used to obtain the Figure in this section can be found in Appendix B.7.

5.8. Calibrated Results of Catheter 3D Shape Estimation using Particle Filter Algorithm

PF tracking results of section 5.4 are calibrated within the CMM workspace frame using equations 5.1 and 5.2. The calibrated PF 3D shape estimations of the four catheters are therefore obtained as shown in Figure 76 which are comparable to the direct CMM measurements of Figures 34 and 35 and can be validated accordingly.

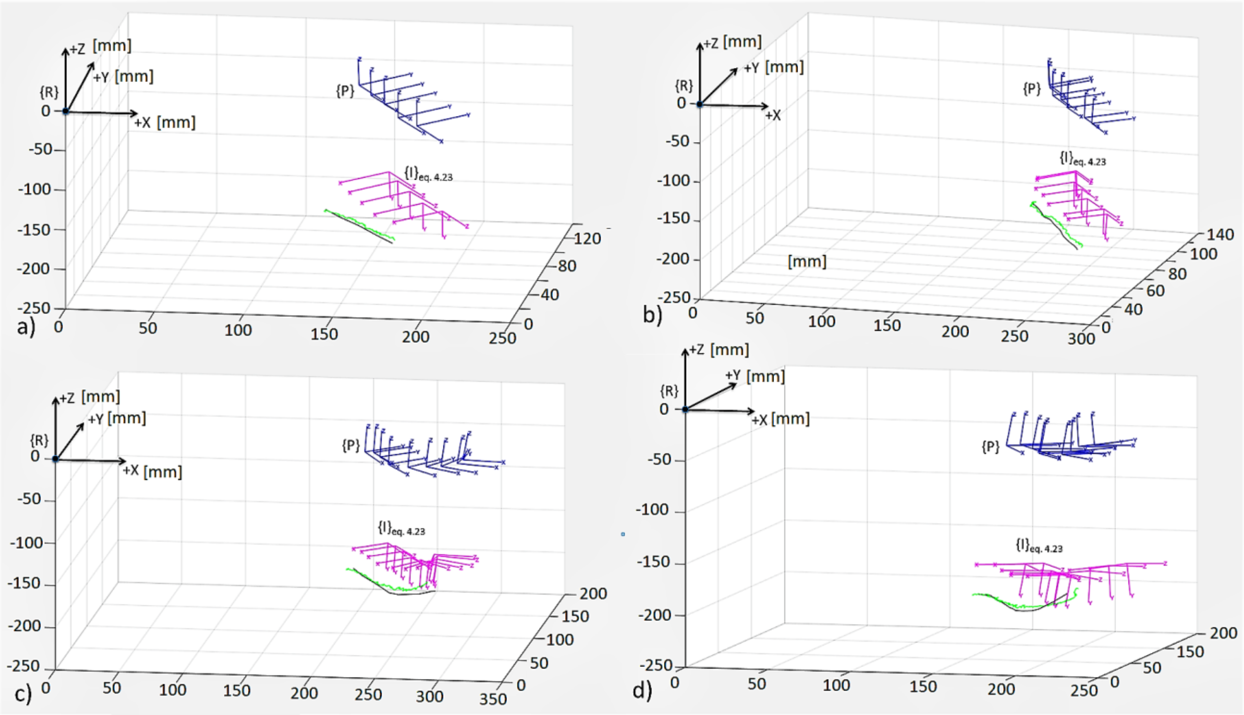


Figure 76 – Calibrated AKF 3D shape estimation of the (a) straight, (b) curvy, (c) planar and (d) nonplanar catheters

The Haussdorff distance between the PF 3D shape estimations and CMM measurements for each of the figures above is summarized in Table 10 of section 5.8.

5.9. Accuracy of Results

Table 6 compiles the RMSE and Haussdorff distance (HD) between the estimations and measurements shown in the uncalibrated KF, AKF and PF results presented in sections 5.2, 5.4 and

5.6. As shown in Figures of these sections the estimated states are shown in red and the measured states are shown in blue. RMSE and HD are calculated between these two-color coded curves.

Table 6
Accuracy of uncalibrated estimated states against the measured states for each of the four catheters (all measurements are in **mm** ± 0.1 **mm**)

Method	**Straight**		**Curvy**		**Planar**		**Un-planar**	
	RMSE	**HD**	**RMSE**	**HD**	**RMSE**	**HD**	**RMSE**	**HD**
KF	0.7	12.0	1.1	24.7	1.0	27.7	0.8	23.0
AKF	0.2	3.3	0.2	5.1	0.2	4.3	0.6	16.3
PF	0.1	0.4	0.1	0.6	0.1	0.6	0.1	0.8

According to Table 6, PF algorithm provides the most accurate tracking of the catheter cross section through consecutive US transverse image frames. After PF, AKF provides the best results. Finally, KF demonstrates the least accurate tracking of the catheter's cross section. These conclusions are consistent in all of the four catheters as shown both in RMSE figures and Haussdorff distances of Table 9.

Table 7
Accuracy of calibrated estimated states against the CMM measurements for each of the four catheters using Haussdorff distance (all measurements are in **mm** ± 0.1 **mm**)

Method	**Straight**	**Curvy**	**Planar**	**Un-planar**
KF: $\mathbf{T}_{\{I\}}^{\{P\}}$ from 4.22	41.4	50.0	36.8	46.0
KF: $\mathbf{T}_{\{I\}}^{\{P\}}$ from 4.23	11.7	10.4	14.9	16.9
AKF: $\mathbf{T}_{\{I\}}^{\{P\}}$ from 4.22	41.2	46.7	36.2	45.8
AKF: $\mathbf{T}_{\{I\}}^{\{P\}}$ from 4.23	11.3	8.8	14.9	16.7
PF: $\mathbf{T}_{\{I\}}^{\{P\}}$ from 4.22	41.1	46.5	35.7	45.4
PF: $\mathbf{T}_{\{I\}}^{\{P\}}$ from 4.23	11.2	8.6	14.8	16.5

For quantifying accuracy in the case of calibrated results (Table 7), RMSE is not applicable since it requires same number of points in the two datasets. However, the number of CMM measured points is not equal to the number of the tracked points. The Haussdorff distance method on the other hand does not require the same amount of points in the two datasets and therefore is applicable to this situation.

As shown in Table 7, the accuracy of 3D shape estimation results is greatly affected by the calibration errors discussed in section 4.6. In particular, the transformation between the US image frame and the US probe frame, $\mathbf{T}_{\{I\}}^{\{P\}}$, introduces the most error in the accuracy of the calibrated results. This is especially apparent in the case of calibration using $\mathbf{T}_{\{I\}}^{\{P\}}$ estimated using calibration phantom in section 4.5 (equation 4.22) where the Haussdorff distances amount to 35.7 - 50.0 mm. On the other hand, $\mathbf{T}_{\{I\}}^{\{P\}}$ derived using direct measurements between the US probe frame and US image frame (i.e., section 4.7) produce Haussdorff distances of about 8.6 - 16.9 mm which are comparable to the Haussdorff distances summarized in Table 9 for uncalibrated results.

Even though, the Haussdorff distances calculated in Table 7 are affected by the calibration errors, the pattern observed in Table 6 still holds. In other words, PF provides the best estimate of the 3D shape of the catheters and KF provides the worst estimate of the 3D shape of the catheters. One also may note that Haussdorff distance calculated in AKF across the four catheters shown in Table 7 is almost identical to the Haussdorff distances calculated in PF across the four catheters.

5.10. Summary

In this section, the KF, AKF and PF 3D shape estimation algorithms were applied to experimentally obtained US transverse images. These transverse US images were each obtained at specific position and orientation which was tracked by NDI Optotracker. First, the catheter cross section was tracked in the sequence of transverse US images using KF, AKF and PF, of which PF provided the best tracking results (Haussdorff distances of 0.4 - 0.8 mm) and KF the worst (Haussdorff distances of 12.0 - 27.7 mm). Next, the tracking results were calibrated in CMM's workspace using the transformation of the US probe frame with respect to the reference frame ($\mathbf{T}_{\{P\}}^{\{R\}}$) and the transformation of the US image frame with respect to the US probe frame ($\mathbf{T}_{\{I\}}^{\{P\}}$). Two versions of $\mathbf{T}_{\{I\}}^{\{P\}}$were used to calibrate the results in CMM workspace. One obtained from manual measurements (eq. 4.23) between the anticipated US image frames with respect to the probe frames and one estimated using calibration phantom (eq. 4.22). It was observed that calibration using $\mathbf{T}_{\{I\}}^{\{P\}}$ from equation 4.22, produced a noticeable offset between the calibrated 3D shape estimation results and their direct CMM measurements. This offset is clearly translated in the Haussdorff distance between the shape estimation results and CMM measurements (i.e., 35.7 - 50.0 mm). On the other hand, it was observed that calibration using $\mathbf{T}_{\{I\}}^{\{P\}}$ from equation 4.23, mapped the 3D shape estimation results to almost the same place as the CMM measurements with calculated Haussdorff distances of 8.6 - 16.9 mm.

Chapter VI

CONCLUSIONS, LIMITATIONS & FUTURE WORK

In this thesis Bayesian tracking algorithms of KF, AKF and PF were developed for 3D shape estimation of catheters. The object of interest tracked using these algorithms is the catheter cross sections captured in a sequence of US transverse B-mode images as the probe scans across the catheters. First, the performance of KF, AKF and PF shape estimation algorithms were validated by applying them to a series of MATLAB generated US images simulating US scanning along catheters with known shapes. The results of these numerical simulations demonstrated that AKF provides the most accurate estimate of the 3D shapes of the catheters (i.e., 0.2 ± 0.1 mm to 0.5 ± 0.1 mm in Haussdorff distance) within the computational time of 0.24 s. This is as opposed to the KF shape estimation with lowest accuracy (i.e., 3.5 ± 0.1 mm to 6.7 ± 0.1 mm in Haussdorff distance) and PF shape estimation with the longest computational time (i.e., 1.98 s)[40].

Since the series of MATLAB simulated US images are not the true representative of actual noise and artifacts typically present in US images, the KF, AKF and PF shape estimation algorithms are further validated on experimentally obtained US transverse B-mode images. For this purpose, four different and conventional configurations of a catheter were fabricated using a 3D printer and were securely placed inside containers. These containers were placed inside CMM workspace and were directly measured along their length using CMM. These direct CMM measurements were used as ground truth to validate the accuracy of 3D shape estimation results of KF, AKF and PF. After obtaining the CMM measurements, the containers were filled with water-based gel and transverse US images were obtained by scanning across the length of the catheters over the water-based gel using an US linear array probe of 60 mm in lateral length. The major artifact present in these US images were identified as the comet-tail artifacts which could easily be filtered out by simple thresholding because of their lower intensity values in the images compared to the catheter cross sections. These experimentally obtained pixel-based US images were then transformed into mm-

[40] Obtained by running the algorithms on the state-of-the-art HP OMEN 3.4 GHz PC with windows 10 operating system

based images using the fact that a single pixel in these images accounts for a 0.1185 x 0.1185 mm square. At this point KF, AKF and PF shape estimation algorithms were applied to each of the four sequences of transverse US images each belonging to one of the four 3D printed catheters. The PF shape estimation algorithm was shown to provide best tracking of catheter cross section along the consecutive experimentally obtained transverse US images with errors (calculated using Haussdorff distance) of 0.4 ± 0.1 mm to 0.8 ± 0.1 mm. Next to PF, AKF provided the most accurate estimates with errors of 3.3 ± 0.1 mm to 16.3 ± 0.1 mm and KF provided the least accurate estimates with errors of 12.0 ± 0.1 to 27.7 ± 0.1 mm.

The tracking results of KF, AKF and PF of each of the four 3D printed catheters were then calibrated in the CMM workspace so that they can be validated against the direct CMM measurements of these catheter obtained earlier before the container was filled with water-based gel. To calibrate the results in the CMM workspace, the position and orientation of US probe in the CMM workspace was tracked by NDI Optotracker with accuracy of 0.0767 mm using three sensors configured to represent the US probe coordinate frame in 3D space. In addition, the US image frame was calibrated in the CMM workspace by deriving the rigid transformation between the US probe coordinate frame and the US image coordinate frame. This was done using two different methods. In the first method, a custom-made calibration phantom is used to estimate the transformation of US probe frame, $\{\mathrm{P}\}$, to the US image frame, $\{\mathrm{I}\}$, $\mathbf{T}_{\{\mathrm{P}\}}^{\{\mathrm{I}\}}$, with RMSE of 7.6223 mm, by implementing N-wire fast calibration method proposed by Pagoulatos *et al.* and quaternion-based point set registration algorithm discussed by Besl *et al.* In the second method, this transformation is calculated directly by measuring the US image frame position and orientation with respect to the US probe reference frame position and orientation. There is no way to mathematically quantify the accuracy of the transformation $\mathbf{T}_{\{\mathrm{P}\}}^{\{\mathrm{I}\}}$ obtained by the second method due to its intuitive nature and being derived by direct measurements from the US probe's orientation notch. However, the calibrated KF, AKF and PF shape estimation results using this manual calibration produces much more accurate results when validated against the CMM measurements of the four catheters (errors calculated by HD: 8.6 ± 0.1 mm to 16.9 ± 0.1 mm). As opposed to the accuracy of the results calibrated by the first method which lead to errors of

35.7 ± 0.1 to 50.0 ± 0.1 mm, demonstrating a noticeable offset between the KF, AKF, PF estimated shapes and the CMM measurements of the respective catheters.

