Ultrasound-based Needle Tracking and Lateral Manipulation Planning for Common Needle Steering

Wanwen Chen

CMU-RI-TR-21-36

August 18, 2021

The Robotics Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA

Thesis Committee:
Dr. John Galeotti
Dr. Cameron Riviere
Benjamin Freed

Submitted in partial fulfillment of the requirements for the degree of Master of Science in Robotics.

Abstract

Many medical interventions, such as drug delivery and biopsy, require steering a needle towards the desired site. Robotic needle insertion can potentially provide reliable needle access independent of practitioner experience and availability. Significant challenges for achieving automatic needle steering are: (1) tracking a needle using non-invasive imaging techniques to provide image feedback of the needle shape and the target location; (2) modeling the tissue deformation and needle bending to accurately estimate the needed manipulation; (3) planning and replanning the needle steering path to improve the safety of the steering procedure and to adapt to the dynamic environment.

This thesis presents novel methods targeting the above challenges. Two different algorithms are presented to track a partially visible needle using 2D ultrasound. One algorithm focuses on tracking a curved needle, and the other algorithm focuses on tracking a straight needle utilizing both computer vision and robot kinematics. We then present a mechanical model-based steering planning algorithm to robotically insert an ordinary clinical needle to successfully reach an internal moving target in the presence of tissue motion and needle bending.

Acknowledgments

I would like to express my gratitude to my advisor, Dr. John Galeotti. I didn't have much knowledge of medical imaging when I joined the lab, and John gave me the opportunity and guidance in my research. I am grateful for his support in academics and my future career. His passion encourages me to pursue a further career in medical robotics. I would also like to thank my thesis committee members, Dr. Cameron Riviere and Benjamin Freed, for serving on my committee and for their valuable feedback and comments on this research.

My sincere thanks go to Dr. Howie Choset for his inspirations, suggestions, and support in research and my career. I also want to acknowledge the brilliant people in Biomedical Image Guidance Lab and Biorobotics Lab, specifically, Ed, Alex, Gautam, Nico, Evan, Abhi, Blake, Tejas, Raghavv, Yizhu, and all other members who have contributed to this research. Without their help and support, the thesis would not have been possible.

This work was sponsored in part by US Army Medical contracts W81XWH-19-C0083 and W81XWH-19-C0101. I would like to thank our collaborators at the University of Pittsburgh, Triton Microsystems, Inc., Sonivate Medical, and URSUS Medical LLC for their support in this research.

I want to thank my parents for their unconditional love. Thank you for trusting me and supporting me to study abroad. I would also like to thank all my friends. The connections between people become more special and valuable during the pandemics, and I can not imagine life without them.

Contents

When this dissertation is viewed as a PDF, the page header is a link to this Table of Contents.

List of Figures

List of Tables

Chapter 1

Introduction

Needle intervention enables many clinical procedures such as blood drawing, endovascular devices insertion, and biopsy, but the intervention poses risk to critical anatomic structures. The use of robotic support for needle steering and insertion can potentially provide needle intervention independent of practitioner experience and availability. Image-based feedback of the needle's shape and position is required to close the loop on fully autonomous needle insertion. Multiple medical imaging techniques can be used to provide imaging feedback, such as MRI [17, 21], CT [6, 41] and ultrasound [20, 47]. Among these techniques, ultrasound has several advantages. Ultrasound has no ionizing radiation, making it safer to be used on patients. Ultrasound also requires a smaller workspace and lower cost than MRI and CT, and it has a shorter imaging time allowing it to provide real-time internal gray-scale images of patients. Among different kinds of ultrasound imaging, 2D point-of-care ultrasound (POCUS) is used most often because of its lower cost, increased portability, and ubiquity, making it ideal to localize the needle inside the patients in real-time, whether in the emergency room or at the patient's bed side.

However, when the needle is inserted into heterogeneous tissue, the complicated needle-tissue interaction can introduce multiple challenges to automatic needle steering using 2D ultrasound as image feedback.

First, the needle-tissue interaction can bend the needle, and segmenting a curved line inside the ultrasound images is challenging under interference from the tissue layers. Another difficulty is that the bending needle can make the needle deviate

from the image plane. Since the 2D ultrasound only shows a 2D plane of anatomy, a small deviation from the image plane can make the needle become partially (or even not) visible inside the ultrasound image.

Second, the needle-tissue interaction can move the critical structures and the targets. For example, in vascular intervention, the inserted needle can introduce vein deformation [13]. The motion of the targets and critical structures requires that the robot should be able to re-plan the needle steering path to adaptively steer a needle inside a dynamic environment.

This thesis provides multiple methods targeting the problems mentioned above. A novel needle tracking algorithm is presented to track a partially visible curved needle using 2D ultrasound, and a novel information-fusion algorithm is presented to track a robotically inserted needle. We also introduce a new planning algorithm based on both a rapidly-exploring random tree (RRT) [26] and Bubble Bending [53] that can re-plan the needle steering path in dynamic environments where the critical structures and the targets are moving slightly due to the needle-tissue interaction.

The rest of the thesis is organized as follows. Chapter 2 provides a literature survey on 2D ultrasound needle tracking and robotic steering for ordinary clinical needles. Chapter 3 introduces a novel partially-visible-and-curved needle tracking algorithm utilizing a novel weighted RANSAC curve fitting algorithm combined with a probabilistic Hough transform to track the curved needle robustly. Chapter 4 describes a new information fusion-based algorithm to localize a robotically inserted needle by fusing the robot kinematics and ultrasound-based tracking. Chapter 5 presents a mechanical model of needle bending and new replanning algorithms to guide a robot to steer the needle towards a moving vessel.

Chapter 2

Related Works

2.1 Needle Localization in 2D Ultrasound

Ultrasound image guidance can provide visual feedback of the shape of the needle and the locations of obstacles and target(s) to enable autonomous needle insertion into non-rigid internal structures, such as inserting needle towards a vessel which can bend, roll and shift [12, 13]. The ultrasound image feedback can bring multiple improvements to the robot's performance. Chen et al. [14] report that robotic cannulation has a higher first-time success rate and shorter total time to access with ultrasound image guidance than the tests without ultrasound even the tissue conditions are different.

2.1.1 Needle Tip and Straight Needle Localization

There are multiple ways to track a needle inside the 2D ultrasound images. Some works only focus on tracking the needle tip and ignore the needle shaft segmentation, since the primary goal of robotic needle insertion is to let the needle tip reach a target. Kaya et al. [29] propose the algorithms to track the needle tip using optical flow in both lateral and transverse 2D US images. Convolutional Neural Network models can also be used to localize the needle's tip [46].

However, in needle control, the shape of the needle shaft is also important if the control inputs include the insertion angle. Knowing the needle shape can also help us

identify the needle bending to estimate the needed control input to control the needle tip. However, the majority of the research assumes that the needle is a straight line in the US images. Therefore, line detection algorithms such as Hough Transform [18], RANSAC [28], and Maximum Likelihood Estimation Sample Consensus (MLESAC) [45, 59, 69] are often used to detect the needle inside the ultrasound images. Some researchers combine the segmentation algorithms with statistical filters to stabilize the tracking results. For example, Mathiassen et al. [42] first use a search-based method to estimate the needle axis in a region of interest (ROI) and then use statistical filters to track the needle tip. Deep learning-based detection is also a recent trend in needle localization. Mwikirize et al. [44] use a Fast-RCNN network to estimate the needle trajectory roughly and then use phase-based features to refine the needle axis estimation and localize the needle tip.

2.1.2 Curved Needle Localization

Needles can bend due to the interaction with soft tissue, and the needle deflection due to the force interaction can cause inaccuracy in percutaneous therapies [1]. Curved needle tracking can address the tip misplacement caused by needle bending and improve the needle segmentation accuracy.

Previous research scans and localizes the curved needle in different ways. In 2D ultrasound, there are two scanning modes: transverse (out-of-plane) scanning and longitudinal (in-plane) scanning. Transverse scanning scans the needle along the needle axis and detects the needle cross-section in the ultrasound images to reconstruct the needle shape. This method is often combined with robot kinematics to reconstruct the needle shape in 3D. Waine et al. [64] localize the needle cross-section with intensity thresholding and use RANSAC fitting to detect the curved needle shaft. Li et al. [40] use thresholding and a Kalman Filter to track the needle cross-section and use least-square fitting to fit a curved needle axis. Other methods include detecting the comet tail artifact [29] or using blob detection [34] to detect the cross-section. Vrooijink et al. [63] estimate needle's 3D curvature using ultrasound needle cross-section location and robotic kinematics. Zhang et al. [71] use Large Margin Mask-RCNN to detect the cross-section of needles and use DBSCAN clustering to localize multiple needles.

Tracking curved needles in the longitudinal plane has the advantage that longitudinal scanning doesn't require moving the probe to see the entire needle shaft. Therefore, longitudinal scanning is more suitable for tracking an inserted needle in real-time. Extension of line detection algorithms, such as Parametric Hough transform [50] and RANSAC polynomial fitting ([10] and [30]) can be used. Ayvali et al. [7] approximate the curved cannula as multiple line segments and use optical flow and edge detection to track the cannula. However, these methods require that the needle shaft is seen clearly. When the needle is bending, the misalignment between the needle axis and the image plane makes the needle hard to be seen. If the needle is partially visible or the images contain other bright structures, it will be harder to localize the curved needle based on ultrasound images only.

2.1.3 Tracking a Partially or Hardly Visible Needle

Many factors can influence the visibility of needles in ultrasound images, such as the size and placement of the needle and transducer properties [11, 54]. A needle can hardly be seen in the ultrasound images due to bending and misalignment with the image plane. There are relatively few methods to detect a needle when it is partially out of plane and hardly visible. One method is to use other features instead of pixel intensity to detect the needle. Hacihaliloglu et al. [25] use Log-Gabor filter to extract phase-based features and select the region of interest inside the images, then use MLESAC to estimate the needle trajectory of a partially visible needle. Another idea is to detect the micro-motion of tissue introduced by the needle. Humans can roughly estimate the needle location utilizing the temporal information and looking at the moving region in the tissue even they can not see the needle in the images. Czajkowska et al. [15] use a Histogram-of-Gradients approach to detect the tissue motion and estimate needle location. Beigi et al. [8] use Support Vector Machines to locate a non-visible needle based on time-domain features. However, the Histogram-of-Gradients method has a strong assumption that the scanned region does not have large relative movement to the ultrasound transducer, which is hard to achieve when the subject is alive, or the scanning surface is not flat. The learning-based algorithm has not yet shown its generalization ability.

2.1.4 Information Fusion-based Needle Tracking

The ultrasound image-based needle tracking can fall short when the algorithms can not detect the needle due to occlusion or misalignment. Besides the ultrasound-based tracking algorithms, the robot's forward kinematics can also estimate the location of a needle. In robotic needle insertion, robotic kinematics can be used to localize the needle with proper calibration [5, 32, 65]. The issue with relying entirely on robot kinematics is the error introduced during calibration procedures or from needle bending induced by external forces. The fusion of ultrasound-based localization and robot kinematics-based localization has the potential to provide a more stable and accurate needle localization. One way is to use robot kinematics as prior knowledge of the needle location to assist computer vision-based localization algorithms. For example, Chen [12] uses robot kinematics to build a probabilistic map of the needle location, and he reports that computer vision improves needle-tracking accuracy over purely kinematic tracking. Su et al. [57] use the kinematics to build a prior mask on the RGB camera image for surgical instrument detection. Another way is to use information fusion algorithms such as statistical filters to fuse the output of different algorithms. For example, Mathiassen et al. [42] use a Kalman Filter or a Particle Filter to track the needle tip, and they add the robot velocity measurement in Kalman Filter and Particle Filter to increase the robustness and accuracy of needle tip localization.

2.2 Needle Steering Planning

The survey in [1] shows that ordinary clinical needles can bend undesirably due to interaction forces with the surrounding soft tissue, and the bending can influence needle-steering accuracy in percutaneous therapies. The bending is hard to ignore when reaching deep targets. However, purposeful ex-vivo manipulation can utilize in-vivo tissue forces to deliberately bend the needle in vivo to reach around critical structures and re-orient the needle-tip trajectory to reach deep moving targets. Thus, it is important to account for needle bending in robotic needle steering applications.