Table 8 compiles the accuracy of KF, AKF, and PF shape estimation algorithms implemented on simulated data along with the ones obtained from their implementation on the experimental and calibrated data. In this table the errors calculated by means of Haussdorff distance method are used to compare the accuracy of the results from simulated and experimental data. The reason for choosing Haussdorff distance instead of RMSE is that RMSE requires same number of points in the two datasets (i.e., estimated data and measured data). However, the number of CMM measured points (e.g., about 20-30 measured points) along the four 3D printed catheters in the experiments is not equal to the number of the estimated points obtained from KF, AKF, and PF algorithms (e.g., about 300-500 estimated points obtained at 33 Hz). The Haussdorff distance method on the other hand does not require the same amount of points in the two datasets and therefore is applicable to this situation. Therefore, to compare the accuracy of results from different validation methods (i.e., simulation vs. experiment), Haussdorff distance which is applicable to either datasets of equal or unequal number of points is a reasonable choice.

Table 8

Simulated data (Table 2) vs. calibrated experimental data (Table 7) vs. errors reported by literature (Table 1)

(all measurements are in $\mathbf{mm \pm 0.1\ mm}$ excluding the ones reported by literature)

Validation Method		**KF**	**AKF**	**PF**
Simulated US images[41]	Catheter 1	6.6	0.3	0.6
	Catheter 2	6.7	0.1	0.8
	Catheter 3	3.5	0.4	0.5
Experimentally obtained & Calibrated US images[42]	Catheter 1	11.7	11.3	11.3
	Catheter 2	10.4	8.8	8.6
	Catheter 3	14.9	14.8	14.8
	Catheter 4	16.9	16.6	16.6
Error reported by relevant literature	H. Ren [117]	2.3 ± 1.5 mm		
	H. Ren [54]	0.804 ± 0.015 mm		
	F. Chen [53]	2.23 ± 0.87 mm		

[41] Catheter 1, 2, and 3 refer to catheters depicted in Figures 6, 7 and 8.

[42] Catheter 1, 2, 3 and 4 refer to straight, curvy, planar and un-planar catheters respectively as depicted in Figure 35.

Even though calibration errors are inherent in the accuracy of the results from experimental data, PF shape estimation algorithm demonstrates the least error as per Table 8. On the other hand, when comparing the accuracy of PF shape estimation with AKF shape estimation in Table 8, one may concede that in experimental validation they are not far away from each other in terms of accuracy. In conclusion, given the fact that AKF produces almost the same accuracy as PF in ten times less computational time (0.24 s vs. 1.98 s)[43], AKF is the best algorithm for the estimation of the 3D shape of the catheter in terms of accuracy and efficiency.

In addition, the accuracy reported in the relevant literature concerning 3D shape estimation using US imaging is included in Table 8 for side by side comparison with simulated and experimental accuracies. The 3D shape estimation accuracy of experimental results is highly influenced by calibration error; therefore, the accuracy reported in Table 8 for calibrated results is much less accurate than the ones reported in literature.

At this point a discussion about the role of US resolution and image slice thickness on accuracy of the results is in place. Regarding US resolution, it is important to notice that the transducer with a relatively high frequency of 14 MHz used to obtain the US images in this thesis provides better spatial resolution for shallow depths of 2-3 cm [140]. This is because high-frequency waves are more attenuated than lower frequency waves for a given distance. Considering that the depth of US images in this thesis range from 4-5 cm, it should be recognized that spatial resolution is degraded at depths >3cm and this factors in the accuracy of tracking algorithms especially in tracking of the contour or center of catheter cross section as segmented in image processing steps. However, one may compensate for such US attenuation by adjusting time gain compensation (TGC) module provided on US systems. Using TGC, signal gain is increased as time passes from the emitted wave pulse. This correction makes equally echogenic medium look the same even if they are in different depths. In this thesis TGC of B-mode images was set to 50%[44] which by inspection produced the best resolution of the scanned catheter cross section. On the other hand, US image slice thickness

[43] obtained by running the algorithms on the state-of-the-art HP OMEN 3.4 GHz PC with windows 10 operating system

[44] Meaning that each echo received from greater depths are amplified by 50%. In this case, the TGC amplification setting can be modeled as a straight line with constant slope representing equal amplification at depth increments.

should also be noted to affect the accuracy of the detected and segmented catheter cross section. Even though the US image frame acquired in real time is interpreted as the result of a thin planar scan plane, it has associated with it out-of-plane thickness (slice thickness), which can seriously affect the information content of the US image [141]. When the scan plane is thin, the resultant US image provides an accurate representation of the scanned medium. When the scan plane is thick, however, the resultant US image represents an average of the echo signals received from the scan plane at each image depth. This can result in distortions of the imaged echo amplitude data. Also, artificial structures may sometimes appear in the image due to strongly scattering mediums located at the other margins of the slice thickness [141]. To make matters more complicated, the US image slice thickness is not constant and instead is seen to change throughout the range of image depths, much like the beam pattern of a simple lens. The minimum slice thicknesses occur at a range of 5-8 cm image depths. At shallow image depths close to the transducer the slice thickness appears to be composed of several thin, closely spaced scan planes [141]. A. Goldstein proposes a method in [141] to experimentally determine the US image slice thickness at different depths.

To improve the results of algorithms on experimentally obtained data, calibration must be refined, especially in the case of the rigid transformation ($\mathbf{T}_{\{P\}}^{\{I\}}$) between the US image frame and US probe frame. To estimate a more accurate rigid transformation matrix using N-wire fast calibration method proposed by Pagoulatos *et al.* and quaternion-based point set registration algorithm discussed by Besl *et al*, more points must be measured that are common in both calibration phantom frame and US image frame. Incorporating more measured points in the rigid transformation estimation algorithm may reduce the error from 7.6223 mm as obtained in this thesis. However, as explained before in experiments conducted using fCal2.0 and L14-5W/60 linear array US probe, no more than ten points are measurable given the small size of fCal2.0 phantom. To be able to measure more points using N-fiducials, the use of fCal3.1 phantom is recommended[45]. This phantom is bigger in size than fCal2.0 phantom (76 mm in width, 220 mm in length and 150 mm in height) and incorporates a total of 238 holes in 14 rows and 17 columns, which can be used to incorporate more N-fiducials. The

[45] The .stl file of fCal3.1 can be downloaded from:

http://perk-software.cs.queensu.ca/plus/doc/nightly/modelcatalog/printable/fCal_3.1.stl more N-fiducials, the more measurable points common to both the calibration phantom and US image and therefore the lower the calculated error will be. Another option is to perform the US image frame calibration using the Plus Toolkit[46] which automatically calibrates the US image with typical calibration error of 0.8-1.5 mm and can even go down to <0.5 mm with an optimal set up. The Plus toolkit software package automatically obtains the image-to-probe transform by point-matching. For this purpose, the toolkit utilizes SonixTouch GPS extension, Stylus tracker and fCal2.x calibration phantom.

Another major limitation inherent in the simulations and experiments presented in this thesis is the lack of consideration for speckle noise in US images. Speckle noise in US emanates from signals generated by tissue cells and connective tissues. However, in the experiments conducted for this thesis, the catheter was submerged into a container filled with water-based gel, which did not emanate speckle noise in the US B-mode transverse images obtained. Therefore in future work, to evaluate the applicability of KF, AKF and PF algorithms in realistic scenarios especially in the presence of speckle noise in US B-mode transverse images, the use of ex-vivo tissue models or realistic tissue-mimicking phantoms instead of the water-filled container is required. In addition, the variability of the speed of sound through different tissues should be taken into consideration. In this thesis the speed of sound is fixed to the default of 1540 m/s. However, the speed of sound varies from tissue to tissue. For example, in bone, blood, fat, kidney and liver the speed of sound is 4080 m/s, 1570 m/s, 1450 m/s, 1560 m/s and 1570 m/s respectively.

Future work also involves implementing the presented AKF shape estimation algorithm in real time where as soon as US transverse images are acquired they are fed into the shape estimation algorithm which continuously updates the anticipated shape of the catheter (especially at its distal end). In addition, AKF can be incorporated in a visual servo control unit to track a catheter using robotically maneuvered US probe. In other words, by considering the next AKF estimate of the US detected catheter cross section, the robot can move the US probe efficiently so that the catheter’s distal end is always within the field of view of the US probe. Such robotic US 3D shape estimation mechanism

[46] http://perk-software.cs.queensu.ca/plus/doc/nightly/user/index.html

not only assists the surgeon in the hand/eye synchronization involved in MIS procedures with tendon-driven catheters but also provide the surgeon with 3D structures of these catheters from 2D US planes obtained by scanning across them over the region of interest.

APPENDIX A

MATLAB CODES

B.1. MATLAB Codes for Section 3.4

B.1.1: Catheter Configuration # 1

```
% **************************************************************************
%            Modeling Catheter Configuration #1 in 3D
%                Written By: Niloufaralsadat Hashemi
%                     Last Updated: June 9, 2018
%                          Libraries used:
%            MATLAB Computer Vision & MATLAB Image Processing
% **************************************************************************

r = 10; % radius of the blob in pixels
j=1;

%  The straight section of the catheter
for i=1:194
        A = zeros(231,447); % Overal size of frames (v,u)
        m = {40,60}; % midpoint (v, u) of the blob
        A(m{:})=1;
        B = imdilate(A,strel('disk', r,0) );
        idisp(B)
        testDemo{i} = B;
end

% The curved section of the catheter
for i=195:223
        A = zeros(231,447); % size of frames (v,u)
        m = {floor(40+(1.5)^j),60}; % midpoint (v, u) of the blob
        A(m{:})=1;
        B = imdilate(A,strel('disk', r,0) );
        idisp(B)
        testDemo{i} = B;
        j=j+0.4;
end

% Extracting the centeroid of the blobs
MM=[];
figure,
hold on
for i=1:223
    [limage, nblobs] = bwlabel(testDemo{i});
    info = regionprops(limage, 'area', 'Centroid');
    x = info.Centroid;
    xm = [x(1); x(2); i];
    MM = [MM xm];
end

% Plotting results
 plot3(MM(3,:), MM(1,:), MM(2,:), 'r-')

 title('Catheter 3D Configuration # 1','FontSize', 12)
 xlabel('z (in number of US frames)', 'FontSize', 12)
 ylabel('u (in pixels)', 'FontSize', 12)
 zlabel('v (in pixels)', 'FontSize', 12)
 grid on
 axis square equal
 hold off

% Stacking on the Images into a volumetric matrix
O=[];
for i=1:223
      O = cat(3, O, uint8(255*testDemo{i}));
end
idisp(testDemo{100})
```

B.1.2: Catheter Configuration # 2

```
% ************************************************************************
%            Modeling Catheter Configuration #2 in 3D
%                Written By: Niloufaralsadat Hashemi
%                     Last Updated: June 9, 2018
%                          Libraries used:
%            MATLAB Computer Vision & MATLAB Image Processing
% ************************************************************************

r = 10; % radius of the blob in pixels
j=1;

%  The straight section of the catheter
for i=1:194
        A = zeros(231,447); % Overal size of frames (v,u)
        m = {40,60}; % midpoint (v, u) of the blob
        A(m{:})=1;
        B = imdilate(A,strel('disk', r,0) );
        idisp(B)
        testDemo{i} = B;
end

% The curved section of the catheter
for i=195:223
```

```
        A = zeros(231,447); % size of frames (v,u)
        m = {40,floor(60+(1.5)^j)}; % midpoint (v, u)
        A(m{:})=1;
        B = imdilate(A,strel('disk', r,0) );
        idisp(B)
        testDemo{i} = B;
        j=j+0.4;
end
% Extracting the centeroid of the blobs
MM=[];
figure,
hold on
for i=1:223
    [limage, nblobs] = bwlabel(testDemo{i});
    info = regionprops(limage, 'area', 'Centroid');
    x = info.Centroid;
    xm = [x(1); x(2); i];
    MM = [MM xm];
end

% Plotting results
 plot3(MM(3,:), MM(1,:), MM(2,:), 'r-')

 title('Catheter 3D Configuration # 1','FontSize', 12)
 xlabel('z (in number of US frames)', 'FontSize', 12)
 ylabel('u (in pixels)', 'FontSize', 12)
 zlabel('v (in pixels)', 'FontSize', 12)
 grid on
 axis square equal
 hold off

% Stacking on the Images into a volumetric matrix
O=[];
for i=1:223
      O = cat(3, O, uint8(255*testDemo{i}));
end
idisp(testDemo{100})
```