Much previous robotic needle steering research focuses on steering flexible needle (such as the nitinol needle) and models the needle tip kinematics as bicycle/unicycle

kinematics [67], nonholonomic stochastic model [51], duty-cycle spinning [43]. The control inputs are axial rotation and insertion [55, 66]. The path planning algorithms included variations of Rapidly-Exploring Random Tree (RRT) algorithm [22, 52, 70], motion primitives-based planning [23] optimization-based planning [3, 19, 58]or considering the motion uncertainty [4, 58]. The advantages of such models are that the needle-tissue interaction model is much simpler. The planning and control operation only needs to consider the needle tip position and orientation. However, this model assumes that the needle stays in the tip cutting path and will not cause large tissue displacement. This assumption is valid when the needle is flexible but can be violated when the needle has larger stiffness.

For more rigid needles, most studies use mechanical models to estimate needle deflection. The main challenge is that it is difficult to measure and model the forces. Okamura et al. [49] measure the forces acting on a needle when it is inserted into a liver sample and build a force model that includes capsule stiffness, friction force and tip cutting force. Khadem et al. [31] divide the interaction into three phases (puncturing, cutting, insertion and retraction) and test their model in inhomogeneous tissue. With force models, many studies can model the needle as a beam reacting to external forces. There exist several models for needle-tissue interaction forces. The virtual spring model considers the tissue as a series of springs supporting the needle. Neubach and Shoham [47] and Zhao et al. [72] both use virtual springs to model tissue support forces in inhomogeneous tissue. Finite element models are also common approaches. Adagolodjo et al. [2] use an inverse Finite Element (FE) simulation to control a robot steering a needle into deformable structures. Dimaio and Salcudean [16] use finite element analysis to analyze the tissue deformation and needle bending.

Some research on mechanical model-based planning is about preoperative planning: finding the best insertion location (Tsumura et al. [62], Zhao et al. [72]). However, preoperative planning is sensitive to model uncertainty and the motion of internal structures and cannot adjust the manipulation based on external feedback. DiMaio and Salcudean [16] use an artificial potential field to plan the needle steering path. While this method can adjust to the dynamic scenes, potential field-based planning can be stuck at a local minimum. Lehmann et al. [39] use an optimization-based method to find the best lateral force profile and adjust the tip trajectory using inverse

kinematics. They do not consider complicated obstacles in the insertion scene, and their goal is to minimize the needle deflection. Rapidly-exploring Random Tree (RRT) is a method that can be easily implemented and widely used in robotics planning. Xu et al. [70] use RRT to plan the needle insertion path in a 3D environment with obstacles. Huang and Lei [26] propose the potential field-guided RRT with iteration learning to plan the needle control sequence. However, their methods can not adjust to dynamic environments. In needle steering, the environment often does not have rapid and severe changes. Elastic Band algorithms can provide real-time path deformation to avoid critical structures and adjust the target location. For example, Pinzi et al. [53] use Bubble Bending, which is a variant of elastic band algorithm, to replan the needle steering path smoothly.

Chapter 3

Ultrasound-based Tracking of Partially In-plane, Curved Needles

Chapter 3 is adapted from the publication: **Wanwen Chen**, Kathan Nilesh Mehta, Bhumi Dinesh Bhanushali, and John Galeotti. "Ultrasound-Based Tracking Of Partially In-Plane, Curved Needles." In *2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI)*, pp. 939-943. IEEE, 2021.

Wanwen Chen's contributions as the first author of the publication include: developing the algorithm, conducting the experiments, analyzing the data, writing the manuscripts, and responding to reviewers with revisions. Kathan Nilesh Mehta and Bhumi Dinesh Bhanushali are responsible for implementing a U-Net needle tracking algorithm as a baseline comparison in this paper, and John Galeotti is the supervising faculty advisor.

This chapter introduces a novel algorithm to track a curved needle in 2D ultrasound images. The algorithm includes a novel weighted RANSAC polynomial fitting approach, which uses the agreement between the kinematics and the segmentation to penalize inaccurate RANSAC fitting for increased accuracy with less influence from similar tissue pixels. It can estimate the kinematic information, including the needle's entry point in the image, inserting angle and length, or utilizing kinematic information from an actual needle-inserting robot.

3.1 Methods

Algorithm Overview

The needle must first be partially inserted so that its tip is visible in ultrasound. The algorithm is then initialized with either a user-provided needle location, or an external needle pose obtained from robot kinematics or electromagnetic tracking. The algorithm first uses probabilistic Hough transform to detect a straight line. Once the length of the detected needle exceeds a threshold, the algorithm uses our novel weighted RANSAC approach to fit a polynomial curve. The detected needle is fed to a Kalman filter to obtain stabilized locations of the needle tip and the insertion site. Kinematic information is used to evaluate the detection and to reject outliers. The process is described in the flowchart in Figure 3.1.

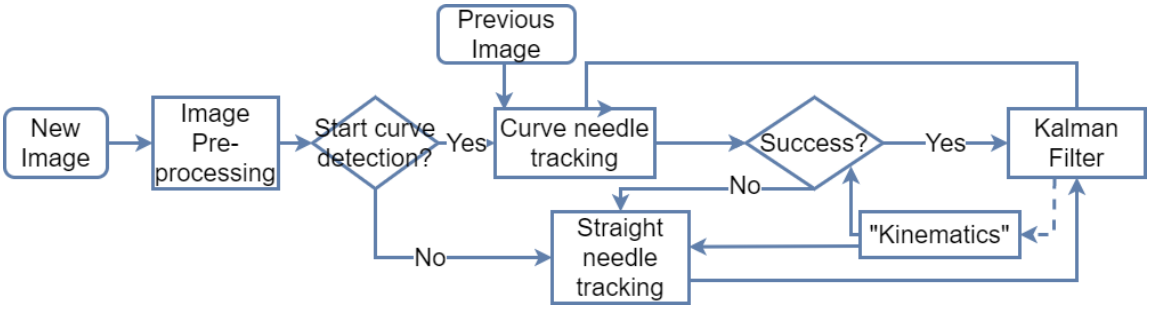

Figure 3.1: The flowchart of the curved needle tracking algorithm. The dashed line means that the connection is optional.

Image Pre-processing

All the ultrasound images are filtered with a median filter to reduce speckle noise. The images are cropped into a rectangular region of interest (ROI) that is 20 pixels wider and deeper than the bounding box from needle tip to insertion site based on the kinematics. The number of pixels is tuned by the insert speed and the image resolution.

Kinematics Estimation

The needle kinematics estimation includes three kinematic variables to represent the pose of the needle: insertion location, insertion length, and insertion angle. The variables are used to reject outliers, which will be described later. The variables can be calculated from the state of the Kalman filter when the external estimation is not

available or can be estimated from external information such as robotic kinematics.

Curved Needle Tracking with Weighted RANSAC

The motivations of our weighted RANSAC are to assign higher weights to the candidate pixels that have a higher probability of being the needle and penalize the curve that deviates from the kinematics estimation.

The curved needle tracking algorithm will fit a third-order polynomial using our novel weighted RANSAC in Algorithm 1. We first use thresholding to select pixels with high intensity and significant response to a Sobel filter in order to detect the edge of the needle. Then the algorithm discards the pixels that are too far away from a straight-line approximation of the prior frame's needle to reject outliers. The distance threshold is necessarily increased in proportion to needle curvature. In our experiments we set the threshold as 25 pixels for needles with a large curve and 10 pixels for others.

Our weighted RANSAC selects the curve with the highest weight as shown in Algorithm 1. The weight is evaluating the possibility of the pixel belonging to the needle. We consider four aspects: (1) the needle usually has higher intensity, (2) the needle usually appears as a line in the images, so the pixels have a larger response to the Sobel filter, (3) the needle might have large intensity changes since it can appear or disappear in the images, (4) we want to add more weights to the pixels that belong to the needle tip since our needle tip has a larger echo in the images. In Algorithm 1, w_p is the weight of the candidate pixel, w_I is the min-max normalized intensity in the ROI, w_G is the min-max normalized response of Sobel filtering, w_D is the min-max normalized absolute difference with the last image, and w_T is the distance between the estimated tip location from the set of pixels: $w_T = exp(-(x_T - x)/c)$, where x is the current x coordinate of the pixel, $x_T = argmax_x|x - x_K|$ (x_K is the x coordinate of the entry point in the Kalman filter), and c is a constant. To penalize deviation from the insertion location on the side of the image, $w_K = 1/(\log|y_C - y_K|)$. y_K is the y coordinate of the entry point in the Kalman filter, and y_C is the estimated y coordinate entry point on the curve, which is the value of x_K of the estimated curve. Curve fitting can fail for many reasons such as if the curved needle is partially out of plane and doesn't have good visibility. Therefore, we use three thresholds to evaluate the result: the distance between the detected tip and the reference tip, the distance between the detected and reference entry point, and the angle of the line

that connects the tip and entry point. If any of these values exceeds its threshold, we will switch to our more robust but potentially less accurate straight-line needle approximation for that frame.

Algorithm 1 Pseudocode for Weighted RANSAC

procedure WEIGHTEDRANSAC(I_{ROI})
  P = Threshold(I_{ROI}) $\cap$ Threshold(Sobel(I_{ROI}))
  P = Reject_Outliers(P)
  for p in P **do**
    $w_p = w_i \times w_G \times w_T \times w_D$
  $max_w = 0$, $model_{best} = None$
  while $i \leq It_{max}$ **do**
    P_{sample} = Random_Sampling(P)
    $model$ = Polynomial_Fitting(P_{sample})
    $w_{curve} = \sum_{p \in P_{close}} w_p \times w_K$
    if $w_{curve} \geq max_w$ **then**
      $max_w = w_{curve}$, $model_{best} = model$
    $i = i + 1$
  Return $model_{best}$

Straight Needle Tracking

We use probabilistic Hough transform [24] to detect a straight needle because it has a smaller tip localization error comparing with straight line RANSAC and Hough transform in [12]. The candidate pixels are provided by Otsu thresholding in the ROI region. Of the resulting lines, we select the longest line meeting two maximum-difference threshold tests: (1) the difference between the line's angle and the kinematic angle estimate, and (2) the difference between the line's intersection with its apparent insertion location and the kinematic estimate of the insertion location. Since the needle might be partially visible, we adjust the length of the detected needle: we extend the needle until it reaches the edge of the image at one end and its length becomes equal to the length in the kinematics, and then search along the deeper end of the needle. As in [47], the algorithm subtracts the previous image frame from the current image frame, and the tip is the pixel having the largest intensity change.

Kalman Filter

An Extended Kalman filter is used in the final stage of the process to stabilize

the tracking. The state variables of the Kalman filter are the coordinate and speed of the tip location, and the coordinate of the insertion location. The observation variables are the detected tip and insertion location. If the distance between the prediction from Kalman filter and the output of the tracking algorithm is smaller than a threshold, the inserted location and angle in the kinematics estimation will be updated to the prediction from the Kalman filter.

3.2 Experiments

We contain two part of experiments to collect both phantom and in-vivo images.

Phantom Experiments

The experiment setup is in Figure 3.2. The data was collected using a PaoLus UF-760AG Portable Diagnostic Ultrasound Imaging Equipment (FUKUDA DENSHI, UK) and a linear transducer with 51mm scanning width. The transducer was set to 12MHz with an imaging depth of 4cm. The original frame rate was 94 fps but was downsampled to 15.67 fps in testing. The gain value was 21dB. The dimension of each image is 512×532 pixels, and the pixel pitch is 0.10mm horizontally and 0.075mm vertically. The images were cropped to 430×496. The ultrasound probe was held by hand, and all images were acquired from an anthropomorphic tissue phantom with simulated blood vessels (Advanced Medical Technologies, WA). Four 2.5-inch echogenic needles were used, and three of which were pre-bent by differing amounts. Each needle was inserted for 6 trials with an insertion angle between 30° to 60°. Each insertion sequence had 50-80 frames. There were 384 test images for straight needles, 364 test images for small curvature, 408 test images for medium curvature and 373 test images for large curvature in total. We asked two graders to contour the boundary of the needle shaft in the images. The needle axis is a third order polynomial to approximate the average of the upper and the lower edges. The needle tip is the end of the top edge.