B.1.3: Catheter Configuration # 3

```
% ***************************************************************************
%            Modeling Catheter Configuration #3 in 3D
%                Written By: Niloufaralsadat Hashemi
%                     Last Updated: June 9, 2018
%                          Libraries used:
%            MATLAB Computer Vision & MATLAB Image Processing
% ***************************************************************************

r = 10; % radius of the blob in pixels
j=1;
```

```
jj=0;
c=0;

% The straight section of the catheter
for i=1:192
        A = zeros(231,447); % Overall size of frames (v,u)
        m = {40+c,60+c}; % midpoint (v, u) of the blob
        A(m{:})=1;
        B = imdilate(A,strel('disk', r,0) );
        idisp(B)
        testDemo{i} = B;
        c=c+1;
end

% The curved section of the catheter
for i=193:223
        A = zeros(231,447); % size of frames (v,u)
        m = {floor(233-(1.5)^j), floor(253-jj^1.5)}; % midpoint (v, u)
        A(m{:})=1;
        B = imdilate(A,strel('disk', r,0) );
        idisp(B)
        testDemo{i} = B;
        j=j+0.4;
        jj=jj+1;
end

% Extracting the centroid of the blobs
MM=[];
figure,
hold on
for i=1:223
    [limage, nblobs] = bwlabel(testDemo{i});
    info = regionprops(limage, 'area', 'Centroid');
    x = info.Centroid;
    xm = [x(1); x(2); i];
    MM = [MM xm];
end

% Plotting results
 plot3(MM(3,:), MM(1,:), MM(2,:), 'r-')


 title('Catheter 3D Configuration # 1','FontSize', 12)
 xlabel('z (in number of US frames)', 'FontSize', 12)
 ylabel('u (in pixels)', 'FontSize', 12)
 zlabel('v (in pixels)', 'FontSize', 12)
 grid on
 axis square equal
 hold off

% Stacking on the Images into a volumetric matrix
O=[];
for i=1:223
```

```
    O = cat(3, O, uint8(255*testDemo{i}));
end
idisp(testDemo{100})
```

B.1.4: Transverse and Longitudinal Images

```
% ***************************************************************************
%     Plotting Catheter Transverse and Longitudinal Cross-sections in 3D
%                 Written By: Niloufaralsadat Hashemi
%                     Last Updated: June 9, 2018
%                         Libraries used:
%       MATLAB Computer Vision, MATLAB Image Processing & image3 [127]
% ***************************************************************************

% Setting the orientation
T = [1 0 0 0;0 1 0 0;0 0 1 0; 0 0 0 1]*trotx(pi/2)*trotz(pi/2)*trotx(pi/2);

% Slicing through the volumetric matrix O
    h0 = slice3(O,T,2,60);
    h1 = slice3(O,T,3,20);
    h2 = slice3(O,T,3,40);
    h3 = slice3(O,T,3,60);
    h4 = slice3(O,T,3,80);
    h5 = slice3(O,T,3,100);
    h6 = slice3(O,T,3,120);
    h7 = slice3(O,T,3,140);
    h8 = slice3(O,T,3,160);
    h9 = slice3(O,T,3,180);
    h10 = slice3(O,T,3,200);
    h11 = slice3(O,T,3,220);

    set([h0, h1, h2, h3, h4, h5, h6,],'EdgeColor','blue','LineStyle','-');
    set([h6, h7, h8, h9, h10, h11],'EdgeColor','blue','LineStyle','-');

    colormap gray(88);
    view(30,30); axis equal; axis vis3d;
    light;
    title('Ultrasound Images', 'FontSize', 14);
```

B.1.5: Adding Noise

```
% ***************************************************************************
%     Adding Noise to Transverse Images of the Catheter Cross Section
%                 Written By: Niloufaralsadat Hashemi
%                     Last Updated: June 9, 2018
%                         Libraries used:
%             MATLAB Computer Vision, MATLAB Image Processing
% *************************************************************************

thresholdF = 0.5;
kernelStd = 1.8;
se = strel('disk',12);
figure,
hold on
for i=1:223
    kernelRadius = ceil(5 * kernelStd);
    kernelLength = (2 * kernelRadius ) + 1;
    G = fspecial('gaussian', [kernelLength, kernelLength], kernelRadius);
    J = imfilter(testDemo{i},G,'same');

    % Adding the speckle noise
    J = imnoise(J,'speckle', 0.05);
    % Adding the Gaussian noise
    J = imnoise(J,'gaussian', 0.2);

    Corrupted{i} = J;
    thresholdedF{i} = (Corrupted{i} > thresholdF);
    idisp(Corrupted{i})
end
```

B.2. MATLAB Codes for Section 3.5

B.2.1: 3D Shape Estimation with Kalman Filter Algorithm

```
% *************************************************************************
%            3D Shape Estimation of the Catheter with KF
%                Written By: Niloufaralsadat Hashemi
%                     Last Updated: June 9, 2018
%                          Libraries used:
%            MATLAB Computer Vision, MATLAB Image Processing &
%                 Peter Cork Computer Vision Toolbox [128]
% *************************************************************************

TT = cputime;
tic
% KF inputs:
DT    = 1/25;   % inter-sample time interval (s)

Q     = eye(4)*0.01;  % continuous process noise covariance
R     = [2 2;2 2];    % measurement noise variance
C     = [1 0 0 0; 0 1 0 0]; % measurement sensitivity matrix
A     = [1 0 DT 0; 0 1 0 DT; 0 0 1 0; 0 0 0 1]; % Transition Matrix

% Initializations:
 xh   = [60;40;0;0]; % initial estimate of x
P    = 100*eye(4);   % initial covariance of estimation uncertainty
```

```
measurements = [];
predictions = [];

for i=2:223

    % State Prediction

    xp = A*xh;
    predictions = [predictions xp];

    % Covariance Prediction

    Pp = Q + A*P*A';

    % Drawing the predicted state on the next frame

    imshow(thresholdedF{i})
    hold on;
    plot(xp(1), xp(2), 'r*', 'MarkerSize', 10);
    hold off;
    pause(0.01)



    % Obtaining the next measurement:
    % step 1 - defining the ROI:

     ROI{i-1} = imcrop(thresholded{i}, [xp(1)-50 xp(2)-50 100 100]);

    % Obtaining New Measurement:

     BLOBS = iblobs(thresholdedF{i});
     CRSEC = BLOBS(2:end);
     CRSEC_Area = CRSEC.area;
     [Area, Index] = max(CRSEC_Area);
     x = CRSEC(Index).p;
     xm = [x(1)+xp(1)-50; x(2)+xp(2)-50];
     xm = [x(1); x(2)];
     measurements = [measurements xm];

    % Drawing the measured cross section center

      imshow(thresholdedF{i})
      hold on;
      plot(xm(1), xm(2), 'b*', 'MarkerSize', 10);
      hold off
      pause(0.01)

    % Kalman Gain:

    K_Gain = Pp*C'*inv(R+C*Pp*C');
    % State Update:

    xh = xp + K_Gain*(xm - C*xp);

    % Covariance Update:

    P = (eye(4) - K_Gain*C)*Pp;

end
toc
EE = cputime-TT;

len_m = size(measurements);
len_p = size(predictions);
Z_m = [1:len_m(2)];
Z_p = [1:len_p(2)];
predictions = [predictions; Z_p];
measurements = [measurements; Z_m];
Catheter_R = 5 % in mm

figure,
hold on

for l=1:len_m(2)
    teta=0:0.01:2*pi ;
    xc=measurements(1,l)*ones(1,629) + Catheter_R*cos(teta);
    yc=measurements(2,l)*ones(1,629) + Catheter_R*sin(teta) ;
    zc=measurements(3,l)*ones(1,629) + zeros(size(xc)) ;
    patch(xc,yc,zc,'k')
    axis equal
    plot3(predictions(1,l),predictions(2,l),predictions(5,l),'.r')
end

grid on

%
% Kalman Filter Results
figure,
hold on
grid on
axis equal
plot3(MM(3,:), MM(1,:), MM(2,:), 'b-')
plot3(predictions(5,:),predictions(1,:),predictions(2,:),'r-')
legend('Actual','Estimated','FontSize')
M = [measurements(1,:)' measurements(2,:)'];
P = [predictions(1,:)' predictions(2,:)'];
A = [MM(1,:)' MM(2,:)'];
xlabel('z (in number of US frames)', 'FontSize', 12)
ylabel('u (in pixels)', 'FontSize', 12)
zlabel('v (in pixels)', 'FontSize', 12)
```

B.3. MATLAB Codes for Section 3.6

B.3.1: 3D Shape Estimation with Adaptive Kalman Filter Algorithm

```
% ********************************************************************
%            3D Shape Estimation of the Catheter with AKF
%                Written By: Niloufaralsadat Hashemi
%                    Last Updated: June 9, 2018
%                        Libraries used:
%            MATLAB Computer Vision, MATLAB Image Processing &
%                Peter Cork Computer Vision Toolbox [128]
% ********************************************************************

TT = cputime;
tic
% KF inputs:
DT    = 1/25;   % inter-sample time interval (s)

Q     = eye(4)*0.01;  % continuous process noise covariance
R     = [2 2;2 2];    % measurement noise variance
C     = [1 0 0 0; 0 1 0 0]; % measurement sensitivity matrix
A     = [1 0 DT 0; 0 1 0 DT; 0 0 1 0; 0 0 0 1]; % Transition Matrix

% Initializations:
 xh   = [60;40;0;0]; % initial estimate of x
P    = 100*eye(4);   % initial covariance of estimation uncertainty
measurements = [];
predictions = [];

% Adaptive KF Set up:

N = 5;
q_i{1} = xh;
q_k{1} = xh;
delta_k{1} = -P;
q_bar_N = 0;

for i=2:223

    % State Prediction

    xp = A*xh;
    predictions = [predictions xp];
```

```
% Covariance Prediction

Pp = Q + A*P*A';

% Figure 17: Drawing the predicted state on the next frame

imshow(thresholdedF{i})
hold on;
plot(xp(1), xp(2), 'r*', 'MarkerSize', 10);
hold off;
pause(0.01)



% Obtaining the next measurement:
% step 1 - defining the ROI:

 ROI{i-1} = imcrop(thresholded{i}, [xp(1)-50 xp(2)-50 100 100]);

% Obtaining New Measurement:

 BLOBS = iblobs(thresholdedF{i});
 CRSEC = BLOBS(2:end);
 CRSEC_Area = CRSEC.area;
 [Area, Index] = max(CRSEC_Area);
 x = CRSEC(Index).p;
 xm = [x(1)+xp(1)-50; x(2)+xp(2)-50];
 xm = [x(1); x(2)];
 measurements = [measurements xm];

% Drawing the measured cross section center

  imshow(thresholdedF{i})
  hold on;
  plot(xm(1), xm(2), 'b*', 'MarkerSize', 10);
  hold off
  pause(0.01)

% Kalman Gain:

K_Gain = Pp*C'*inv(R+C*Pp*C');

% State Update:

xh = xp + K_Gain*(xm - C*xp);

% Covariance Update:

P = (eye(4) - K_Gain*C)*Pp;

% Adaptive KF starts here
```

```
    q_i{k} = (Predictions{k} - A*Predictions{k-1});
    q_k{k} = (Predictions{k} + A*Predictions{k-1});
    delta_k{k} = A*P_stack{k-1}*A' - P_stack{k};

    if k > N

        if ((k-1) == N)

            for i=(k-N+1):k
                q_bar_N = q_bar_N + q_i{i};
            end

            q_bar_N = (1/N) * q_bar_N;

            q_bar_k{k} = q_bar_N + (1/N) * (q_k{k} - q_k{k-N});

        else

            q_bar_k{k} = q_bar_k{k-1} + (1/N) * (q_k{k} - q_k{k-N});

        end


        Q = Q + (1/(N-1))*(q_k{k} - q_bar_k{k})*(q_k{k} - q_bar_k{k})'...
            -(1/(N-1))*(q_k{k-N} - q_bar_k{k})*(q_k{k-N} - q_bar_k{k})'...
            +(1/((N^2)-N))*(q_k{k} - q_k{k-N})*(q_k{k} - q_k{k-N})'...
            +(1/N)*(delta_k{k-N} - delta_k{k});




end
toc
EE = cputime-TT;

len_M = size(Measurements);
len_P = size(Predictions);
Z_M = [1:len_M(2)];
Z_P = [1:len_P(2)];
Measurements = [Measurements; Z_M];
Catheter_R = 10 % in pixels

figure,
hold on

for l=1:len_M(2)
    teta=0:0.01:2*pi ;
    xc=Measurements(1,l)*ones(1,629) + Catheter_R*cos(teta);
    yc=Measurements(2,l)*ones(1,629) + Catheter_R*sin(teta) ;
    zc=Measurements(3,l)*ones(1,629) + zeros(size(xc)) ;
    patch(xc,yc,zc,'k')
    axis equal
end

PP=[];
for h=1:len_P(2)
    Predictions{h} = [Predictions{h}; h];
    plot3(Predictions{h}(1),Predictions{h}(2),Predictions{h}(5),'.r')
    PP = [PP Predictions{h}(1:2)];
end
grid on
hold off

%
% Adaptive Kalman Filter Results
figure,
hold on
grid on
axis equal
plot3(MM(3,:), MM(1,:), MM(2,:), 'b-')
plot3(MM(3,:),PP(1,:),PP(2,:),'r-')
legend('actual','predicted','measured')
M = [Measurements(1,:)' Measurements(2,:)'];
P = PP';
A = [MM(1,:)' MM(2,:)'];
xlabel('z (in number of US frames)', 'FontSize', 12)
ylabel('u (in pixels)', 'FontSize', 12)
zlabel('v (in pixels)', 'FontSize', 12)
axis square equal
```