In-vivo Porcine Experiments

We did in-vivo robotic needle insertion experiments into the legs of anaesthetised pigs, and the illustration of the robotic mechanism is in Figure 3.3. The needles were inserted into the muscle or toward the femoral artery. The data was collected using the same ultrasound machine and transducer, with an imaging depth of 5cm. The

frame rate is 30 fps and were downsampled to 10 fps in testing. The probe is fixed to the end-effector of a UR5 robot arm (Universal Robots, Denmark), next to the needle insertion mechanism. Before the insertion, we collected the ultrasound images of the robot inserting the needle into a water tank, and we had a linear regression on the x, y coordinate of the tip and the linear actuator feedback. The relation between the linear actuator feedback and tip location then provided our kinematic estimates. The images were captured by ROS for synchronization. The dimension of the images is 503×510 pixels, with pixel pitch being 0.10mm horizonally and vertically. One grader was asked to label every third frame and in total, we have three valid insertion trials, 527 test images, with 177 labeled images for evaluation.

We compare the result of probabilistic Hough transform, probabilistic Hough transform with classical RANSAC ("classic RANSAC"), and probabilistic Hough transform with our weighted RANSAC ("weighted RANSAC"), and a trained U-Net [56]. We compare the metrics below: (1) error of the tip localization, which is defined as the Euclidean distance between the detected and labeled tips, (2) tip angle error, which is the absolute difference between the detected and labeled tip angles (the tangent angle of the polynomials at the tip), (3) shaft fitting error, which is the mean Euclidean distance between the pixels on the polynomial of the labeled needle to the pixels on the detected needle which has the same x coordinate. When calculating these metrics from U-Net, we used morphology before fitting a polynomial in phantom images but without it in pig because it degrades the performance.

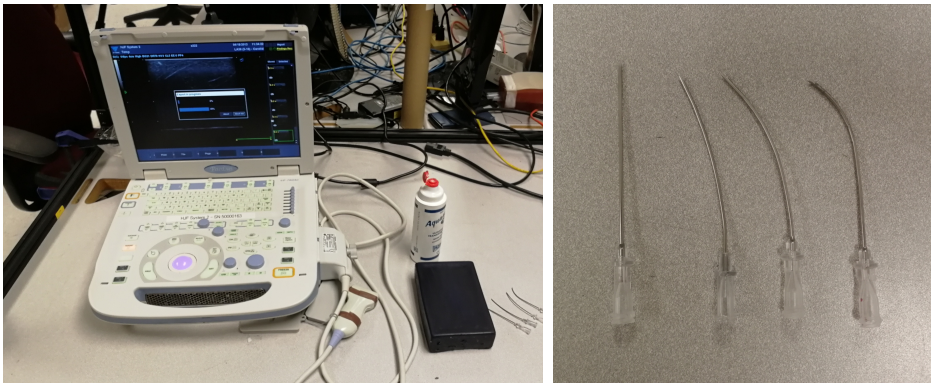

Figure 3.2: The image on the left is the phantom experiments set up, and the image on the right are the curved needles used in phantom experiments.

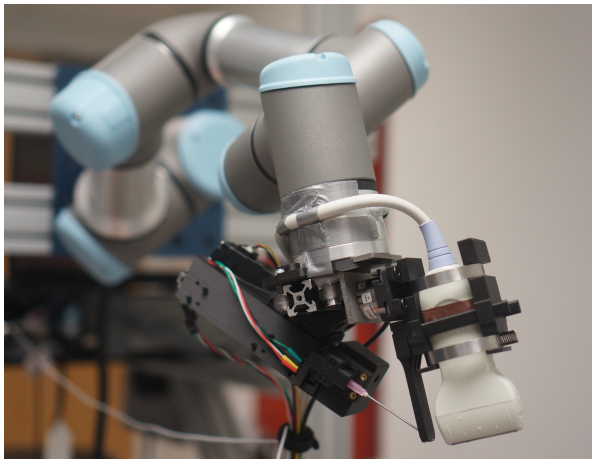

Figure 3.3: The robotic mechanism in the in-vivo experiments.

Needle	Straight	Small	Medium	Large
Tip	1.77 ± 1.29	2.30 ± 1.95	2.50 ± 2.53	2.61 ± 2.05
Shaft	0.60 ± 0.62	0.89 ± 1.09	0.56 ± 0.63	0.59 ± 0.59

Table 3.1: The maximum error of tip localization and shaft fitting compared with two graders for our weighted RANSAC. All errors are in mm.

Method	Tip error/mm			
	Test 1	Test 2	Test 3	Avg.
Method 1	3.30 ± 1.36	2.78 ± 1.85	1.81 ± 1.14	2.26 ± 1.46
Method 2	3.14 ± 1.51	2.78 ± 1.79	1.69 ± 1.06	2.16 ± 1.43
Method 3	3.08 ± 1.28	2.68 ± 1.77	1.47 ± 0.90	$\mathbf{1.98 \pm 1.35}$
Method	Shaft fitting error/mm			
	Test 1	Test 2	Test 3	Avg.
Method 1	0.69 ± 0.29	0.57 ± 0.56	0.90 ± 0.89	$\mathbf{0.81 \pm 0.77}$
Method 2	1.00 ± 0.63	0.97 ± 0.91	1.16 ± 1.11	1.10 ± 1.00
Method 3	0.95 ± 0.52	0.77 ± 0.81	0.97 ± 0.86	0.94 ± 0.80

Table 3.2: Error of tip localization and shaft fitting in in-vivo experiment of inserting a straight needle that slightly bent in the tissue. Method 1 is probabilistic Hough transform, method 2 is classical RANSAC and method 3 is our weighted RANSAC.

3.3 Results and Discussion

Phantom experiments: Our weighted RANSAC algorithm is implemented in Python and C++, running at 11 fps (0.093 sec/image). Figure 3.4 shows that our weighted RANSAC algorithm has the smallest error in tip localization, with the quantitative results in Table 3.1. Figure 3.5 is an example of the results of the tracking algorithms. The error of probabilistic Hough transform increases with the curvature because it can fit only partial of the needle. Compared with our weighted RANSAC, the classical RANSAC is less robust, with a larger shaft fitting error for some needle curvatures and larger error in tip localization, showing that our weighted RANSAC improves the performance of classical RANSAC. The reasons might be that (1) our designed weights allow our weighted RANSAC to focus on more pixels instead of only the bright structures that look similar to the needle (2) our weights add more weights to the tip, so it has better ability to keep track to the needle tip (3) our penalized kinematics also make sure that the selected curve does not deviate from the kinematics too much.

To evaluate the performance of tip angle estimation, we selected the frames that both our weighted RANSAC and classical RANSAC have a tip error smaller than 2mm and ran a two-sample two-sided t-test. The mean error of the tip angle in these frames is 8.98° for weighted RANSAC, 9.67° for classical RANSAC, and 7.10° between the graders. The null hypothesis is that the tip angle error of classical RANSAC and weighted RANSAC has no difference. The p value is 0.049, showing that our method has a significantly smaller error in tip angle estimation. The reason of the large error in the angle fitting for all algorithms and in-grader comparison might be the needle tip is hard to perceive sometimes. The U-Net model has a Dice loss of 0.34. The tip error is 4.97 ± 5.99mm for straight needle, 3.14 ± 3.43mm for small curvature, 1.99 ± 4.17mm for medium curvature and 3.54 ± 3.59mm for large curvature. The results of the U-Net model show the difficulty to segment the needle and clean the false positive to detect the tip because the shaft is often partially visible, and similar structures confuse the classifier.

In vivo experiments. Table 3.2 shows the in-vivo error of tip localization and shaft fitting, and results are shown in Figure 3.5. The U-Net model has a Dice loss of 0.27. The test set is Test 1 and 2 in Table 3.2: the mean tip error is 2.70 ± 3.09mm,

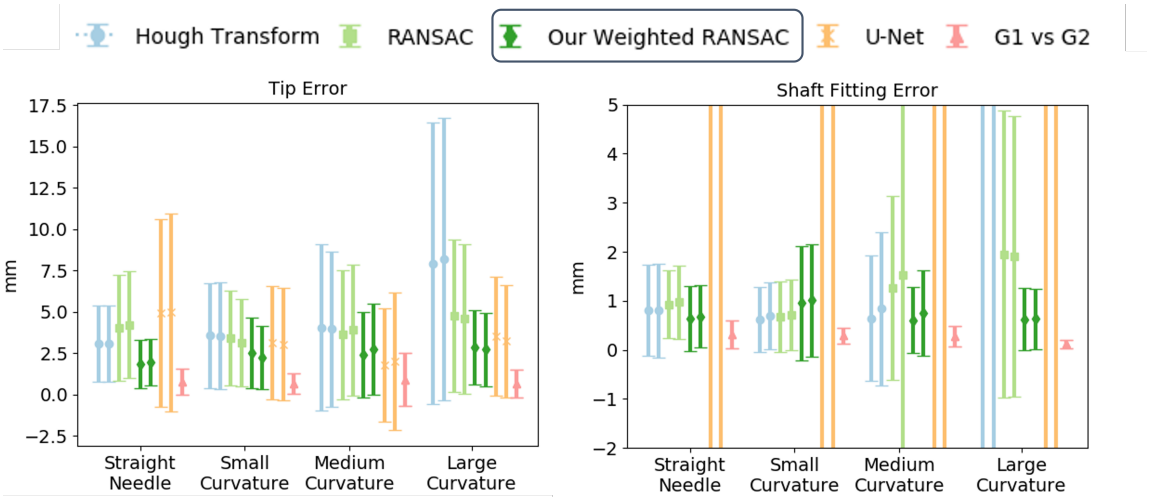

Figure 3.4: The accuracy of the tracking algorithms for different needle curvatures vs two graders and between two graders (G1 vs G2). The point is the mean of the error, and the line is the error bar.

and the shaft error is 2.71 ± 4.68mm. The U-Net has a better tip localization in in-vivo experiments, but behaves poorly in shaft fitting. Our weighted RANSAC performs better in localizing the needle tip than other classical methods. It has worse performance in shaft fitting than probabilistic Hough transform, because the natural curvature was small in a shallow stick and also because the complex, bright tissue layers had undue influence on RANSAC. However, the needle tip is bright in these test cases, and our weighted RANSAC tracks the needle tip well. We suspect that future in-vivo studies of deeper needle punctures will include larger naturally occurring needle bending, for which our method might have a greater performance improvement over other methods.

A shortcoming of our algorithm is that it can not be initialized with computer vision alone but rather needs human initialization or external measurements. However, this is not a concern for robotic needle insertion, which is our primary application and the subject of our experiments. Another problem is that it can't consistently discriminate a bending needle from other bright, gently curing nearby structures, such as transverse vessel walls or tissue layers. Even on such a challenging problem, our algorithm has high accuracy on tip localization, which is more important in needle tracking.

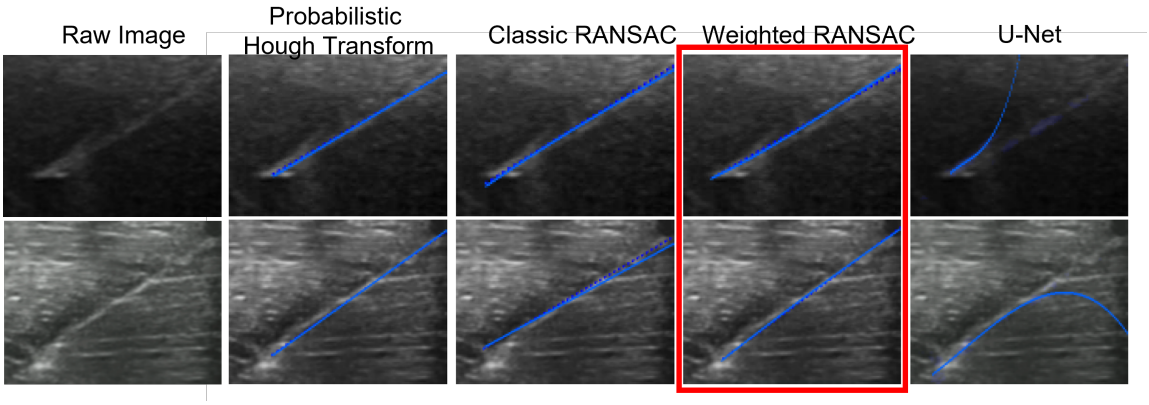

Figure 3.5: Results in phantom experiments (upper row) and in vivo experiments (lower row). The images from left to right are the raw images and the results of probabilistic Hough transform, classic RANSAC, weighted RANSAC and U-Net. The solid line is the result of segmentation. The dashed line connects the tip and the entry point from the Kalman filter.

3.4 Conclusion

This chapter presents a novel curved-needle tracking algorithm in 2D ultrasound. This approach was validated with different needle curvatures in phantom and in-vivo robotic needle insertion experiments. It achieved higher overall accuracy than the other algorithms we tested, especially for needle-tip localization. Future work may include automatic tracking initialization or multi-algorithm fusion with machine learning.

Chapter 4

On-line Information Fusion-based Needle Tracking in Ultrasound for Robotic Needle Insertion

Chapter 4 is adapted from the manuscript: **Wanwen Chen**, Evan Harber, Nicolas Zevallos, Kirtan Patel, Yizhu Gu, Kenneth Sladick, Raghavv Goel, Howie Choset, and John Galeotti. "Real-time Needle Tracking in Ultrasound for Robotic Needle Insertion."

This chapter presents the original work of Wanwen Chen, whose specific contributions as first author of the publications include: developing the needle tracking algorithm, analyzing the data, writing the manuscripts and responding to reviewers with revisions. Evan Harber and Nicolas Zevallos are responsible for the needle control algorithms and experiments. Evan Harber and Raghavv Goel are responsible for the needle kinematics calibration. Kirtan Patel, Yizhu Gu and Kenneth Sladick are responsible for designing the needle insertion mechanism. Howie Choset and John Galeotti are the supervising faculty advisors.

This chapter introduces a new data fusion algorithm that combines robot kinematics and computer vision to track the needle robustly with the time-delay and online asynchronous tracking from ultrasound images. The contributions of this work include:

1. an ultrasound image-based needle tracking algorithm to estimate the location

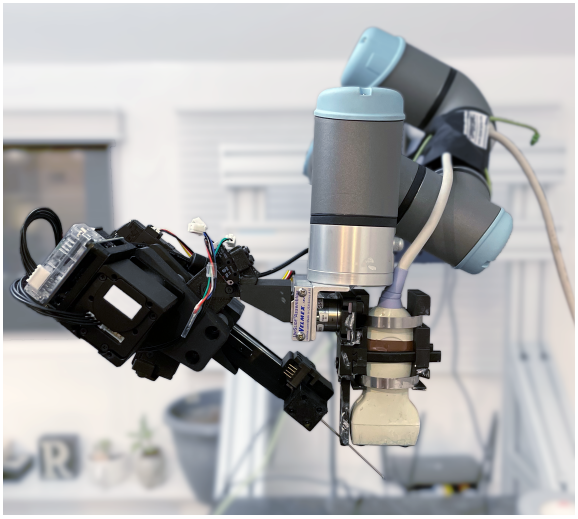

Figure 4.1: The two degrees of freedom (DOFs) needle insertion mechanism mounted to a UR3e arm along side an ultrasound probe.

of a poorly visible needle;

2. an adaptive Kalman filter-based data fusion method to localize the needle fusing the measurements from ultrasound-based tracking and robotic kinematics, and the method to use the ultrasound-based tracking to correct the kinematics estimation on-the-fly;
3. an approach for fusing asynchronous and time-delayed data in from visual tracking and robot kinematics, making them more flexible for real-time applications.

4.1 Methods

In this section we introduce the three individual needle tracking methods that comprise our novel needle tracking algorithm, and the data fusion method.

4.1.1 Robot Kinematics

Insertion Mechanism

We used a system designed for the development of automatic femoral access as seen in Fig. 4.1. Specifically we focused on the last two degrees of freedom of the system

(Fig. 4.2), the angular and the linear degree of freedom (DOF). Together they can be used to control the angle and depth of the needle tip in the 2D ultrasound image.

Based on our mechanical constraints, we derive the following functions to map angular and linear actuator feedback to needle tip position:

$$x_{pixel} = (p_l l + l_{off}) \cos(p_\theta \theta + \theta_{off}) x_{scale} - x_{off} \tag{4.1}$$

$$y_{pixel} = (p_l l + l_{off}) \sin(p_\theta \theta + \theta_{off}) y_{scale} - y_{off} \tag{4.2}$$

Where l and θ are the feedback from the angular and linear DOFs, $p_l, l_{off}, p_\theta, \theta_{off}$ are parameters in linear functions to convert the actuator feedback to insert length and angle. x_{off} and y_{off} are the location of the center of rotation in pixel space and x_{scale} and y_{scale} are ultrasound scaling factors due to distortions of the ultrasound image. In total this gives 8 parameters necessary to estimate the forward and inverse kinematics of the needle in the ultrasound image frame.

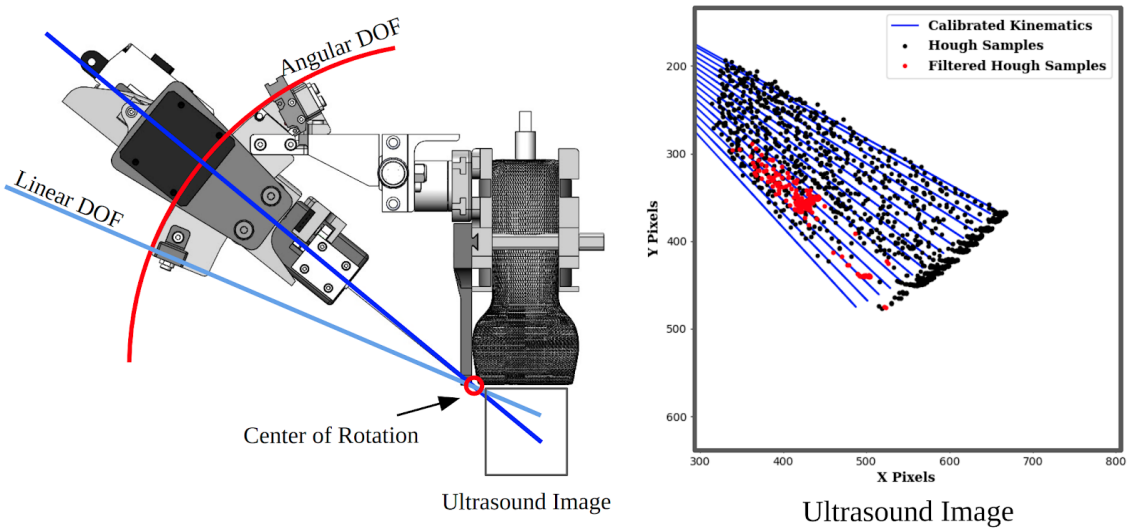

Figure 4.2: The diagram on the left shows the simplified mechanism design with its two DOFs used to measure the position of the needle in the ultrasound frame. The right half shows our calibration result converting the feedback from the actuators to the ultrasound image plane. The black dots are the detected tip locations from the Hough transform, and the red dots are the selected tip locations filtered out due to noisy detection. The blue lines are the calibrated kinematics.

Kinematic Calibration

The calibration routine is to fit the 8 parameters described in Eq. 4.1 and Eq. 4.2. We place the ultrasound tip in a water bath to provide clear images and free movement of the insertion mechanism without any outside forces. Next, we discretely sweep linear and angular actuator positions, covering the entire workspace of the mechanism. In order to estimate the tip of the needle we use the Otsu thresholding and Hough transform described in Sec. 4.1.2, to automatically estimate x_h and y_h as our ground truth needle tip locations. From here the 8 kinematic parameters can be optimized to reduce reprojection error of the needle tip (x_{pixel}, y_{pixel}) in Eq. 4.3.

$$\sum_{(x^i,y^i)} [(x^i_{pixel} - x^i_h)^2 + (y^i_{pixel} - y^i_h)^2] \tag{4.3}$$

4.1.2 Probabilistic Hough Transform

The Probabilistic Hough transform [33] is a line detector which is a variant of the Hough transform. A Hough transform uses the line equation $r = x\cos\theta + y\sin\theta$ to associate line parameters (r, θ) with point (x, y): given a single point (x, y) in the image plane, the set of all possible lines going through this point corresponds to a curve in (r, θ) plane. Hough transform finds the intersection points in the (r, θ) plane, meaning that the straight line with the parameters passes considerable points in the (x, y) plane. The Probabilistic Hough transform randomly selects a subset of the points as the input for the Hough transform to improve the efficiency with little impairment on the performance.

We use probabilistic Hough transform because it has higher needle tracking accuracy and speed than Hough transform and RANSAC in [12]. Before the detection, the images are cropped into a rectangular region of interest (ROI) 20 pixels wider and deeper than the bounding box from needle tip to insertion site based on the kinematics. Then the candidate pixels are selected by Otsu threshold. Of the resulting lines, we select the longest line meeting two maximum-difference threshold tests: (1) the absolute difference between the line's angle and the kinematic angle estimation, and (2) the Euclidean distance between the line's insertion location and the kinematic estimate of the insertion location in the image coordinate system. In our dataset, the

thresholds are set to be 8° in angle difference and 30 pixels in the distance between the insertion location.

4.1.3 Optical Flow-based Tissue Motion Segmentation

The idea of this algorithm is to detect the motion introduced by the needle insertion in the images. When the needle is inserted, the needle motion often introduces tissue motion locally, and experienced clinicians can tell the needle location based on the tissue motion even if the needle is not seen in the ultrasound. Live tissue usually has visible structures and small structures that produce speckle noise, both of which allow block matching [35, 47] or optical flow [27, 60] to use these features to track the pixel-level motion in ultrasound images. We specifically use Brox Optical flow [9] to estimate the pixel-level motion, then estimate and subtract a global affine motion from the motion vectors to obtain the dense optical flow. The graph-based min-cut segmentation in [68] is used to segment the dense optical flow. The segmentation algorithm assigns every pixel x with a binary label $L(x)$ into "static" and "moving" regions. Suppose $L(x) = 0$ is "static" and $L(x) = 1$ is "moving". The algorithm finds $L(x)$ which minimizes the energy in Eq. 4.4, and in our dataset we set $\lambda = 1.5$. The static likelihood ξ_{static} is set to 1, while the motion likelihood ξ_{motion} is calculated using Eq. 4.5, where $\vec{v}_{of}$ is the flow vector of the pixel, $\vec{v}_{robot}$ is the unit vector of the insertion direction estimated by the forward kinematics multiplied by a tuning scalar k. We set $k = 2$ after tuning. The likelihood of the pixel which is recognized as "moving" in the previous frame will be multiplied by a constant scalar α (we set $\alpha = 3$ in our dataset) to improve the consistency of the segmentation in the video. To ensure the continuity of the region, we also include a neighbor cost $\xi_n = ||\vec{v}_{\hat{x}} - \vec{v}_x||_2$, where $\hat{x}$ is the pixel of 4-neighborhood of pixel x.

$$E = \sum_x -L(x)\xi_{motion} - (1 - L(x))\xi_{static} - \lambda \sum_{\hat{x} \in \mathcal{N}_4(x)} \xi_n \tag{4.4}$$

$$\xi_{motion} = ||\vec{v}_{of} \cdot \vec{v}_{robot}||k \tag{4.5}$$

After the moving region is segmented, the direction of the needle is estimated by the first component of principle component analysis relative to the center of the region

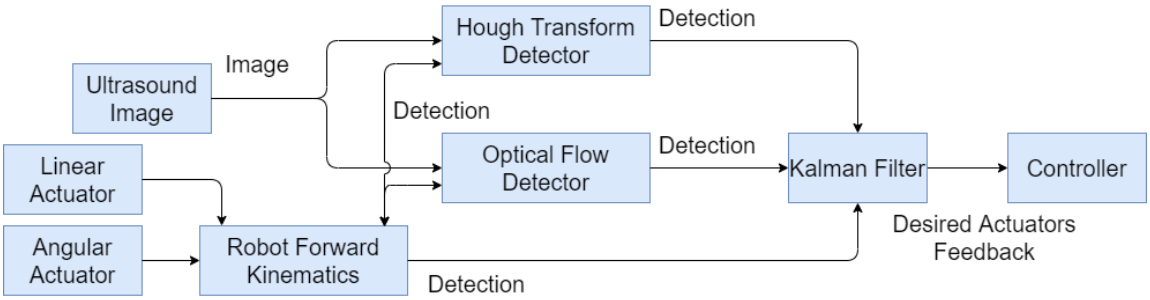

Figure 4.3: The block diagram of the needle tracking algorithms. The arrows represent the communication between the different nodes.

and the optical flow direction. We select the direction that has the smallest difference from the estimation from the robot kinematics. We then draw a line from the center of the moving region along the selected direction until it reaches the region's boundary. The needle axis is the line that passes the region center, and the needle tip is the intersection of the needle axis and the region's boundary.