B.4. MATLAB Codes for Section 3.7

B.4.1: 3D Shape Estimation with Particle Filter Algorithm

```
% **********************************************************************
%              3D Shape Estimation of the Catheter with PF
%                 Written By: Niloufaralsadat Hashemi
%                     Last Updated: June 9, 2018
%                         Libraries used:
%             MATLAB Computer Vision, MATLAB Image Processing &
%                  Peter Cork Computer Vision Toolbox [128] &
%                 Ellipse Fit (Direct Method) Function [129]
% **********************************************************************

TTT = cputime;
tic
numFrames = size(thresholded, 2);
[MR, MC, Dim] = size(thresholded{1}); % The size of US frames
```

```
% Kalman Filter Initializations

DT    = 1/25;   % intersample time interval (s)

Q     = eye(4)*0.01;  % continuous process noise covariance
R     = [22.5 15; 15 10];   % measurement noise variance

C     = [1 0 0 0; 0 1 0 0]; % measurement sensitivity matrix
A     = [1 0 DT 0; 0 1 0 DT; 0 0 1 0; 0 0 0 1];

xm   = [60;40;0;0];           % initial estimate of x
                              % Assuming we know the 1st insertion
                              % Particles are defined about this

% Contour Tracking by Particle Filter

nparticles = 400;
x = zeros(4, 1, nparticles, numFrames); % state vectors
P = zeros(4, 4, nparticles, numFrames); % est. covariance of state vector
for i = 1 : nparticles % Initialize estimated covariance
    for j = 1  : numFrames
        P(1,1,i,j) = 100;
        P(2,2,i,j) = 100;
        P(3,3,i,j) = 100;
        P(4,4,i,j) = 100;
    end
end

% Initialize weigths to equal
weigths(1, 1:nparticles) = (1/nparticles);

% Initialize hypothesis to plausible values for states

area1PR = 20; % Range of pixels of area 1
X1 = randi([floor(xm(1)-area1PR)  floor(xm(1)+area1PR)], 1, nparticles);
X2 = randi([floor(xm(2)-area1PR)  floor(xm(2)+area1PR)], 1, nparticles);
X3 = zeros(2, nparticles);
X = [X1; X2; X3]; % particle states

% show_particles of hypothesis:
figure(1)
imshow(thresholdedF{1})
title('Showing Particles', 'FontSize', 14)
hold on
plot(X(1,:), X(2,:), 'r.')
Pred = [];
Meas = [];

% For each time step:
```

```
for t = 2:223

imshow(thresholdedF{t})
hold on

    % Forecasting - Using Kalman Filter/ Adaptive Kalman Filter



    for np = 1 : nparticles
        x(:,1,np,t) = A*X(:,np); % state prediction
        P(:,:,np,t) = Q + A*P(:,:,np,t)*A'; % covariance prediction
       plot(x(1,1,np,t), x(2,1,np,t),'b.');  % imp code
    end

    hold off

    % Calculating the weigths: Set weigth for particles according to
    % measurement model

    ROI = edge(thresholded{t});
    imshow(ROI)
    pixel_num = sum(ROI(:) == 1);

    [u,v] = find(ROI); % (v,u) are the pixel coordinates - measurements

    % Normalise weigths

    BLOBS = iblobs(thresholded{t});
    CRSEC = BLOBS(2:end);
    CRSEC_Area = CRSEC.area;
    [Area, Index] = max(CRSEC_Area);
    x_middle = CRSEC(Index).p;



    D = [];
    for i=1:pixel_num
        D = [D; sqrt((v(i)-x_middle(1))^2+(u(i)-x_middle(2))^2)];
    end
    Dmin = min(D);
    Dmax = max(D);
    Dmid = median(D);

    DP = [];
    for i=1:nparticles
        W = sqrt((x(1,1,i,t)-x_middle(1))^2+(x(2,1,i,t)-x_middle(2))^2);
        DP = [DP 1/abs(Dmid-W)];
    end
    total = sum(DP);
    weigths = (1/total)*DP;
```

```
% Resampling
% Extracting the closest particles to the contour
for i=1:sqrt(nparticles)
    [M, Index] = max(weigths);
    weigths(Index) = -1;
    P_index(i) = Index;
End

% Draw the particles:
x_ellipse = [];
y_ellipse = [];
imshow(thresholdedF{t})
hold on
for i=1:sqrt(nparticles)
     plot(x(1,1,P_index(i),t), x(2,1,P_index(i),t),'b.'); % imp code
    x_ellipse = [x_ellipse; x(1,1,P_index(i),t)];
    y_ellipse = [y_ellipse; x(2,1,P_index(i),t)];
end
E = [x_ellipse y_ellipse];

% Fit an ellipse to the particles

[a b]=EllipseDirectFit(E, [0 1 0]); % Predicted Ellipse in red
Pred=[Pred b'];

pause(5)
hold off
x_corrected = [];

% Correction and corrected Ellipse
for i=1:sqrt(nparticles)
    Pp = P(:,:,P_index(i),t);
    K_Gain = Pp*C'*inv(R+C*Pp*C'); % Kalman Gain

    % Finding corresponding points on measured edge:
    OG1 = abs(v-x(1,1,P_index(i),t));
    OG2 = abs(u-x(2,1,P_index(i),t));
    [k1 og1] = min(OG1);
    [k2 og2] = min(OG2);
    x_corrected = [x_corrected (x(:,1,P_index(i),t) + K_Gain*([v(og1);
    u(og2)] - C*x(:,1,P_index(i),t)))]; % State Update
    P(:,:,i,t+1) = (eye(4) - K_Gain*C)*Pp; % Covariance Update
end
imshow(thresholdedF{t})
hold on
plot(x_corrected(1,:), x_corrected(2,:), 'r*')

% Fit an ellipse to the corrected particles
F=[x_corrected(1,:)', x_corrected(2,:)'];
[aa bb]= EllipseDirectFit(F, [0  0 1]); % corrected Ellipse in blue
Meas=[Meas bb'];
```

```
    hold off

    % Breeding new particles for the next round
    X = [];
    X = [X x_corrected];
    newParticles = sqrt(nparticles);
      range = 30;
      for h=1:sqrt(nparticles)
          X_B1 = randi([floor(x_corrected(1,h)-range) …
                 … floor(x_corrected(1,h)+range)], 1, newParticles);
          X_B2 = randi([floor(x_corrected(2,h)-range) …
                 … floor(x_corrected(2,h)+range)], 1, newParticles);
          X_B3 = x_corrected(3:4,h)*ones(1,newParticles);
          X = [X [X_B1;X_B2;X_B3]];
          for lp = h*(sqrt(nparticles)+1):(h+1)*sqrt(nparticles)
            P(:,:,lp,t+1) = P(:,:,h,t+1);
          end
      end

end
EEE = cputime-TTT;
toc

% Particle Kalman Filter Results

figure,
hold on
grid on
axis equal
plot3(MM(3,:), MM(1,:), MM(2,:), 'b-')
plot3([1:222],Pred(1,:),Pred(2,:),'r-')
legend('actual','estimated')
M = [Meas(1,:)' Meas(2,:)'];
P = [Pred(1,:)' Pred(2,:)'];
A = [MM(1,:)' MM(2,:)'];
hd = HausdorffDist(A,M)
xlabel('z (in number of US frames)', 'FontSize', 12)
ylabel('u (in pixels)', 'FontSize', 12)
zlabel('v (in pixels)', 'FontSize', 12)
axis square equal
```

B.5. MATLAB Codes for Section 4.5

B.5.1: Converting US image from pixels to millimeters

```
% **************************************************************************
%                    US images from pixels to mm
%                Written By: Niloufaralsadat Hashemi
%                    Last Updated: June 9, 2018
%                        Libraries used:
%             MATLAB Computer Vision, MATLAB Image Processing
%                        ULTRASONIX SDK
% **************************************************************************

% Reading the data
[Data, Header] = RPread('test3.b8', 241);

% Creating a new plot
hF = figure(1);
hA = axes;

% Plotting the first frame
plot_SonixRP(Data(:, :, 1), Header , [hA 1], 1);

% The lateral length of the L14-5W/60 linear transducer
lateralTotalWidth_mm=0.46*128

%The lateral size in relation to the header is the total elements times
%each pitch then divided by the header size listed
lateralSinglePixel=lateralTotalWidth_mm/(Header.ur(1)-Header.ul(1)+1)

axialSinglePixel = lateralSinglePixel

axialTotalLength_mm = axialSinglePixel*(Header.bl(2)-Header.ul(2)+1)

Iin = Data(:,:,120);
Iin = Iin(82:578,142:473);

%  Converting axes to mm
idisp(Iin)
conversion=axialSinglePixel; % in mm/pixel
addMM=@(x) sprintf('%.2fmm',x*conversion);
xticklabels(cellfun(addMM,num2cell(xticks'),'UniformOutput',false));
yticklabels(cellfun(addMM,num2cell(yticks'),'UniformOutput',false));

%  Resizing the Image to mm
Iout = imageresize(Iin,axialTotalLength_mm,lateralTotalWidth_mm,2); % cubic interpolation
figure,
idisp(Iout)
```

B.5.2: Deriving the 3D rigid transformation between $F^{\{I\}}$ and $F^{\{PH\}}$

```
% ***************************************************************************
%                        3D Rigid Transformation
%                  Written By: Niloufaralsadat Hashemi
%                       Last Updated: June 9, 2018
% ***************************************************************************
% F_I: points measured in the US image frame
% F_PH: Points measured in fCal2.0 phantom frame
% N: Total number of points measured which is the same in {I} and {PH}
function [T, rmse] = Rigid3D(F_I, F_PH, N)
% Center of mass of each dataset
    u_F_I  = mean(F_I);
    u_F_PH = mean(F_PH);
% Calculating the cross covariance matrix
    cov = 0;
    for i=1:N
        H=(F_PH(i,:)-u_F_PH)'*(F_I(i,:)-u_F_I);
        cov = cov + H;
    end
    cov = (1/N) * cov;

% Calculating the anti-symmetric matrix
    A = cov - cov';
    Delta = [A(2,3) A(3,1) A(1,2)]';

% Calculating the 4x4 matrix
% of which eigenvalues and eigenvectors are calculated
    Q = [
        trace(cov)     Delta'
        Delta          cov+cov'-trace(cov)*eye(3)
        ];

    [V, D] = eig(Q);

% Deriving the quaternion rotaation vector
    q = V(:,4);
    q0 = q(1);
    q1 = q(2);
    q2 = q(3);
    q3 = q(4);

% Calculating the rotation matrix that relates {PH} to {I}
    R = [
        q0^2+q1^2-q2^2-q3^2  2*(q1*q2-q0*q3)      2*(q1*q3+q0*q2)
        2*(q1*q2+q0*q3)      q0^2+q2^2-q1^2-q3^2  2*(q2*q3-q0*q1)
        2*(q1*q3-q0*q2)      2*(q2*q3+q0*q1)      q0^2+q3^2-q1^2-q2^2
        ];
% Calculating translation vector that relates {PH} to {I}
    t = u_F_I' - R*(u_F_PH)';

%  Calculating the 3D rigid transformation matrix
```

```
    T = [
        R           t
        zeros(1,3)   1
        ];

% Calculating RSME
rmse = 0;
for i=1:N
    err = norm((F_I(i,:))'-R*(F_PH(i,:))'-t);
    err = err^2;
    rmse = rmse+err;
end
rmse = (1/(N))*rmse;
end
```