4.1.4 Data Fusion

The above three tracking methods are implemented in python using the Robot Operating system (ROS) to run independently in three different nodes whose detection results (the tip location and the insertion location) are published to the other nodes (Fig. 4.3). Our proposed information fusion has three aspects:

1. Kinematics estimation is used as a reference in the image-based needle detection as described in 4.1.2 and 4.1.3.

2. Kinematics calibration update: The kinematics calibration is often inaccurate because (1) When the needle punctures through the skin and tissue layer, the insertion angle introduces deflections [61]; (2) Ultrasound artifacts encountered during calibration interfere the Hough transform detection, so the result of the ultrasound-based detection is often noisy.

We use ultrasound-based tracking to correct the kinematics estimation. When the outputs of the two ultrasound-based tracking algorithms pass a maximum-difference test of the tip location and angle, we trust the estimation from the ultrasound images and update the kinematics parameters based on the visual detection. The measurement from probabilistic Hough transform is used to optimize parameters,

because probabilistic Hough transform is more accurate than the tissue motion-based detector. We use the same optimization procedure in the kinematics calibration to optimize the new x_{off}, y_{off}, l_{off} and θ_{off} to minimize the cost function Eq. 4.6, where (x_{pixels}, y_{pixels}) and θ_{pixels} is the tip and insert angle estimated from robotic forward kinematics, (x_h, y_h) and θ_h is the tip and insert angle detected from the probabilistic Hough transform. We found that these 4 parameteres were enough to 'fix' the error in the kinematics previously mentioned.

$$\sum_i (x^i_{pixels} - x^i_h)^2 + (y^i_{pixels} - y^i_h)^2 + (\theta^i_{pixels} - \theta^i_h)^2 + \Delta p \tag{4.6}$$

Where Δp is the change in parameters from the calibrated values.

3. Kalman filter: An adaptive Extended Kalman filter (EKF) which is able to take the measurements with time-delay and asynchronous measurement modified from [48] is used to fuse the estimation from kinematics and ultrasound-based detection. The states of the EKF are the insert location, angle, insert length and insert speed in the ultrasound image coordinate system, as in Eq. 4.7. The kinematics model and the observation model are in Eq.4.8, 4.9 and 4.10. The detection from probabilistic Hough transform, optical flow-based motion detector and robot kinematics are the measurements to update the Kalman filter. Inspired by [37], we adapt the intersection over union (IoU) of the bounding box between the detection from kinematics estimation and the ultrasound-based estimation to give a confidence score to the detection. The confidence score is used as a scalar to adjust the covariance of the measurement. When the detection is of lower confidence, the covariance matrix will divide the confidence score, making this detector has less effect in the Kalman Filter update. Because the Kalman filter only needs to do small matrix operations, it can keep publishing its estimation of the needle localization at 100Hz, having the potential to return tracking estimation in real-time.

$$X = [p_x, p_y, \theta, l, v]^T \tag{4.7}$$

$$\begin{aligned} X_k &= F X_{k-1} \\ y_k &= H X_k \end{aligned} \tag{4.8}$$

$$F = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & dt \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.9}$$

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{4.10}$$

4.2 Needle Tracking Experiments

All the images are collected using a PaoLus UF-760AG Portable Diagnostic Ultrasound Imaging Equipment (FUKUDA DENSHI, UK) and a linear transducer (51mm scanning width). The transducer was set to 12MHz with an imaging depth of 5cm. The gain value was 21dB. The images were captured by ROS for synchronization with 30 fps. The dimension of the images is 503×510 pixels, with pixel pitch being 0.10mm horizontally and vertically. Furthermore, the angular DOF was controller by H42P-020-S300-R Dynamixel. The linear DOF was controlled by PA-12-22017512R linear actuator with an 18 Gauge GWI1802 guide wire introducer needle mounted at the tip.

The proposed tracking algorithm was evaluated in in-vivo robotic needle insertion experiments. The needles were inserted into the muscle or toward the femoral artery of anesthetized pigs. During the insertion, the angular degree of freedom is set to be a constant 45° based on clinicians' recommendation. Afterwards, two graders were asked to label the videos using the Computer Vision Annotation Tool using a straight line to label the tip location and the insertion location. The dataset includes three insertion attempts into the artery and three attempts of insertion into muscle. In total, there are 2037 images for muscle insertion and 5147 images of artery insertion attempts.

The evaluation metrics for the needle tracking algorithms are: (1) the mean Euclidean distance of the tip localization, (2) the mean absolute error between the

angle, (3) the dropout rate, which is the ratio of the number of images for which the needle cannot be detected to the total number of test images in the trial. For the individual modules, we test them offline by time synchronizing the data source and running the individual needle detection algorithms. We compare the three different tracking modules in our algorithms with the fusion result in Table 4.1. The result shows that our proposed fusion algorithm improves the accuracy of tip localization, which is the most important metrics in needle tracking, using multiple tracking methods with lower accuracy.

Table 4.1: Quantitative evaluation of the individual modules and the final results from our needle tracking algorithm.

Algorithm		HoG		Optical Flow		Hough Transform		Robot Kinematics		Kalman Filter		In-
Grader		1	2	1	2	1	2	1	2	1	2	Graders
Tip Error/mm	muscle	27.5 ± 8.9	28.0 ± 8.9	6.3 ± 4.9	6.6 ± 5.1	4.6 ± 4.1	4.7 ± 3.9	5.0 ± 2.1	5.2 ± 2.2	$\mathbf{3.6 \pm 3.0}$	$\mathbf{3.8 \pm 3.1}$	1.2 ± 0.8
	artery	17.5 ± 7.9	17.6 ± 7.9	6.1 ± 4.8	5.9 ± 4.9	$\mathbf{3.0 \pm 2.6}$	$\mathbf{3.0 \pm 2.6}$	5.2 ± 1.2	5.3 ± 1.3	4.0 ± 2.2	4.0 ± 2.3	0.5 ± 0.4
Angle Error/°	muscle	6.4 ± 8.4	6.5 ± 8.6	18.6 ± 13.4	18.1 ± 13.8	3.7 ± 2.8	3.7 ± 2.8	$\mathbf{3.5 \pm 2.0}$	$\mathbf{3.3 \pm 2.9}$	4.5 ± 4.4	4.4 ± 5.3	1.0 ± 1.0
	artery	14.1 ± 15.5	13.6 ± 19.0	21.4 ± 12.9	21.9 ± 19.3	3.7 ± 2.8	$\mathbf{5.5 \pm 14.6}$	$\mathbf{3.0 \pm 2.4}$	6.4 ± 13.7	4.8 ± 5.1	7.1 ± 13.8	3.8 ± 13.6
Dropout Rate	muscle	22%		30%		75%		**0%**		**0%**		/
	artery	83%		69%		83%		**0%**		**0%**		/

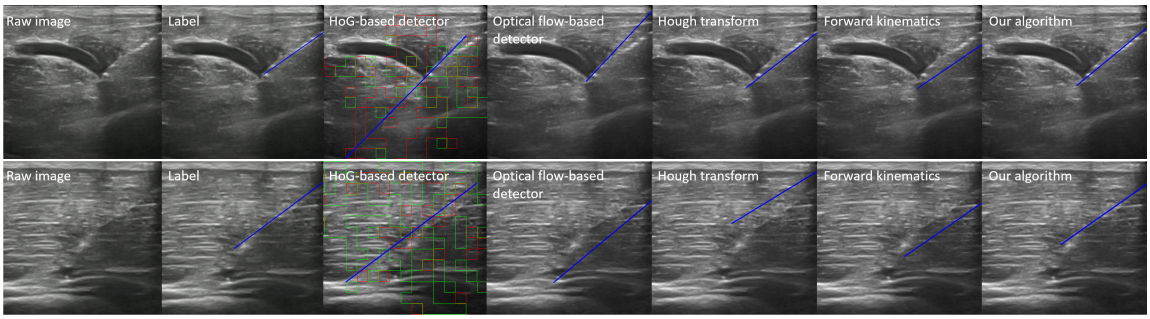

Figure 4.4: Example images for different needle tracking algorithms. The upper row is the tracking for artery insertion, and the lower row is the test trial of muscle insertion. The blue line is the estimate line in the image. The green and red boxes are the detected moving regions using the HoG-based motion detection.

Different algorithms have different dropout rates and tracking accuracy. The probabilistic Hough transform has the highest dropout rate since the complicated structures in the ultrasound disturb the detection, along with the needle being hard to perceive in some frames. The optical flow has a higher dropout rate, but the accuracy is lower than other detection because it is influenced by local motions such

as the pulses from a vessel. It also has difficulty detecting a needle when the needle insertion stops. The robotic kinematics has a zero dropout rate because it is stable data source, but the calibration error introduces large error in needle tracking. Our algorithm successfully combines their advantages to keep track of an inserted needle under difficult situations, providing a reliable estimation of the needle tip location for the controller of robotic needle insertion. For the needle angle estimation, the robotic kinematics has the lowest estimation error. The large inter-grader disagreement in the angle estimation shows that the needle angle is hard to estimate from the ultrasound images because the needle shaft is often partially visible. Hence the image-based segmentation degrades our algorithm's performance. However, the accuracy of tip localization is more important in our application.

To evaluate our optical flow-based motion detection algorithm, we compare it with the state-of-the-art Histogram-of-Gradient-based motion detector in [15]. Their algorithm compares the histogram of gradients of the current image with a pre-collected static image to find the moving region. We use an average of 50 images before the insertion start as the static image. The quantitative result in Table 4.1 shows that our proposed optical flow-based motion detector performs much better than the HoG-based tissue motion detector. Figure 4.4 shows that the HoG-based motion detector overestimates the moving region. This is because in the in-vivo experiments the breath and heart-beat of a subject can introduce external motions. Compared with the HoG-based detector, our proposed optical flow-based detector is more reliable in in-vivo studies. The optical flow-based detector only compares the current image with the previous image in a short time interval, so that accumulated large motions are ignored. Furthermore, we estimate a global affine motion from the motion flow so that the component of motion from probe movement can be eliminated.

4.3 Needle Insertion Experiments

In order to further evaluate the performance of our data fusion algorithm, we performed needle insertion experiments using the robotic system shown in Fig. 4.1. For each trial, the robot was moved to a new position on the chicken breast, the kinematics were reset to their original calibrated state, a target point was chosen within the

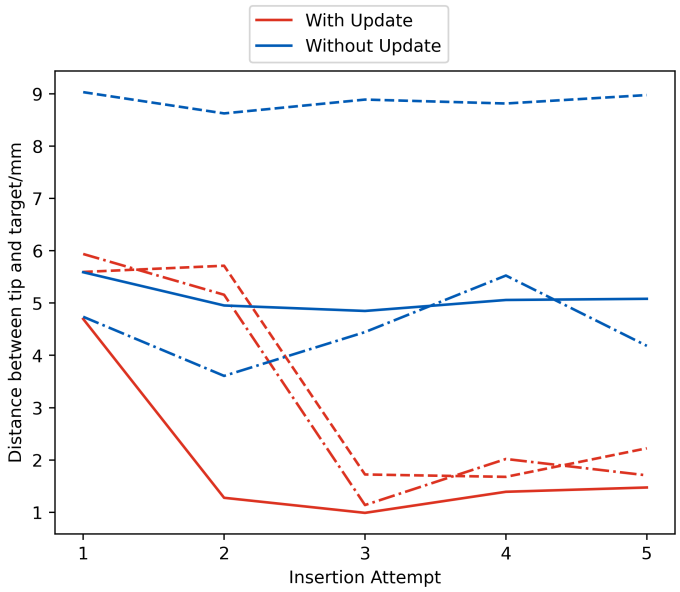

Figure 4.5: The change of the targeting error with different insertion attempts. The red lines are the test trials with kinematics update and the blue lines are the test trials without update. Different trials are distinguished by the linestyle.

workspace of the mechanism, and the needle was inserted five times. The needle tracking algorithm is used to track the needle and to provide kinematics update.