B.6. MATLAB Codes for Section 5.1

B.6.1: Importing, cropping, and thresholding of US images and equivalent ellipse/square

```
% ************************************************************************
%                Importing and cropping of US images &
%       Catheter cross section segmentation and equivalent ellipse/square
%                Written By: Niloufaralsadat Hashemi
%                     Last Updated: June 9, 2018
%                          Libraries used:
%            MATLAB Computer Vision, MATLAB Image Processing
%                Peter Cork Computer Vision Toolbox [128]
% ************************************************************************
% Straight catheter:
% Importing .b8 US file
[Data1, Header1] = RPread('straight001.b8', 209);
% cropping the region of interest using Header1 info
Data1 = Data1(82:578,142:473,1:176);
Num1 = 176; % number of useful frames in the file
T1 = 130; % The threshold value
% converting pixel-based image to mm-based image
lateralTotalWidth_mm=0.46*128;
lateralSinglePixel=lateralTotalWidth_mm/(Header1.ur(1)-Header1.ul(1)+1);
axialSinglePixel = lateralSinglePixel;
axialTotalLength_mm = axialSinglePixel*(Header1.bl(2)-Header1.ul(2)+1);
% Thresholding and estimating an ellipse for catheter cross-section
for i=1:num
    Data1_mm(:,:,i) = imageresize(Data1(:,:,i),axialTotalLength_mm*10,lateralTotalWidth_mm*10,2);
    thresholdedF(:,:,i) = Data1_mm(:,:,i)>T1;
    blobAnalysis1 = iblobs(thresholdedF(:,:,i));
    [section1 index1] = max(blobAnalysis1(2:end).area);
    radius1X = blobAnalysis1(index1+1).a; % major axis length of equivalent ellipse
    radius1Y = blobAnalysis1(index1+1).b; % minor axis length of equivalent ellipse
    center1Y = blobAnalysis1(index1+1).uc; % centroid, horizontal coordinate
    center1X = blobAnalysis1(index1+1).vc; % centroid, vertical coordinate
    imageSize1X = 588;
    imageSize1Y = 393;
    [columnsInImage1 rowsInImage1] = meshgrid(1:imageSize1X, 1:imageSize1Y);
    ellipsePixels1 = (rowsInImage1 - center1Y).^2 ./ radius1Y^2 ...
    + (columnsInImage1 - center1X).^2 ./ radius1X^2 <= 1;
    straight1{i} = ellipsePixels1;
    out{i} = straight1{i};
end

% Curvy catheter:
% Importing .b8 US file
[Data2, Header2] = RPread('curvy005.b8', 257);
% cropping the region of interest using Header1 info
Data2 = Data2(82:578,142:473,1:252);
num2 =252; % number of useful frames in the file
T2 = 120; % The threshold value
% converting pixel-based image to mm-based image
lateralTotalWidth_mm=0.46*128;
lateralSinglePixel=lateralTotalWidth_mm/(Header2.ur(1)-Header2.ul(1)+1);
axialSinglePixel = lateralSinglePixel;
axialTotalLength_mm = axialSinglePixel*(Header2.bl(2)-Header2.ul(2)+1);
% Thresholding and estimating an ellipse for catheter cross-section
for i=1:num2
    Data2_mm(:,:,i)= imageresize(Data2(:,:,i),axialTotalLength_mm*10,lateralTotalWidth_mm*10,2);
    thresholdedF(:,:,i) = Data2_mm(:,:,i)>T2;
    blobAnalysis2 = iblobs(thresholdedF(:,:,i));
    [section2 index2] = max(blobAnalysis2(2:end).area);
    radius2X = blobAnalysis2(index2+1).a; % major axis length of equivalent ellipse
    radius2Y = blobAnalysis2(index2+1).b; % minor axis length of equivalent ellipse
    center2Y = blobAnalysis2(index2+1).uc; % centroid, horizontal coordinate
    center2X = blobAnalysis2(index2+1).vc; % centroid, vertical coordinate
    imageSize2X = 588;
    imageSize2Y = 393;
    [columnsInImage2 rowsInImage2] = meshgrid(1:imageSize2X, 1:imageSize2Y);
    ellipsePixels2 = (rowsInImage2 - center2Y).^2 ./ radius2Y^2 ...
    + (columnsInImage2 - center2X).^2 ./ radius2X^2 <= 1;
    curvy5{i} = ellipsePixels2;
    out{i} = curvy5{i};
end

% planar catheter:
% Importing .b8 US file
[Data3, Header3] = RPread('planar009.b8', 346);
% cropping the region of interest using Header1 info
Data3 = Data3(82:578,142:473,1:346);
num3 = 346; % number of useful frames in the file
T3 = 120; % The threshold value
% converting pixel-based image to mm-based image
lateralTotalWidth_mm=0.46*128;
lateralSinglePixel=lateralTotalWidth_mm/(Header3.ur(1)-Header3.ul(1)+1);
axialSinglePixel = lateralSinglePixel;
axialTotalLength_mm = axialSinglePixel*(Header3.bl(2)-Header3.ul(2)+1);
```

```
% Thresholding and estimating an ellipse for catheter cross-section
for i=1:num3
    Data3_mm(:,:,i)= imageresize(Data3(:,:,i),axialTotalLength_mm*10,lateralTotalWidth_mm*10,2);
    thresholdedF(:,:,i) = Data3_mm(:,:,i)>T3;
    blobAnalysis3 = iblobs(thresholdedF(:,:,i));
    [section3 index3] = max(blobAnalysis3(2:end).area);
    radius3X = blobAnalysis3(index3+1).a; % major axis length of equivalent ellipse
    radius3Y = blobAnalysis3(index3+1).b; % minor axis length of equivalent ellipse
    center3Y = blobAnalysis3(index3+1).uc; % centroid, horizontal coordinate
    center3X = blobAnalysis3(index3+1).vc; % centroid, vertical coordinate
    imageSize3X = 588;
    imageSize3Y = 393;
    [columnsInImage3 rowsInImage3] = meshgrid(1:imageSize3X, 1:imageSize3Y);
    ellipsePixels3 = (rowsInImage3 - center3Y).^2 ./ radius3Y^2 ...
    + (columnsInImage3 - center3X).^2 ./ radius3X^2 <= 1;
    planar9{i} = ellipsePixels3;
    out{i} = planar9{i};
end

% unplanar catheter:
% Importing .b8 US file
[Data4, Header4] = RPread('unplanar007.b8', 402);
% cropping the region of interest using Header1 info
Data4 = Data4(82:578,142:473,1:363);
num4 = 363; % number of useful frames in the file
T4 = 140; % The threshold value
% converting pixel-based image to mm-based image
lateralTotalWidth_mm=0.46*128;
lateralSinglePixel=lateralTotalWidth_mm/(Header4.ur(1)-Header4.ul(1)+1);
axialSinglePixel = lateralSinglePixel;
axialTotalLength_mm = axialSinglePixel*(Header4.bl(2)-Header4.ul(2)+1);
% Thresholding and estimating an ellipse for catheter cross-section
for i=1:num4
    Data4_mm(:,:,i) = imageresize(Data4(:,:,i),axialTotalLength_mm*10,lateralTotalWidth_mm*10,2);
    thresholdedF(:,:,i) = Data4_mm(:,:,i)>T4;
    blobAnalysis4 = iblobs(thresholdedF(:,:,i));
    [section4 index4] = max(blobAnalysis4(2:end).area);
    radius4X = blobAnalysis4(index4+1).a; % major axis length of equivalent ellipse
    radius4Y = blobAnalysis4(index4+1).b; % minor axis length of equivalent ellipse
    center4Y = blobAnalysis4(index4+1).uc; % centroid, horizontal coordinate
    center4X = blobAnalysis4(index4+1).vc; % centroid, vertical coordinate
    imageSize4X = 588;
    imageSize4Y = 393;
    [columnsInImage4 rowsInImage4] = meshgrid(1:imageSize4X, 1:imageSize4Y);
    ellipsePixels4 = (rowsInImage4 - center4Y).^2 ./ radius4Y^2 ...
    + (columnsInImage4 - center4X).^2 ./ radius4X^2 <= 1;
    unplanar7{i} = ellipsePixels4;
    out{i} = unplanar7{i};
end
```

B.7. MATLAB Codes for Sections 5.2-5.9

B.7.1: KF, AKF and PF results and accuracy - Uncalibrated

```
% ***************************************************************************
%           Uncalibrated KF, AKF and PF results and accuracy
%               Written By: Niloufaralsadat Hashemi
%                   Last Updated: June 9, 2018
%                       Libraries used:
%            MATLAB Computer Vision, MATLAB Image Processing
%               Peter Cork Computer Vision Toolbox [128]
% ***************************************************************************
% Uncalibrated KF Results
figure(1),
hold on,
grid on
axis equal
plot3(predictions_KF(5,:),predictions_KF(1,:)/10, predictions_KF(2,:)/10,'r-')
plot3(measurements_KF(3,:),measurements_KF(1,:)/10, measurements_KF(2,:)/10,'b-')
xlabel('z (in number of US frames)', 'FontSize', 12)
ylabel('Lateral Distance (in mm)', 'FontSize', 12)
zlabel('Axial Distance (in mm)', 'FontSize', 12)
UnCal_RMSE_KF = rmse(predictions_KF,measurements_KF);
UnCal_hd_KF = HausdorffDist(predictions_KF,measurements_KF)

% Uncalibrated AKF Results
figure(2),
hold on,
grid on
axis equal
plot3(predictions_AKF(5,2:end),predictions_AKF(1,2:end)/10, predictions_AKF(2,2:end)/10,'r-')
plot3(measurements_AKF(3,:),measurements_AKF(1,:)/10, measurements_AKF(2,:)/10,'b-')
xlabel('z (in number of US frames)', 'FontSize', 12)
ylabel('Lateral Distance (in mm)', 'FontSize', 12)
zlabel('Axial Distance (in mm)', 'FontSize', 12)
UnCal_RMSE_AKF = rmse(predictions_AKF,measurements_AKF);
UnCal_hd_AKF = HausdorffDist(predictions_AKF,measurements_AKF);

% Uncalibrated PF Results
figure(3),
hold on,
grid on
axis equal
plot3([1:size(Pred,2)],Pred(1,:)/10, Pred(2,:)/10,'r-')
plot3([1:size(Meas,2)],Meas(1,:)/10, Meas(2,:)/10,'b-')
xlabel('z (in number of US frames)', 'FontSize', 12)
ylabel('Lateral Distance (in mm)', 'FontSize', 12)
zlabel('Axial Distance (in mm)', 'FontSize', 12)
UnCal_RMSE_PF = rmse(Pred,Meas);
UnCal_hd_PF = HausdorffDist(Pred,Meas);
```