We tested the performance of the system with and without the kinematics update. The ground truth was estimated by asking one grader who was not present for the experiment to label the location of the needle tip when the insertion stopped. Fig. 4.5 shows that our proposed data fusion algorithm greatly improves the performance of the mechanism and reduces the distance of the needle tip to the goal. The targeting error is reduced 3.61 ± 0.44 mm on average when the controller has kinematics update and only 0.37 ± 0.23 mm without the update procedure. The percentage of improvement between the first and last attempts is 66.72% for the tests with kinematics update and only 7.15% for the tests without kinematics update. The results show that our information fusion-based algorithm is able to update the kinematics calibration parameters during the insertion to improve the targeting accuracy.

4.4 Conclusion

We introduce a novel data fusion algorithm to track a needle inserted by a robot which has higher tip localization accuracy then individual algorithms and a zero dropout rate. We also propose a new algorithm for needle detection in 2D ultrasound images by estimating the tissue motion induced by needle insertion, which is more robust than the state-of-the-art tissue motion-based detector. Our data fusion algorithm achieves robust on-line needle tracking, which can be applied in the control of robotic needle insertion.

Chapter 5

Planning Robotic Lateral Manipulation with Ordinary Deflecting Clinical Needles

Parts of Chapter 5 are adapted from the manuscript: **Wanwen Chen**, Blake Buchanan, Evan Harber, Howie Choset, and John Galeotti. "Planning Robotic Lateral Manipulation with Ordinary Deflection Clinical Needles by Potential Field-guided RRT."

This chapter presents the original work of Wanwen Chen, whose specific contributions as the first author of the publications include: developing the mechanical model, steering simulation and planning algorithms, analyzing the data, writing the manuscripts. Blake Buchanan's contributions include giving feedback on the mechanical model and planning algorithms. Evan Harber is responsible for data collection in the model verification experiment. Howie Choset and John Galeotti are the supervising faculty advisors.

This chapter proposes path planning methods based on a cantilever beam model. Two methods are compared, each of which is able to adapt to dynamic environments where both the target vessel and critical structures are moving. The primary contributions are: (1) a novel, broadly applicable model that is suitable for ordinary clinical needles of different stiffness, size and with either symmetric or asymmetric tips, (2) a novel artificial potential-guided Rapidly-exploring Random Tree (RRT)

planning algorithm that does not require specifying a fixed starting point and (3) extended planning methods that work in dynamic environments.

The method can be divided into two parts. First is the mechanical model of the needle-tissue interaction. In this part, the model predicts how the needle shape will change given the lateral manipulation, current tissue layout, and current needle shape in the simulation. The model is also used to calculate the Jacobian matrix to estimate how to move the needle base given the desired tip location. The second part is the planning method: given the tissue layout and the current tip location, the planning algorithm should be able to give a desired tip steering path.

5.1 Needle Tissue Interaction Modeling

Our mechanical model leverages [39] in the development of modeling the motion at the needle base. As in [39], we model the needle as a cantilever beam $y(x)$ with the free end inserted into the tissue, but we consider its base to move under actuation. This model assumes that when the needle is in its tip cutting path (the trajectory of the tip), the tissue does not have deformation. The tissue is modeled as a series of virtual springs connected to the needle, and the tip cutting path is the free end of the spring in its "relaxed" position. When an external force is applied at the base, the needle is pushed, moving it away from the tip cutting path, compressing or expanding the tissue as it moves. The amount by which the free end of the virtual spring was displaced from its "relaxed" position can be estimated by the difference between the shape of the needle and its natural cutting path.

The model assumes that the system is quasistatic to use the Rayleigh-Ritz method. The potential energy of the system is given by Eq. 5.1, where Π_{needle} is the strain energy stored in the needle, Π_{tissue} is the compression energy stored in the tissue, W_{base} is the work done by the lateral manipulation force, and W_{tip} is the cutting force introduced by the bevel tip.

$$\Pi = \Pi_{needle} + \Pi_{tissue} - W_{base} - W_{tip} \tag{5.1}$$

The needle bending energy is calculated using the Euler cantilever beam theory:

$$\Pi_{needle} = \int_0^l \frac{EI}{2} \left(\frac{\partial^2 y(x)}{\partial x^2} \right)^2 dx \tag{5.2}$$

where $y(x)$ is the shape of the needle shaft, E is the elastic modulus and I is the second moment of area of the needle.

The tissue strain energy is in Eq. 5.3 with the assumption that the tissue is not deformed when it is not cut:

$$\Pi_{tissue} = \int_d^l \frac{K}{2} \left(y(x) - y_{tip}(x) \right)^2 dx \tag{5.3}$$

where $y_{tip}(x)$ is the tip cutting path, and K is the tissue stiffness.

Finally, the work done by the lateral force and by the bevel tip cutting force is in Eq. 5.4 and Eq. 5.5, where F_l is the force acting at the needle base, and F_{tip} is the force acting at the needle tip.

$$W_{base} = F_l y(0) \tag{5.4}$$

$$W_{tip} = F_{tip} y(l) \tag{5.5}$$

From minimal potential energy, we have:

$$\delta \Pi = 0 \tag{5.6}$$

Previous work in [39] assumes that the base of the needle is fixed, so the solution y has a fixed boundary. Since we allow the base to move, we approximate y with the series,

$$y(x) = p(x) + \sum_{i=1}^{n} q_i(x) g_i \tag{5.7}$$

where $q_i(x)$ are the modal functions with zero boundary conditions used in [39] and g_i are the curvature coefficients to be solved. The $p(x)$ is used to model the moving boundary. In our case, the boundary is the location of the insertion point and the insertion angle. That is, we have $y(0) = d_y$ and $y'(0) = \theta$. Therefore, we use $p(x) = d_y + \theta x$ to model the non-zero boundary, and choose $q_i(x)$ with zero boundary conditions. After inserting Eq. 5.7 into the Eq. 5.6, we have Eq. 5.8 to solve the

unknown coefficients g_i, where g_i is $\mathbf{g}(i,1)$.

$$\Phi \mathbf{g} = F_{tip} + \Omega \tag{5.8}$$

The elements in Φ and the elements in Ω are in Eq. 5.9 and Eq. 5.10, respectively.

$$\Phi(i,j) = EI \int_0^l q_i'' q_j'' dx + K \int_{l-d}^l q_i q_j dx \tag{5.9}$$

$$\Omega(i,1) = K \int_{l-d}^l (y_{tip} - d_y - \theta x)\, dx \tag{5.10}$$

To solve the inverse kinematics of the needle we use the numerical Jacobian method in [16] to calculate the inverse Jacobian matrix. The control input is lateral motion and rotation at the needle base.

5.2 Potential Field-guided Rapidly-exploring Random Tree (PF-RRT)

The PF-RRT planning algorithm is based on Huang et al.'s implementation that grows the RRT from a given initial insertion point [26]. In our method, we extend the algorithm to be able to select the initial insertion point. Before growing the RRT, we employ an optimization-based method to select the initial insertion location. We first conduct an optimization that finds the straight line maximizing the sum of the distance between the line and the critical structures. For the line with insertion location y_{base} and angle θ, the line equation can be written as $y = y_{base} + x \tan\theta$. We select the initial insertion location and orientation that maximize the objective function in Eq. 5.11.

$$\begin{aligned} L(y_{base}, \theta) &= \sum_{n=1}^{N} (D_n - r_n) \\ s.t.\ y_{target} &= y_{base} + x_{target} \tan\theta \end{aligned} \tag{5.11}$$

where D_n is the distance from the center of the n-th critical structure to the pre-planned straight line, and r_n is the radius of the n-th critical structure. The initial insertion location y_{base} is used to grow an RRT using PF-RRT.

The PF-RRT algorithm is shown in Algorithm 2. The difference between the PF-RRT and classical RRT is that while the classical RRT grows the new node by random sampling F_{sample} only, the PF-RRT grows the new node by $F_{pf} + F_{sample}$, where F_{pf} is the direction of the gradient of the potential field at the randomly sampled point. Our potential field includes the repulsive field generated by the critical structures and the attractive field generated by the target [16], along with a curvature field, which is the square of the distance from the point (x, y) to the preplanned line. The curvature field is intended to reduce the curvature of the preplanned path. The total potential field is $U_{pf} = \mu_1 U_{repulsive} + \mu_2 U_{attractive} + \mu_3 U_{curvature}$, and each potential field is in Eq. 5.12, 5.13 and 5.14. μ_1, μ_2, μ_3 are the constant to normalize the potential field to have the max magnitude as 1.

$$U_{curvature}(x, y) = (x \tan\theta + y_{base} - y)^2/(1 + \tan^2\theta) \tag{5.12}$$

$$U_{attractive}(x, y) = \frac{1}{2}\left((x - x_{target})^2 + (y - y_{target})^2\right) \tag{5.13}$$

$$U_{repulse}(x, y) = \begin{cases} 1 - \left(\frac{x-x_{cs}}{a}\right)^2 - \left(\frac{y-y_{cs}}{b}\right)^2 & \left(\frac{x-x_{cs}}{a}\right)^2 + \left(\frac{y-y_{cs}}{b}\right)^2 < 1 \\ 0 & other \end{cases} \tag{5.14}$$

We then use the pruning method in [36] to prune the returned path. However, the path is still not optimal since the randomly selected points can be closed to the critical structures. We use spline interpolation to generate control points along the path, and directly optimize the control points to minimize the cost in Eq. 5.15. The cost includes the potential of the points on the path C_{field} (since the critical structures have higher potential and the target has low potential) and a continuous cost C_{con} (to reduce the length of the path).

$$C_{path} = w_1 C_{field} + w_2 C_{con} \tag{5.15}$$

where $C_{field} = \sum_{i=1}^{M} U_{pf}(x_i, y_i)$, and $C_{con} = \sum_{i=1}^{M} (y_i - y_{i-1})^2$.

Algorithm 2 Potential field-guided RRT algorithm.

procedure PF-RRT(x_{start}, I_{map})
  Graph vertices $V \leftarrow \{x_{start}\}$, edges $E \leftarrow \phi$, $N = 1$
  while no point in V reaches the target or $N < N_{max}$ **do**
    Randomly sample a point x_{sample} in I_{map}
    $x_{sample} = x_{sample} + F_{pf}$
    Find the nearest point $x_{nearest}$ in V
    $x_{new} = Steer(x_{nearest}, x_{sample})$
    if ObstacleFree($x_{new}, x_{nearest}, I_{map}$) **then**
      $V \leftarrow V \bigcup \{x_{new}\}$ $E \leftarrow E \bigcup \{(x_{new}, x_{nearest})\}$
  if Reach target **then**
    Find the shortest path in (V, E) that reaches the target
  Return Graph (V, E), Shortest path P

5.3 Replanning

This section introduces two different replanning algorithms.

5.3.1 Replanning RRT

The first replanning method is to re-run the PF-RRT algorithm introduced above, with the current tip location as the starting location. To increase the speed of replanning, the number of random sampling points is also linearly related to the random sampling area.

5.3.2 Bubble Bending

The second replanning algorithm is based on the modified version of Bubble Bending algorithm in [53]. The Bubble Bending algorithm models the path as an elastic band that deforms under the application of external and internal forces. In the simulation, the path is modeled as a sequence of spheres (bubbles). The external forces are generated from the critical structures to push the bubbles away, and the internal attractive forces are generated between the consecutive bubbles to reduce the displacement. The main algorithm of bubble bending replanning is listed in Algorithm 3, which includes three steps.