B.7.2: KF, AKF and PF results and accuracy - Calibrated

```
% ******************************************************************************************
%             Calibrated KF, AKF and PF results and accuracy
%                Written By: Niloufaralsadat Hashemi
%                     Last Updated: June 9, 2018
%                          Libraries used:
%            MATLAB Computer Vision, MATLAB Image Processing
%                Peter Cork Computer Vision Toolbox [128]
% ******************************************************************************************
% CMM Measurements of the straight catheter
CMM_X_straight = [122.176
                 133.564
                 146.945
                 162.546
                 173.158];

CMM_Y_straight = [95.811
                  83.375
                  70.524
                  53.379
                  42.63];

CMM_Z_straight = [-201.166
                   -199.218
                   -196.954
                   -194.341
                   -192.892];

CMM_Z_straight = CMM_Z_straight - 1.5;
CMM_straight = [CMM_X_straight';CMM_Y_straight';CMM_Z_straight'];

% CMM Measurements of the curvy catheter
CMM_X_curvy = [188.767
              197.314
              202.677
              205.994
              210.216
              214.59
              220.584
              228.11
              233.409
              243.547
              248.648
              251.905
              253.606];

CMM_Y_curvy = [158.893
              150.315
              144.654
              140.192
              135.707
              131.495
              124.11
              117.17
              110.656
               99.733
               93.839
               91.28
               88.256];

CMM_Z_curvy = [-198.709
              -199.762
              -202.582
              -204.04
              -202.469
              -201.685
              -202.51
              -204.945
              -208.844
              -209.95
              -208.907
              -211.171
              -210.73];

CMM_Z_curvy = CMM_Z_curvy - 1.5;
CMM_curvy = [CMM_X_curvy';CMM_Y_curvy';CMM_Z_curvy'];

% CMM Measurements of the planar catheter
CMM_X_planar = [194.565
               205.265
               223.04
               235.863
               244.408
               260.952
               269.077
               273.078];

CMM_Y_planar = [158.233
               142.66
               120.005
               103.889
                98.344
                97.692
               101.047
               100.18];

CMM_Z_planar = [-203.682
               -203.73
               -202.429
               -202.736
               -201.629
               -199.939
               -198.416
               -197.495];

CMM_Z_planar = CMM_Z_planar - 1.5;
CMM_planar = [CMM_X_planar';CMM_Y_planar';CMM_Z_planar'];

% CMM Measurements of the unplanar catheter
CMM_X_unplanar = [132.193
                 142.048
                 152.385
                 156.197
                 165.803
                 167.176
                 173.016
                 176.657
                 176.26
                 181.04
                 187.086
                 191.175
                 200.426
                 207.445];

CMM_Y_unplanar = [113.21
                 106.702
                  91.852
                  87.421
                  74.045
                  71.622
                  63.226
                  59.793
                  58.448
                  57.888
                  57.109
                  56.909
                  56.826
                  59.818];

CMM_Z_unplanar = [-205.969
                 -204.921
                 -205.204
                 -204.879
                 -205.128
                 -205.156
                 -204.02
                 -205
                 -204.158
                 -204.402
                 -203.696
                 -201.958
                 -194.215
                 -186.922];

 CMM_Z_unplanar = CMM_Z_unplanar - 1.5;
 CMM_unplanar = [CMM_X_unplanar';CMM_Y_unplanar';CMM_Z_unplanar'];

 num = size(out,2);
% KF Calibrated Results
T_probe_wrt_Image = [
-0.3672    -0.9172    -0.1542     25.5202
-0.0457     0.1834    -0.9820    -97.6940
 0.9290    -0.3535    -0.1094    -13.4112
 0          0          0          1
];
T_IwrtP = (T_probe_wrt_Image)^-1;

% % Uncomment for Straight Catheter NDI Optotracker Sensor Readings
% % units: deg/mm
% X = xlsread('straight001.xlsx','S616:S791');
% Y = xlsread('straight001.xlsx','T616:T791');
% Z = xlsread('straight001.xlsx','U616:U791');
% Rx = xlsread('straight001.xlsx','R616:R791');
% Ry = xlsread('straight001.xlsx','Q616:Q791');
% Rz = xlsread('straight001.xlsx','P616:P791');

% % Uncomment for curvy Catheter NDI Optotracker Sensor Readings
% % units: deg/mm-->251 frames
% X = xlsread('curvy005.xlsx','S651:S902');
% Y = xlsread('curvy005.xlsx','T651:T902');
% Z = xlsread('curvy005.xlsx','U651:U902');
% Rx = xlsread('curvy005.xlsx','R651:R902');
% Ry = xlsread('curvy005.xlsx','Q651:Q902');
% Rz = xlsread('curvy005.xlsx','P651:P902');

% % Uncomment for planar Catheter NDI Optotracker Sensor Readings
% % units: deg/mm -->338
% X = xlsread('planar009.xlsx','S609:S947');
% Y = xlsread('planar009.xlsx','T609:T947');
% Z = xlsread('planar009.xlsx','U609:U947');
% Rx = xlsread('planar009.xlsx','R609:R947');
% Ry = xlsread('planar009.xlsx','Q609:Q947');
% Rz = xlsread('planar009.xlsx','P609:P947');

% % Uncomment for unplanar Catheter NDI Optotracker Sensor Readings
% % units: deg/mm -->389
% X = xlsread('unplanar007.xlsx','S790:S1179');
% Y = xlsread('unplanar007.xlsx','T790:T1179');
% Z = xlsread('unplanar007.xlsx','U790:U1179');
% Rx = xlsread('unplanar007.xlsx','R790:R1179');
% Ry = xlsread('unplanar007.xlsx','Q790:Q1179');
% Rz = xlsread('unplanar007.xlsx','P790:P1179');

KF_m_PinI = [
    measurements_KF(1,:)./10 % X
    measurements_KF(2,:)./10 % Y
    zeros(1,size(measurements_KF,2))
    ones(1,size(measurements_KF,2))];
KF_p_PinI = [
    predictions_KF(1,:)./10 % X
    predictions_KF(2,:)./10 % Y
    zeros(1,size(predictions_KF,2))
    ones(1,size(predictions_KF,2))];
KF_m_PinR = zeros(4,size(measurements_KF,2));
KF_p_PinR = zeros(4,size(predictions_KF,2));

AKF_m_PinI = [
    measurements_AKF(1,:)./10 % X
    measurements_AKF(2,:)./10 % Y
    zeros(1,size(measurements_AKF,2))
    ones(1,size(measurements_AKF,2))];
AKF_p_PinI = [
    predictions_AKF(1,:)./10 % X
    predictions_AKF(2,:)./10 % Y
    zeros(1,size(predictions_AKF,2))
    ones(1,size(predictions_AKF,2))];
AKF_m_PinR = zeros(4,size(predictions_AKF,2));
AKF_p_PinR = zeros(4,size(predictions_AKF,2));

PF_m_PinI = [
    Meas(1,:)./10 % X
    Meas(2,:)./10 % Y
    zeros(1,size(Meas,2))
    ones(1,size(Meas,2))];
PF_p_PinI = [
    330.8659/10  Pred(1,:)./10 % X
    307.0230/10  Pred(2,:)./10 % Y
    zeros(1,size(Pred,2))
    ones(1,size(Pred,2))];
PF_m_PinR = zeros(4,size(Meas,2));
PF_p_PinR = zeros(4,size(Pred,2));

for i=1:num
    T_PwrtR(:,:,i) = transl(X(i),Y(i),Z(i))*trotx(Rx(i),'deg')*troty(Ry(i),'deg')*trotz(Rz(i),'deg');
    T_IwrtR(:,:,i) =  T_PwrtR(:,:,i) * T_IwrtP;
    IPose(:,:,i) = T_PwrtR(:,:,i) * transl(-15,22,-117)*troty(90,'deg')*trotz(-90,'deg');
    KF_m_PinR(:,i) =  T_IwrtR(:,:,i) * KF_m_PinI(:,i);
    KF_p_PinR(:,i) =  T_IwrtR(:,:,i) * KF_p_PinI(:,i);
    KF_m(:,i) =  IPose(:,:,i) * KF_m_PinI(:,i);
    KF_p(:,i) =  IPose(:,:,i) * KF_p_PinI(:,i);
    AKF_m_PinR(:,i) =  T_IwrtR(:,:,i) * AKF_m_PinI(:,i);
    AKF_p_PinR(:,i) =  T_IwrtR(:,:,i) * AKF_p_PinI(:,i);
    AKF_m(:,i) =  IPose(:,:,i) * AKF_m_PinI(:,i);
    AKF_p(:,i) =  IPose(:,:,i) * AKF_p_PinI(:,i);
    PF_m_PinR(:,i) =  T_IwrtR(:,:,i) * PF_m_PinI(:,i);
    PF_p_PinR(:,i) =  T_IwrtR(:,:,i) * PF_p_PinI(:,i);
    PF_m(:,i) =  IPose(:,:,i) * PF_m_PinI(:,i);
    PF_p(:,i) =  IPose(:,:,i) * PF_p_PinI(:,i);
```

```
end

figure(1),
grid on
hold on
for i=1:10:num
    trplot(T_PwrtR(:,:,i), 'length',20)
    trplot(T_IwrtR(:,:,i), 'length',20, 'color', 'r')
    trplot(IPose(:,:,i), 'length',20, 'color', 'g')
end

  plot3(KF_p_PinR(1,1:end), KF_p_PinR(2,1:end), KF_p_PinR(3,1:end), 'r-')
  plot3(KF_m_PinR(1,1:end), KF_m_PinR(2,1:end), KF_m_PinR(3,1:end), 'b-')

  plot3(KF_p(1,1:end), KF_p(2,1:end), KF_p(3,1:end), 'r-')
  plot3(KF_m(1,1:end), KF_m(2,1:end), KF_m(3,1:end), 'b-')

  plot3(AKF_p_PinR(1,2:end), AKF_p_PinR(2,:), AKF_p_PinR(3,:), 'r-')
  plot3(AKF_m_PinR(1,2:end), AKF_m_PinR(2,:), AKF_m_PinR(3,:), 'b-')

  plot3(AKF_p(1,1:end), AKF_p(2,1:end), AKF_p(3,1:end), 'r-')
  plot3(AKF_m(1,1:end), AKF_m(2,1:end), AKF_m(3,1:end), 'b-')

  plot3(PF_p_PinR(1,:), PF_p_PinR(2,:), PF_p_PinR(3,:), 'r-')
  plot3(PF_m_PinR(1,:), PF_m_PinR(2,:), PF_m_PinR(3,:), 'b-')

  plot3(PF_p(1,:), PF_p(2,:), PF_p(3,:), 'r-')
  plot3(PF_m(1,:), PF_m(2,:), PF_m(3,:), 'b-')
% % Uncomment for the straight catheter
% plot3(CMM_X_straight, CMM_Y_straight, CMM_Z_straight)
% % Uncomment for the curvy catheter
% plot3(CMM_X_curvy, CMM_Y_curvy, CMM_Z_curvy)
% % Uncomment for the planar catheter
% plot3(CMM_X_planar, CMM_Y_planar, CMM_Z_planar)
% % Uncomment for the unplanar catheter
% plot3(CMM_X_unplanar, CMM_Y_unplanar, CMM_Z_unplanar)
plot3(0,0,0, 'g*')
xlabel('X')
ylabel('Y')
zlabel('Z')

% Error Calculations
% % Uncomment for straight catheter:
% Cal_hd_KF_v1 = HausdorffDist(KF_p_PinR,CMM_straight);
% Cal_hd_AKF_v1 = HausdorffDist(AKF_p_PinR,CMM_straight);
% Cal_hd_PF_v1 = HausdorffDist(PF_p_PinR,CMM_straight);
% Cal_hd_KF_v2 = HausdorffDist(KF_p,CMM_straight);
% Cal_hd_AKF_v2 = HausdorffDist(AKF_p,CMM_straight);
% Cal_hd_PF_v2 = HausdorffDist(PF_p,CMM_straight);
% % Uncomment for curvy catheter:
% Cal_hd_KF_v1 = HausdorffDist(KF_p_PinR,CMM_curvy);
% Cal_hd_AKF_v1 = HausdorffDist(AKF_p_PinR,CMM_curvy);
% Cal_hd_PF_v1 = HausdorffDist(PF_p_PinR,CMM_curvy);
% Cal_hd_KF_v2 = HausdorffDist(KF_p,CMM_curvy);
% Cal_hd_AKF_v2 = HausdorffDist(AKF_p,CMM_curvy);
% Cal_hd_PF_v2 = HausdorffDist(PF_p,CMM_curvy);
% % Uncomment for planar catheter:
% Cal_hd_KF_v1 = HausdorffDist(KF_p_PinR,CMM_planar);
% Cal_hd_AKF_v1 = HausdorffDist(AKF_p_PinR,CMM_planar);
% Cal_hd_PF_v1 = HausdorffDist(PF_p_PinR,CMM_planar);
% Cal_hd_KF_v2 = HausdorffDist(KF_p,CMM_planar);
% Cal_hd_AKF_v2 = HausdorffDist(AKF_p,CMM_planar);
% Cal_hd_PF_v2 = HausdorffDist(PF_p,CMM_planar);
% % Uncomment for unplanar catheter:
% Cal_hd_KF_v1 = HausdorffDist(KF_p_PinR,CMM_unplanar);
% Cal_hd_AKF_v1 = HausdorffDist(AKF_p_PinR,CMM_unplanar);
% Cal_hd_PF_v1 = HausdorffDist(PF_p_PinR,CMM_unplanar);
% Cal_hd_KF_v2 = HausdorffDist(KF_p,CMM_unplanar);
% Cal_hd_AKF_v2 = HausdorffDist(AKF_p,CMM_unplanar);
% Cal_hd_PF_v2 = HausdorffDist(PF_p,CMM_unplanar);
```

BIBLIOGRAPHY

[1] J. Burgner-Kahrs et al., “Continuum robots for medical applications: A Survey,” *IEEE Trans. Robot.,* vol. 31, no. 6, pp. 1261–1280, 2015.

[2] H. B. Gilbert et al., “Concentric tube robots: The state of the art and future directions,” in Inaba M., Corke P. (eds) *Robotics Research. Springer Tracts in Advanced Robotics*, vol 114, pp. 253–269, 2016.

[3] C.Bergeles et al, “Concentric tube robot design and optimization based on task and anatomical constraints,” *IEEE Trans. Robot*., vol. 31, no. 1, pp 67-84, 2015.

[4] T. Anor et al., “Algorithms for design of continuum robots using the concentric tubes approach: A neurosurgical example,” in *Proc. IEEE Int. Conf. Robot. Autom*., pp. 667–673, Shanghai, China, 2011.

[5] J. Burgner et al., “Debulking from within: A robotic steerable cannula for intracerebral hemorrhage evacuation,” *IEEE Trans. Biomed. Eng*., vol. 60, no. 9, pp. 2567–75, 2013.

[6] J. Burgner et al., “A telerobotic system for transnasal surgery,” *IEEE/ASME Trans. Mechatronics*, vol. 19, no. 3, pp. 996–1006, 2014.

[7] A. Bajo et al., “Robotic-assisted micro-surgery of the throat: the trans-nasal approach,” in *Proc. IEEE Int. Conf. Robot. Autom*., pp. 232–238, Bajo, Andrea, Karlsruhe, Germany, 2013.

[8] H. Yu et al., “Design, calibration and preliminary testing of a robotic telemanipulator for OCT guided retinal surgery,” in *Proc. IEEE Int. Conf. Robot. Autom*., pp. 225–231, Karlsruhe, Germany, 2013.

[9] Christos Bergeles, “Keeping an eye on robotics”, [online] Aop.org.uk. Available at: https://www.aop.org.uk/ot/CET/2016/11/14/keeping-an-eye-on-robotics/article [Accessed 1 Oct. 2017].

[10] Loschak et al., “Algorithms for automatically pointing ultrasound imaging Catheters,” in *IEEE Trans. Robot.*, vol. 33, no. 1, 2017.

[11] A. H. Gosline et al., “Percutaneous intracardiac beating-heart surgery using metal MEMS tissue approximation tools,” *Int. J. Robot. Res.,* vol. 31, no. 9, pp. 1081–1093, 2012.

[12] D. Thomas et al., “Initial experience with robotic navigation for catheter ablation of paroxysmal and persistent atrial fibrillation,” *J. Electrocardiol.,* vol. 45, no. 2, pp. 95–101, 2012.

[13] S. B. Kesner et al., “Position control of motion compensation cardiac catheters,” *IEEE Trans. Robot.*, vol. 27, no. 6, pp. 1045–1055, 2011.

[14] L. Swanstrom et al., “Development of a new access device for transgastric surgery,” *J. Gastrointestinal Surgery*, vol. 9, no. 8, pp. 1129–1137, 2005.

[15] P. Swain, “The ShapeLock system adapted to intragastric and transgastric surgery”, *Endoscopy*, vol. 39, no. 5, pp. 466–470, 2007.

[16] Y. Fu et al., “Steerable catheters in minimally invasive vascular surgery,” *Int. J. Med. Robot. Comput. Assisted Surgery*, vol. 5, no. 4, pp. 381–391, 2009.

[17] J. Guo et al, "Design and performance evaluation of a novel robotic catheter system for vascular interventional surgery," *Microsystem Technologies*, vol. 22, no. 9, pp. 2167-2176, 2016.

[18] R. E. Goldman et al., “Design and performance evaluation of a minimally invasive telerobotic platform for transurethral surveillance and intervention,” *IEEE Trans. Biomed. Eng.*, vol. 60, no. 4, pp. 918–25, 2013.

[19] R. J. Hendrick et al., “A multi-arm handheld robotic system for transurethral laser prostate surgery,” in *Proc. IEEE Int. Conf. Robot. Autom.*, pp. 2850–2855, Hong Kong, China, 2014.