Algorithm 3 Bubble bending replanning.

procedure BUBBLEBENDINGREPLANNING($\mathbf{p}_{bubble}, I_{map}$)
  $\mathbf{p}_{new}$ = BubbleBending($\mathbf{p}_{bubble}, I_{map}$)
  $\mathbf{p}_{new}$ = BubbleReorganize($\mathbf{p}_{new}, I_{map}$)
  $\mathbf{p}_{new}$ = Smoothing($\mathbf{p}_{new}, I_{map}$)
  Return new bubble center $\mathbf{p}_{new}$

Algorithm 4 Bubble bending.

procedure BUBBLEBENDING($\mathbf{p}_{bubble}, I_{map}$)
  $i = 0$
  $U_{repulse} = U_{repulse}(I_{map})$
  $U_{curvature} = U_{curvature}(I_{map})$
  $\mathbf{F} = -\nabla(U_{repulse} + U_{curvature})$
  $\delta_{moving} = 0$
  $\mathbf{p}_{new} = \mathbf{p}_{bubble}$
  while $i < I_{max}$ and $\delta_{moving} > \delta_{threshold}$ **do**
    for $j = 2 : N_{bubble} - 1$ **do**
      $\mathbf{v}_{j-1} = \text{normalize}(\mathbf{p}_{new}^{j-1} - \mathbf{p}_{new}^{j})$
      $\mathbf{v}_{j+1} = \text{normalize}(\mathbf{p}_{new}^{j+1} - \mathbf{p}_{new}^{j})$
      $\mathbf{f}_j = \mu\text{normalize}(\mathbf{F}(\mathbf{p}_{new}^{j})) + \mathbf{v}_{j-1} + \mathbf{v}_{j+1}$
      $\delta_{moving} = \delta_{moving} + ||\mathbf{f}_j||_2$
      $\mathbf{p}_{new}^{j} = \mathbf{p}_{new}^{j} + \lambda\mathbf{f}_j$
      $i = i + 1$
    $\delta_{moving} = 0$
  Return new bubble center $\mathbf{p}_{new}$

BubbleBending: This step is to adjust the centers of the bubbles with forces. In our case, the external forces generated from the critical structures are the forces from the potential field $U_{repulse} + U_{curvature}$ in Eq. 5.14 and 5.12. The algorithm iteratively moves each bubble center until it reaches equilibrium or else the number of iterations reaches the maximum iteration time, as shown in Algorithm 4.

BubbleReorganization: This step is to generate a new sequence of bubbles. After the BubbleBending step, some bubbles might no longer be overlapping, and some bubbles may have become redundant. In this step, we interpolate the bubble centers generated from BubbleBending to generate a smooth path and then re-generate overlapping bubbles around the path.

Smoothing: This step is to use optimization to smooth the locations of bubble

Algorithm 5 Path Smoothing.

procedure SMOOTHING($\mathbf{p}_{bubble}$)
$\mathbf{p}^1_{new} = \mathbf{p}^1_{bubble}$
$\mathbf{p}^{N_{bubble}}_{new} = \mathbf{p}^{N_{bubble}}_{bubble}$
$O_1 = \sum_{i=2}^{N_{bubble}-1} ||2\mathbf{p}^i - \mathbf{p}^{i+1} - \mathbf{p}^{i-1}||^2$
$O_2 = \sum_{i=2}^{N_{bubble}-1} ||\mathbf{p}^i - \mathbf{p}^i_{bubble}||$
$\mathbf{p}_{new} = \text{argmin}_{\mathbf{p}}\, O_1 + O_2$, s.t. $\mathbf{p}^1 = \mathbf{p}^1_{bubble}$, $\mathbf{p}^{N_{bubble}} = \mathbf{p}^{N_{bubble}}_{bubble}$
Return new bubble center $\mathbf{p}_{new}$

centers to smooth the generated path. We minimize two objective functions related to the path smoothness and the curvature constraints in Algorithm 5.

5.3.3 Replanning Simulation

We focus on the application of vascular intervention, so we use a simple model to simulate the motion of the critical structures and the target vessel in the simulation experiments in later sections. When the depth of the needle tip x_{tip} is smaller than the depth of the object ($x_{tip} > x_{object}$), the object is fixed to the virtual spring and moves along with the spring. When the difference between the depth of the object and the needle tip is smaller than a moving threshold δ_{moving} ($x_{object} - x_{tip} < \delta_{moving}$), the needle tip starts pushing the tissue in front of the tip so the object is moving with the tissue. We use Eq. 5.16 to model the motion of the critical structures and the target.

$$\begin{aligned} dx &= (exp(1 - \frac{x_{object} - x_{tip}}{\delta_{moving}}) - 1)/(exp(1) - 1) \times dx_{max} \\ dy &= \nu dx \end{aligned} \tag{5.16}$$

where dx_{max} is the max moving distance of the objects and ν is a given constant.

5.4 Experiments and Results

5.4.1 Model verification

We first measured the needle bending inside a phantom to verify the mechanical model using the robotics system in Chapter 4. The experiment configuration is shown in Fig. 5.1.

We inserted the needle into a phantom with a layer of water above. The ultrasound probe was placed above the water. After part of the needle was inserted into the phantom, we rotated the angular actuator to move the needle base and use the robot kinematics to localize the needle base. One human labeler annotated the needle in the ultrasound images. The images were collected with PaoLus UF-760AG Portable Diagnostic Ultrasound Imaging Equipment (FUKUDA DENSHI, UK) with a linear transducer. The phantom we used is an anthropomorphic tissue phantom with simulated blood vessels (Advanced Medical Technologies, WA). The needle is a 7cm 18 gauge guidewire introducer needle (Argon Medical Devices, TX). We measured the elasticity of the phantom to be 25.56kPa by measuring the deformation and the applied forces on the phantom.

We compare the Euclidean distance between the annotated needle tip and the prediction with and without the bending estimation. The results in Table 5.1 show that the bending effects can not be ignored in the lateral manipulation, and that our model reduces the error in tip localization. Fig. 5.2 visualizes the estimation of the needle shape from robot kinematics and our simulation, showing that the simulation model fits the annotation better.

Table 5.1: The mean Euclidean distance between the annotated tip and the tip prediction.

Trial	Number of Manipulation	Without Bending/mm	With Bending/mm
1	5	1.6683	1.3046
2	4	1.7882	1.4891

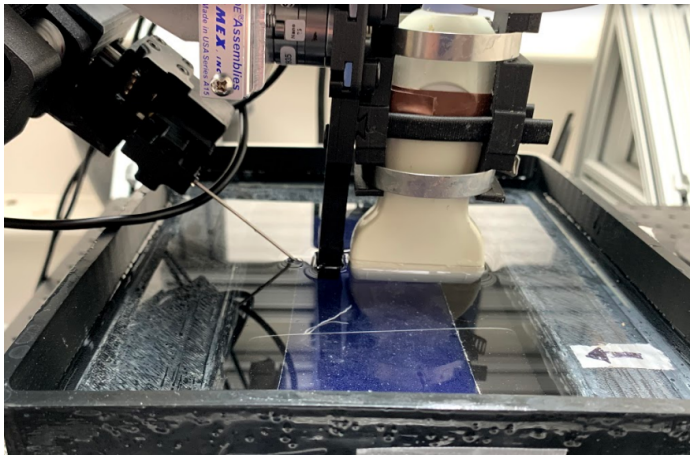

Figure 5.1: The experiment set up. The needle insertion mechanism and the ultrasound probe are attached to a robot arm.

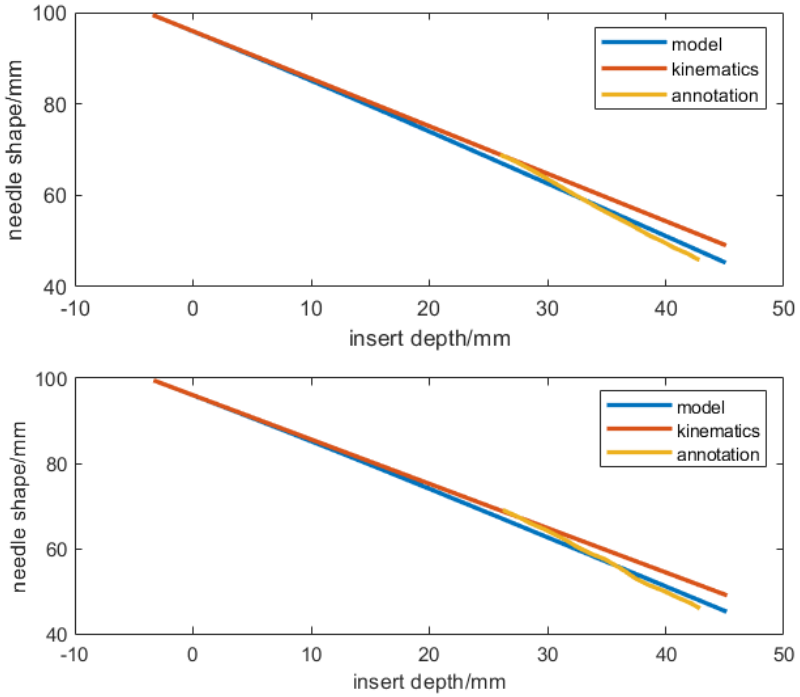

Figure 5.2: Results of the needle shape in the model verification. The "model" is the estimated needle shape in the simulation, "kinematics" is the estimated needle shape from robot kinematics, which assumes that the needle is a straight line. The "annotation" is the annotated needle shape from the ultrasound images.

Test Scene	Success Rate		Curvature	
	RRT	PF-RRT	RRT	PF-RRT
1	76.00%	72.00%	1017	1156
2	72.67%	74.67%	1512	1452
3	82.67%	86.67%	1582	1347
4	78.67%	81.33%	1517	1330

Table 5.2: The quantitative evaluation of the RRT and PF-RRT in four test scenes.

5.4.2 Simulation Experiments

The simulation code runs in Matlab R2020b. If not explicitly stated, the parameters are set as the values below. The tissue elasticity is 16.24kPa, the same value in [38]. The needle is a 22 gauge stainless steel needle with a symmetric tip. That is, the stiffness of the needle is 200GPa, the outer diameter is 0.711mm and the inner diameter is 0.394mm. The number of coefficients modeling the bending (the g_i in Eq. 5.7) is set to be 6.

Evaluation on the Preplanning

We first compare the RRT-based preplanning with and without the potential field guidance in four different configurations of the critical structures and the target. The evaluation metrics include (1) success rate, which is the rate that the planning method successfully generates an obstacle-free path, (2) the path curvature, which is defined as $\kappa = |y''|/(1 + y'^2)^{1.5}$. For each algorithm, we run 150 times to preplan a path for each given configuration. The results in Table 5.2 show that in most cases, the potential field guidance can increase the success rate of RRT and also reduce the max curvature of the preplanned path. The visualization of the preplanned path in the sample Fig. 5.3 shows that PF-RRT generates paths with higher quality and less abrupt turns.

Replanning Methods Comparison

We run the experiments with two different replanning algorithms: rerunning PF-RRT and Bubble Bending algorithm. We evaluate three different aspects of the planning.

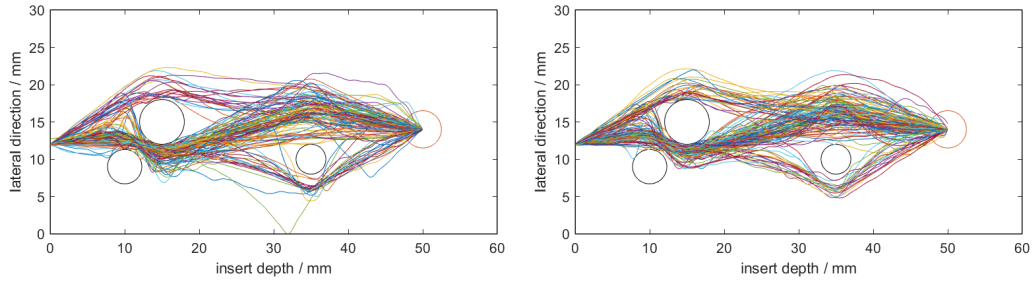

Figure 5.3: Example of preplanned paths in Test 3. The upper is planned with classical RRT and the lower one is planned with artificial potential field-guided RRT.