[20] Shi et al., “Shape sensing techniques for continuum robots in minimally invasive surgery: a survey,” in *IEEE Trans. Biomed. Eng.*, vol. 64, no. 8, 2017.

[21] E. J. Butler et al., “Robotic neuro-endoscope with concentric tube augmentation,” in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems* (IROS ’12), pp. 2941–2946, Vilamoura, Portugal, 2012.

[22] L. G. Torres et al., “Task oriented design of concentric tube robots using mechanics based models,” in *Proc. IEEE/RSJ Int. Con. Intelligent Robots and Systems* (IROS ’12), pp. 4449–4455, Vilamoura, Portugal, 2012.

[23] M. Dalvand et al, “Fast vision-based catheter 3D reconstruction” in *Phys. Med. Biol.*, vol. 61, no. 14, pp. 5128–5148, 2016.

[24] A. Vandini et al, “Unified tracking and shape estimation for concentric tube robots,” in *IEEE Trans. Robot.*, vol. 33, no. 4, pp. 901-915, 2017.

[25] F. Taffoni et al., “Optical fiber-based MR-compatible sensors for medical applications: an overview,” *Sensors*, vol. 13, no. 10, pp. 14105–14120, 2013.

[26] S. Hasanzadeh, "Modeling, force estimation and control of steerable catheters for robot-assisted intra-cardiac navigation." 2016.

[27] C. Tercero et al, "Catheter insertion reference trajectory construction method using photoelastic stress analysis for quantification of respect for tissue during endovascular surgery simulation," *Int. J. Optomechatronics*, vol. 5, no. 4, pp. 322, 2011.

[28] A. Abushagur et al., "Advances in bio-tactile sensors for minimally invasive surgery using the fibre Bragg grating force sensor technique: A survey," *Sensors*, vol. 14, no. 4, pp. 6633–6665, 2014.

[29] S. Song et al., "Electromagnetic positioning for tip tracking and shape sensing of flexible robots," *IEEE Sensors J.*, vol. 15, no. 8, pp. 4565–4575, 2015.

[30] A. Vandini et al., "Vision-based intraoperative shape sensing of concentric tube robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, pp. 2603–2610, Hamburg, Germany, 2015.

[31] E. J. Lobaton et al., "Continuous shape estimation of continuum robots using X-ray images," in *Proc. IEEE Int. Conf. Robot. Autom.*, pp. 725–732, Karlsruhe, Germany, 2013.

[32] B. Kim et al., "Optimizing curvature sensor placement for fast, accurate shape sensing of continuum robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, pp. 5374–5379, Conference Location, 2014.

[33] R. Roesthuis et al., "On using an array of fiber Bragg grating sensors for closed-loop control of flexible minimally invasive surgical instruments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst*., pp. 2545–2551, Tokyo, Japan, 2013.

[34] A. Vandini et al., "Vision-based motion control of a flexible robot for surgical applications," in *Proc. IEEE Int. Conf. Robot. Autom.*, pp. 6205-6211, Hong Kong, China, 2014.

[35] S. Nazarian et al, "Feasibility of real-time magnetic resonance imaging for catheter guidance in electrophysiology studies," *Circulation*, vol. 118, no. 3, pp. 223-229, 2008.

[36] Whiting et al., "Real-time MRI-guided catheter tracking using hyperpolarized silicon particles," in *Scientific Reports*, vol. 5, no. 1, pp. 12842, 2015.

[37] M.Wagner et al., "4D interventional device reconstruction from biplane fluoroscopy," *Med. Phys.*, vol. 43, no. 3, pp. 1324–1334, Mar. 2016.

[38] J. Burgner et al., "Toward fluoroscopic shape reconstruction for control of steerable medical devices," in *Proc. ASME Dyn. Syst. Control Conf. Bath/ASME Symp*. Fluid Power Motion Control, pp. 791–794, Virginia, USA, 2011.

[39] R. Reilink et al., "3D position estimation of flexible instruments: Markerless and marker-based methods," *Int. J. Comput. Assisted Radiol. Surg*., vol. 8, no. 3, pp. 407–417, 2013.

[40] C. Kim et al., “Robot for ultrasound-guided prostate imaging and intervention,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, pp. 943–948, San Francisco, CA, USA, 2011.

[41] C. Nadeau et al., “Intensity-based visual servoing for instrument and tissue tracking in 3d ultrasound volumes,” *IEEE Trans. Autom. Sci. Eng.*,vol. 12, no. 1, pp. 367–371, 2015.

[42] Salazar et al., “Ultrasound-guided peripheral venous access for therapeutic apheresis procedures reduces need for central venous catheters,” in *J. Clinical Apheresis*, vol. 32, no. 4, pp. 266-269, 2017.

[43] I. J. Nixon et al, "Nomogram for selecting thyroid nodules for ultrasound guided fine needle aspiration biopsy based on a quantification of risk of malignancy," in *Head & Neck*, vol. 35, no. 7, pp. 1022-1025, 2013.

[44] N. Figueredo et al, "Accuracy of office-based ultrasound guided needle biopsies performed by dedicated breast surgeons," in *Cancer Research*, vol. 69, (24 Supplement), pp. 6022-6022, San Antonio, TX, 2009.

[45] J. W. Kim et al, "Ultrasound-guided percutaneous radiofrequency ablation of liver tumors: how we do it safely and completely," in *Korean Journal of Radiology*, vol. 16, no. 6, pp. 1226-1239, 2015.

[46] A. L. Trejos et al., “Robot-assisted minimally invasive lung brachytherapy,” *Int. J. Med. Robot.*, vol. 3, no. 1, pp. 41–51, 2007.

[47] J. Soverow et al., "Adoption of routine ultrasound guidance for femoral arterial access for cardiac catheterization," in *J. Invasive Cardiology*, vol. 28, no. 8, pp. 311, 2016.

[48] L. Zaremski et al, "Prospective comparison of palpation versus ultrasound-guided radial access for cardiac catheterization," in *J. of Invasive Cardiology*, vol. 25, no. 10, pp. 538, 2013.

[49] K. Gutleben et al, "Ultrasound in the electrophysiological cardiac catheterization laboratory," in *Herzschrittmachertherapie & Elektrophysiologie*, vol. 23, no. 4, pp. 260, 2012.

[50] J. Stoll et al., “Passive markers for tracking surgical instruments in real time 3-D ultrasound imaging,” *IEEE Trans. Med. Imag.*, vol. 31, no. 3, pp. 563–575, 2012.

[51] Z. Neubach et al., “Ultrasound-guided robot for flexible needle steering,” *IEEE Trans. Biomed. Eng.*, vol. 57, no. 4, pp. 799–805, 2010.

[52] K. Mathiassen et al, "Robust real-time needle tracking in 2-D ultrasound images using statistical filtering," in *IEEE Trans. Control Systems Technology*, vol. 25, no. 3, pp. 966-978, 2017.

[53] F. Chen et al., "3D catheter shape determination for endovascular navigation using a two-step

particle filter and ultrasound scanning," in *IEEE Trans. Medical Imaging*, vol. 36, no. 3, pp. 685-695, 2017.

[54] H. Ren et al., “Tubular structure enhancement for surgical instrument detection in 3D ultrasound,” in *Proc. IEEE Eng. Med. Biol. Soc. Annu. Int. Conf.*, pp. 7203–7206, Boston, MA, USA, 2011.

[55] M. Abayazid et al, "Integrating Deflection Models and Image Feedback for Real-Time Flexible Needle Steering," in *IEEE Trans. on Robotics*, vol. 29, no. 2, pp. 542-553, 2013.

[56] G. J. Vrooijink, M. Abayazid and S. Misra, "Real-time three-dimensional flexible needle tracking using two-dimensional ultrasound," in *Proc. IEEE Int. Conf. Robot. Autom.*, ICRA, Karlsruhe, Germany, 2013.

[57] M. Abayazid et al, "Experimental evaluation of ultrasound-guided 3D needle steering in biological tissue," in *Int. J. Computer Assisted Radiology and Surgery*, vol. 9, no. 6, pp. 931-939, 2014.

[58] M. Abayazid et al., "Three-dimensional needle steering towards a localized target in a prostate phantom," in *5th IEEE RAS/EMBS Int. Con. on Biomedical Robotics and Biomechatronics*, pp. 7-12, Sao Paulo, Brazil, 2014

[59] G. J. Vrooijink et al, "Needle path planning and steering in a three-dimensional non-static environment using two-dimensional ultrasound images," in *Int. J. of Robotics Research*, vol. 33, no. 10, pp. 1361-1374, 2014.

[60] P. Moreira et al, "Needle steering in biological tissue using ultrasound-based online curvature estimation," in *Proc. IEEE Int. Conf. Robot. Autom.* (ICRA), pp. 4368 – 4373, Hong Kong, China, 2014.

[61] M. Abayazid et al, "Ultrasound-guided three-dimensional needle steering in biological tissue with curved surfaces," in *Medical Engineering and Physics*, vol. 37, no. 1, pp. 145-150, 2015.

[62] P. Moreira et al., "Biomechanics-based curvature estimation for ultrasound-guided flexible needle steering in biological tissues," in *Annals of Biomedical Engineering*, vol. 43, no. 8, pp. 1716-1726, 2015.

[63] N. Shahriari et al, "Steering an actuated-tip needle in biological tissue: Fusing FBG-sensor data and ultrasound images," in *Proc. IEEE Int. Conf. Robot. Autom*. (ICRA), pp. 4443 – 4449, Stockholm, Sweden, 2016.

[64] M. Abayazid et al, "Experimental evaluation of co-manipulated ultrasound-guided flexible needle steering: Co-manipulated ultrasound-guided needle steering," in *Int. J. Medical Robotics and Computer Assisted Surgery*, vol. 12, no. 2, pp. 219-230, 2016.

[65] J. Stoll et al., "Ultrasound-based servoing of manipulators for telesurgery," in *Proc. SPIE* (Telemanipulator and Telepresence Technologies VIII), pp. 78 – 85, Stockholm, Sweden, 2002.

[66] Stoll et al, "Real-time 3D ultrasound-based servoing of a surgical instrument," in *Proc. IEEE Int. Conf. Robot. Autom.* (ICRA), pp. 613 - 618, Orlando, FL, USA, 2006.

[67] P. M. 50 et al, "Real-time visual servoing of a robot using three-dimensional ultrasound," in *Proc. IEEE Int. Conf. Robot. Autom.*, pp. 2655 – 2660, Roma, Italy, 2007.

[68] S. G. Yuen et al., "Quasiperiodic predictive filtering for robot-assisted beating heart surgery," in *Proc. IEEE Int. Conf. Robot. Autom*., pp. 3875 – 3880, Pasadena, CA, USA, 2008.

[69] S. G. Yuen et al, "Robotic motion compensation for beating heart intracardiac surgery," *Int. J. of Robotics Research*, vol. 28, no. 10, pp. 1355-1372, 2009.

[70] S. B Kesner et al., “Ultrasound servoing of catheters for beating heart valve repair” in *Information Processing in Computer-Assisted Interventions* vol. 6135, pp. 168–178, Berlin, Heidelberg, 2010.

[71] S. G. Yuen et al, "Robotic tissue tracking for beating heart mitral valve surgery," in *Medical Image Analysis*, vol. 17, no. 8, pp. 1236-1242, 2013.

[72] S. B. Kesner and R. D. Howe, "Robotic catheter cardiac ablation combining ultrasound guidance and force control," in *Int. J. Robotics Research*, vol. 33, no. 4, pp. 631-644, 2014.

[73] E. M. Boctor et al, "Robotically assisted intraoperative ultrasound with application to ablative therapy of liver cancer," in *Proc. of SPIE* (Medical Imaging 2003: Visualization, Image-Guided Procedures, and Display), pp. 281 - 291, San Diego, California, United States, 2003.

[74] E. M. Boctor et al, "Robotically assisted ablative treatment guided by freehand 3D ultrasound," in *Int. Congress Series*, vol. 1268, pp. 503-508, Baltimore, MD, USA, 2004.

[75] E. M. Boctor et al, "A dual-armed robotic system for intraoperative ultrasound guided hepatic ablative therapy: A prospective study," in *Proc. IEEE Int. Conf. Robot. Autom*. (ICRA), vol. 3, pp. 2517 – 2522, New Orleans, LA, USA, 2004

[76] E. M. Boctor et al, "Three-dimensional ultrasound-guided robotic needle placement: an experimental evaluation," in *Int. J. of Medical Robotics and Computer Assisted Surgery*, vol. 4, no. 2, pp. 180-191, 2008.

[77] Q. Ma et al, "A new robotic ultrasound system for tracking a catheter with an active piezoelectric element," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems* (IROS), pp. 2321 – 2328, Daejeon, South Korea, 2016.

[78] P. Chatelain et al., "Real-time needle detection and tracking using a visually servoed 3D ultrasound probe," in *Proc. IEEE Int. Conf. Robot. Autom*., pp. 1676 – 1681, Karlsruhe, Germany, 2013.

[79] P. Chatelain et al., "3D ultrasound-guided robotic steering of a flexible needle via visual servoing," in *Proc. IEEE Int. Conf. Robot. Autom*. (ICRA), pp. 2250 - 2255, Seattle, WA, USA, 2015.

[80] O. Zettinig et al, "Toward real-time 3D ultrasound registration-based visual servoing for interventional navigation," in *Proc. IEEE Int. Conf. Robot. Autom*. (ICRA), pp. 945 – 950, Stockholm, Sweden, 2016.

[81] R. Kojcev et al, "Dual-robot ultrasound-guided needle placement: closing the planning-imaging-action loop," in *Int. J. of Computer Assisted Radiology and Surgery*, vol. 11, no. 6, pp. 1173-1181, 2016.

[82] M. Khadem et al, "Semi-automated needle steering in biological tissue using an ultrasound-based deflection predictor," in *Annals of Biomedical Engineering*, vol. 45, no. 4, pp. 924-938, 2017.

[83] M. Waine et al, "Needle tracking and deflection prediction for robot-assisted needle insertion using 2D ultrasound images," in *J. of Medical Robotics Research*, vol. 1, no. 1, 2016.

[84] Z. Wei et al, "Robot-assisted 3D-TRUS guided prostate brachytherapy: System integration and validation," in *Medical Physics*, vol. 31, no. 3, pp. 539-548, 2004.

[85] T. Ortmaier et al, "Robust real-time instrument tracking in ultrasound images for visual servoing," in *Proc. IEEE Int. Conf. Robot. Autom*., pp. 2167 – 2172, Barcelona, Spain, 2005.

[86] M. Vitrani et al., "Automatic guidance of a surgical instrument with ultrasound based visual servoing," in *Proc. IEEE Int. Conf. Robot. Autom*., pp. 508 – 513, Barcelona, Spain, 2005.

[87] M. Vitrani et al, "A robust ultrasound-based visual servoing approach for automatic guidance of a surgical instrument with in vivo experiments," in *IEEE/RAS-EMBS Int. Conf. on Biomedical Robotics and Biomechatronics* (BioRob 2006), pp. 35 – 40, Pisa, Italy, 2006.

[88] M. Vitrani et al, "Robust ultrasound-based visual servoing for beating heart intracardiac surgery," in *Proc. IEEE Int. Conf. Robot. Autom*., pp. 3021 – 3027, Roma, Italy, 2007.

[89] J. S. Hong et al., “A motion adaptable needle placement instrument based on tumor specific ultrasonic image segmentation.” in *Medical Image Computing and Computer-Assisted Intervention* (MICCAI), vol. 2488, pp. 122-129, 2002.

[90] J. Hong et al, "An ultrasound-driven needle-insertion robot for percutaneous cholecystostomy," in *Physics in Medicine and Biology*, vol. 49, no. 3, pp. 441-455, 2004.

[91] M. Sauvée, P. Poignet and E. Dombre, "Ultrasound image-based visual servoing of a surgical instrument through nonlinear model predictive control," in *Int. J. of Robotics Research*, vol. 27, no. 1, pp. 25-40, 2008.

[92] M. Kaya et al, "Visual tracking of biopsy needles in 2D ultrasound images," in *Proc. IEEE Int. Conf. Robot. Autom.* (ICRA), pp. 4386 – 4391, Stockholm, Sweden, 2016.

[93] M. Kaya et al, "Visual tracking of multiple moving targets in 2D ultrasound guided robotic percutaneous interventions," in *Proc. IEEE Int. Conf. Robot. Autom*. (ICRA), pp. 1996, Singapore, 2017.

[94] T. K. Adebar et al., "Recursive estimation of needle pose for control of 3D-ultrasound-guided robotic needle steering," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 4303 – 4308, Chicago, IL, USA, 2014.

[95] J. Whitman et al, "Autonomous surgical robotics using 3-D ultrasound guidance: feasibility study," in *Ultrasonic Imaging*, vol. 29, no. 4, pp. 213-219, 2007.

[96] K. Liang et al, "Three-dimensional ultrasound guidance of autonomous robotic breast biopsy: feasibility study," in *Ultrasound in Medicine & Biology*, vol. 36, no. 1, pp. 173-177, 2010.

[97] K. Liang et al, "Simulation of autonomous robotic multiple-core biopsy by 3D ultrasound guidance," in *Ultrasonic Imaging*, vol. 32, no. 2, pp. 118-127, 2010.

[98] N. Hungr et al, "A 3-D ultrasound robotic prostate brachytherapy system with prostate motion tracking," in *IEEE Trans. on Robotics*, vol. 28, no. 6, pp. 1382-1397, 2012.

[99] J. Long et al, "Development of a novel robot for transperineal needle based interventions: focal therapy, brachytherapy and prostate biopsies," in *J. Urology,* the, vol. 188, no. 4, pp. 1369-1374, 2012.

[100] K. Mathiassen et al., "Visual servoing of a medical ultrasound probe for needle insertion," in *Proc. IEEE Int. Conf. Robot. Autom.* (ICRA), pp. 3426 – 3433, Stockholm, Sweden, 2016.

[101] A. I. Chen et al, "Real-time needle steering in response to rolling vein deformation by a 9-DOF image-guided autonomous venipuncture robot," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems* (IROS), pp. 2633 – 2638, Hamburg, Germany, 2015.

[102] M. L. Balter et al, "The System Design and Evaluation of a 7-DOF Image-Guided Venipuncture Robot," in *IEEE Trans. on Robotics*, vol. 31, no. 4, pp. 1044-1053, 2015.

[103] M. L. Balter et al, "Adaptive Kinematic Control of a Robotic Venipuncture Device Based on Stereo Vision, Ultrasound, and Force Guidance," in *IEEE Trans. on Industrial Electronics*, vol. 64, no. 2, pp. 1626-1635, 2017.

[104] S. McDermott et al., "Radiofrequency Ablation of Liver Tumors," in *Semin Intervent Radiol*, vol. 30, no. 1, pp. 049-055, 2013.

[105] M. M. Awad et al., “Radiofrequency Ablation of Liver Tumors”, pp. 409-415 [online], Available at: http://eknygos.lsmuni.lt/springer/533/409-415.pdf. [Accessed 10-May-2018].

[106] T. L. Szabo et al., "Ultrasound Transducer Selection in Clinical Imaging Practice," in J. of *Ultrasound in Medicine*, vol. 32, no. 4, pp. 573-582, 2013.

[107] A. Peltier et al, "3D versus 2D Systematic Transrectal Ultrasound-Guided Prostate Biopsy: Higher Cancer Detection Rate in Clinical Practice," in *Prostate Cancer*, vol. 2013, pp. 783243, 2013.

[108] Canadian Cancer Society, “Transrectal ultrasound (TRUS)”, 2018. [Online]. Available at: http://www.cancer.ca/en/cancer-information/diagnosis-and-treatment/tests-and-procedures/transrectal-ultrasound-trus/?region=on

[109] H. R. S. Neshat et al., "Real-time parametric curved needle segmentation in 3D ultrasound images," in *IEEE RAS & EMBS Inte. Conf. on Biomedical Robotics and Biomechatronics*, pp. 670 – 675, Scottsdale, AZ, USA, 2008.

[110] P. M. Novotny et al, "GPU based real-time instrument tracking with three-dimensional ultrasound," in *Medical Image Analysis*, vol. 11, no. 5, pp. 458-464, 2007.

[111] M. Aboofazeli et al, "A new scheme for curved needle segmentation in three-dimensional ultrasound images," in *IEEE Int. Symposium on Biomedical Imaging: From Nano to Macro*, pp. 1067 – 1070, Boston, MA, USA, 2009.

[112] M. J. Mack, "Minimally Invasive and Robotic Surgery," in *Jama*, vol. 285, no. 5, pp. 568-572, 2001.

[113] P. I. Corke, “Robotics, Vision and Control: Fundamental Algorithms in MATLAB”, chapter 12-15, pp. 285-479, 2011.

[114] Grewal M.S., “Kalman Filtering” in *Lovric M. (eds) Int. Encyclopedia of Statistical Science*, Springer, Berlin, Heidelberg, 2011.

[115] D. S. Minhas et al, "Modeling of needle steering via duty-cycled spinning," in 29th *Annu. Int. Conf. IEEE Engineering in Medicine and Biology Society*, pp. 2756 – 2759, Lyon, France, 2007.

[116] E. M. Boctor et al, "Virtual remote center of motion control for needle placement robots," *Computer Aided Surgery: Official Journal of the International Society for Computer Aided Surgery*, vol. 9, no. 5, pp. 175-183, 2004.

[117] H. Ren, N. V. Vasilyev and P. E. Dupont, "Detection of curved robots using 3D ultrasound," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 2083 – 2089, San Francisco, CA, USA, 2011.

[118] J. W. Cannon et al, "Real-Time Three-Dimensional Ultrasound for Guiding Surgical Tasks," in *Computer Aided Surgery*, vol. 8, no. 2, pp. 82-90, 2003.

[119] S. Yagel et al, "3D and 4D ultrasound in fetal cardiac scanning: a new look at the fetal heart," in *Ultrasound in Obstetrics and Gynecology*, vol. 29, no. 1, pp. 81-95, 2007.

[120] J. Stoll et al., "Passive markers for ultrasound tracking of surgical instruments," in *Medical Image Computing and Computer-Assisted Intervention*, pp. 41-48, Berlin, Heidelberg, 2005.

[121] M. S. Grewal and A. P. Andrews, "Kalman filtering: theory and practice using MATLAB". (Fourth ed.) Ch 1, pp. 1-36, published by *John Wiley & Sons, Inc*. 2015.

[122] S. J. D. Prince, "Computer vision: models, learning and inference", Ch 19, pp. 538-570, published by *Cambridge University Press* 2012.

[123] M. Ficocelli et al., "Adaptive filtering for pose estimation in visual servoing" in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Maui, Hawaii, USA, 2001.

[124] A. Fitzgibbo et al., "Direct least square fitting of ellipses," in *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 21, no. 5, pp. 476-480, 1999.

[125] J. A. Jensen and I. Nikolov, "Fast simulation of ultrasound images," in *IEEE Ultrasonic Symposium*, vol. 2, pp. 1721 – 1724, San Juan, Puerto Rico, 2000.

[126] http://field-ii.dk, "Calculation of B-mode image of cyst phantom", 2012, [Online]. Available: http://field-ii.dk/?examples/cyst_phantom/cyst_phantom.html. [Accessed: 9-Jun-18]
[127] https://www.mathworks.com, "File exchange: image3", 2008, [Online]. Available: https://www.mathworks.com/matlabcentral/fileexchange/21881-image3. [Accessed: 9-Jun-18]

[128] http://petercorke.com, "Machine vision toolbox", 2018, [Online]. Available: http://petercorke.com/wordpress/toolboxes/machine-vision-toolbox. [Accessed: 9-Jun-18]

[129] https://www.mathworks.com, "File exchange: Ellipse Fit (Direct Method)", 2009, [Online]. Available:https://www.mathworks.com/matlabcentral/fileexchange/22684-ellipse-fit--direct-method-. [Accessed: 9-Jun-18]

[130] https://www.mathworks.com, "File exchange: Hausdorff distance", 2013, [Online], Available: https://www.mathworks.com/matlabcentral/fileexchange/26738-hausdorff-distance. [Accessed: 9-Jun-18]

[131] https://www.mathworks.com, "File exchange: RMSE", 2016, [Online], Available: https://www.mathworks.com/matlabcentral/fileexchange/21383-rmse. [Accessed: 9-Jun-18]

[132] H. M. Choset, "Principles of Robot Motion: Theory, Algorithms, and Implementation." MIT Press, 2004. [Accessed via Wikipedia]

[133] http://www.vaultrasound.com, "Transducer Selection", 2017, [online], Available: http://www.vaultrasound.com/educational-resources/scanning-principles/transducers/. [Accessed: 28-Jun-18]

[134] F. Lindseth et al. "Ultrasound-Based Guidance and Therapy", provisional chapter, pp. 2-56, 2013

[135] http://www.ndk.com, "Ultrasonic probe (Transducer)", [online], Available: http://www.ndk.com/en/products/search/ultrasonic/. [Accessed: 28-Jun-18]

[136] N. Pagoulatos, D. R. Haynor and Y. Kim, "A fast calibration method for 3-D tracking of ultrasound images using a spatial localizer," in *Ultrasound in Medicine & Biology*, vol. 27, no. 9, pp. 1219-1229, 2001.

[137] doxygen 1.8.7, "Plus applications user manual", 2018, [online], Available: http://perk-software.cs.queensu.ca/plus/doc/nightly/user/ApplicationfCal.html. [Accessed: 17-Jul-18]

[138] P. J. Besl and N. D. McKay, "A method for registration of 3-D shapes," in *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239-256, 1992.

[139] M. W. Spong, S. Hutchinson and M. Vidyasagar, "Robot Modeling and Control". *John Wiley & Sons. Inc.*, pp. 440-450, 2006.

[140] S.N. Narouze, *"*Atlas of Ultrasound-Guided Procedures in Interventional Pain Managemen.", *Springer Science+Business Media, LLC*. 2011.

[141] A. Goldtein, "Slice Thickness Measuements", J US Med., vol. 7, no. 9, 1988

[142] A. Myronenko et al., "Point Set Registeration: Coherent Point Drift", *IEEE trans. Pattern Analysis and Machine Intelligence*, vol. 32, no. 12, 2010.