1. Smoothness of the replanned path. In this evaluation, we compare a metric that measures the curvature $\kappa = |y''|/(1 + y'^2)^{1.5}$ of the path, and we calculate the derivatives based on numerical differentiation.
2. Repeatability. This is to study whether the replanning algorithm is able to generate the same path in the same tissue configurations. To do this, we compare the standard deviation of the curvature and the needed lateral manipulation. We run the replanning algorithm with the same preplanned path and compare the lateral manipulation when the needle tip reaches the same depth. We report the max absolute magnitude of the lateral manipulation dy and $d\theta$, and the max standard deviation of the dy and $d\theta$. We also visualize the curve of the lateral manipulation to evaluate the repeatability qualitatively.
3. Sensitivity with respect to the noise in K. We execute the planning and steering simulation using the two different replanning algorithms with the same needle and tissue parameters. K is often hard to measure in actual tissue and is not homogeneous, so measuring K as in our compression experiments will often result in only a rough estimation of the whole system. To simulate inaccurate real-world estimation of K, we divide the simulated tissue into several layers and add Gaussian noise on the K in different layers. In the simulation, we set the width of the layer as 10mm and the standard deviation of the noise as 2kPa and 5kPa.

We design three different configurations of the target and the critical structures to evaluate the metrics mentioned above. For each test scene, we run the replanning algorithm 10 times, with the same setting of the parameters and preplanned path. The

Table 5.3: The results of the maximum curvature during the insertion under different level of noise in K.

Test		Scene 1		Scene 2		Scene 3	
Noise in K	Algorithm	mean	std	mean	std	mean	std
0kPa	rrt	175840	54158	30056	3806	122070	40632
	bubble	1174	0	3164	0	3988	0
2kPa	rrt	230010	96148	36221	18279	89150	36098
	bubble	1175	3	3151	26	3999	18
5kPa	rrt	212830	100400	34836	10980	105710	59015
	bubble	1181	12	3156	29	4010	17

results are shown in Table 5.3, 5.4 and 5.5. Figure 5.4 and 5.5 show the manipulation curve in test scene 1.

The results in Table 5.3 show that Bubble Bending generates smoother paths, since the max curvature in all test scenes are all much smaller than the path generated by PF-RRT. Also, the standard deviation of the max curvature mostly increases with the noise in K when using PF-RRT, but it remains relatively small when using Bubble Bending. All these indicate that the Bubble Bending algorithm can generate a smoother path and is less influenced by noise in K.

The results in Table 5.4 and 5.5 show that the Bubble Bending algorithm does not necessarily introduce smaller operation magnitude. It shows that the smoother tip steering path does not guarantee that the needed lateral manipulation is also small. However, the smaller standard deviation indicates that the Bubble Bending algorithm has better repeatability, especially when the noise in K is large. Accordingly, the Bubble Bending algorithm might be more stable in the real world, when actual tissue is imperfectly approximated by K.

Sensitivity Analysis of Imaging Error

Computer vision-based segmentation algorithms usually return noisy and inaccurate segmentations, so we also investigate multiple simulations of the Bubble Bending algorithm to test its performance under different noise levels when estimating the needle's location. Since the needle shape has the parameters of y_{base}, θ_{base} and the curvature coefficients g_i in Eq. 5.7, we add noises on these parameters directly. We

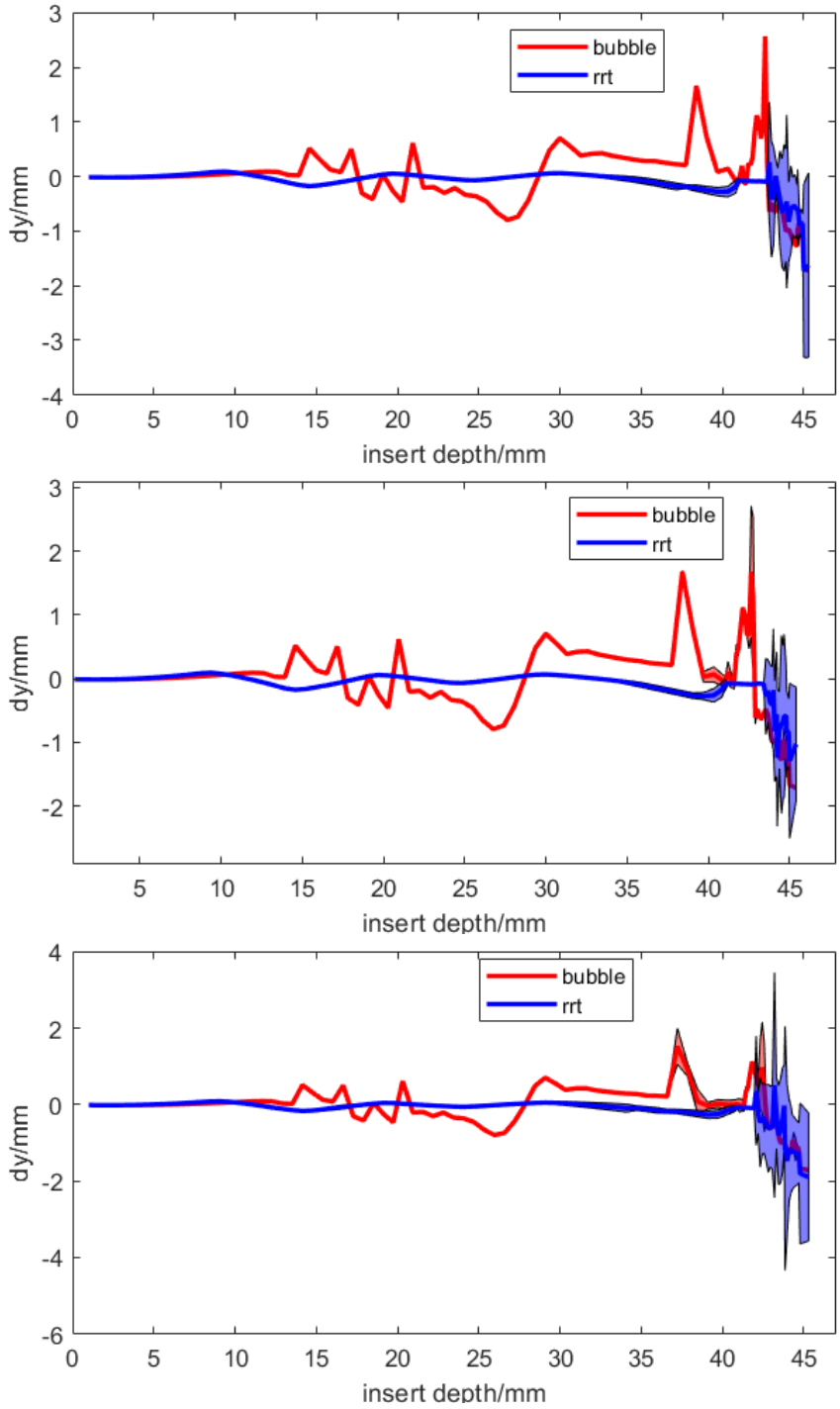

Figure 5.4: The lateral translation during the insertion under different level of noise. The upper image is without noise and the lower image is with 5kPa noise.

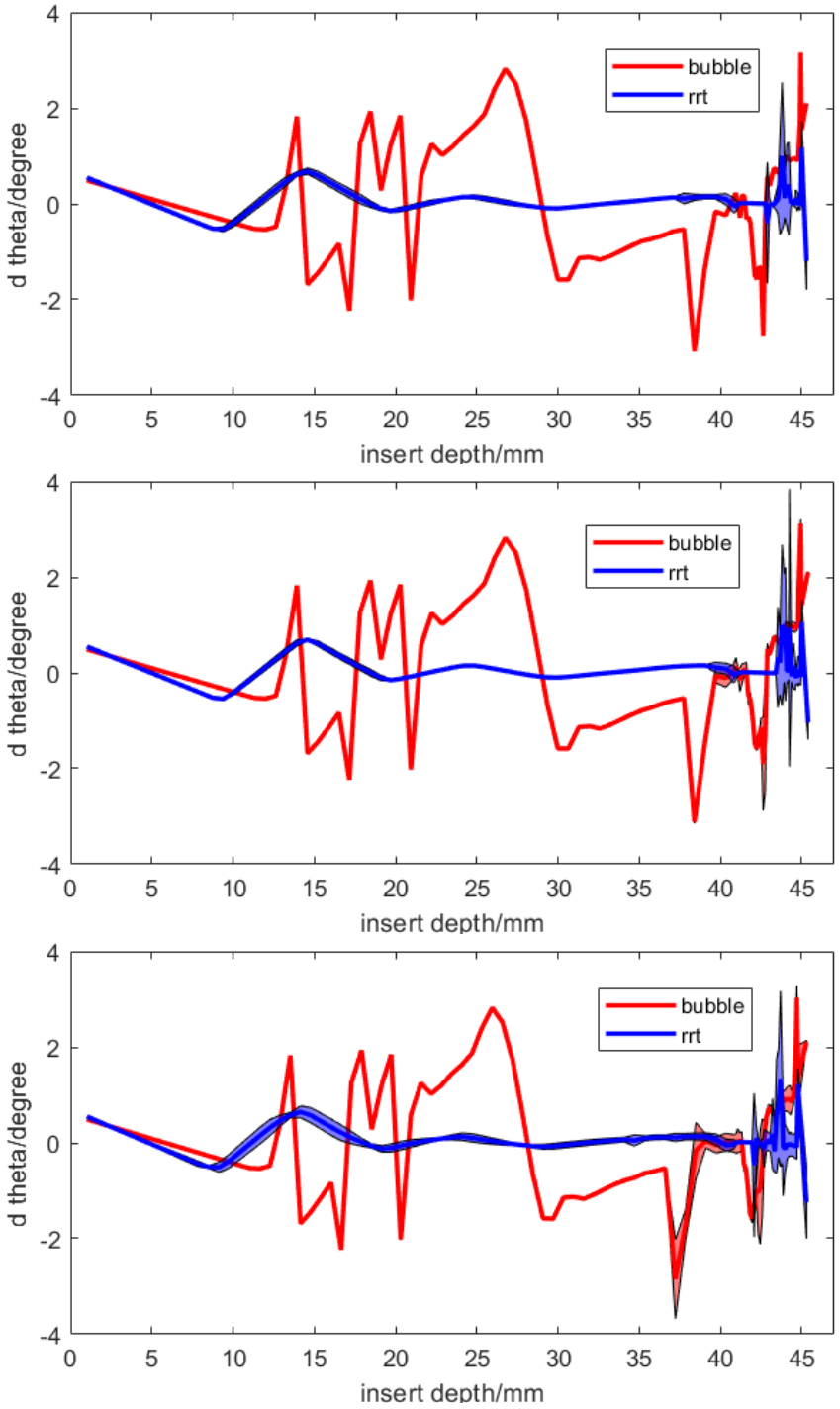

Figure 5.5: The lateral rotation during the insertion under different level of noise. The upper image is without noise and the lower image is with 5kPa noise.

Table 5.4: The results of the maximum lateral translation operation during the insertion under different level of noise in K.

Test		Scene 1		Scene 2		Scene 3	
Noise in K	Algorithm	max mean	max std	max mean	max std	max mean	max std
0kPa	rrt	1.73	1.69	1.94	1.68	2.90	1.41
	bubble	2.57	4.68E-16	2.51	4.68E-16	3.36	4.68E-16
2kPa	rrt	1.29	1.40	1.78	2.16	1.41	2.34
	bubble	1.72	1.25	2.34	0.87	2.85	1.51
5kPa	rrt	1.89	3.19	1.90	2.01	1.31	3.48
	bubble	1.71	1.19	2.22	0.79	2.80	1.46

Table 5.5: The results of the maximum lateral rotation during the insertion under different level of noise in K.

Test		Scene 1		Scene 2		Scene 3	
Noise in K	Algorithm	max mean	max std	max mean	max std	max mean	max std
0kPa	rrt	1.20	1.54	3.72	1.58	1.69	4.13
	bubble	3.16	4.68E-16	4.50	4.68E-16	5.01	9.36E-16
2kPa	rrt	1.06	2.91	3.58	1.61	1.53	3.69
	bubble	3.12	1.18	4.34	1.32	5.02	2.86
5kPa	rrt	1.34	1.84	3.36	1.79	1.78	3.86
	bubble	3.04	1.18	3.55	2.23	5.01	2.61

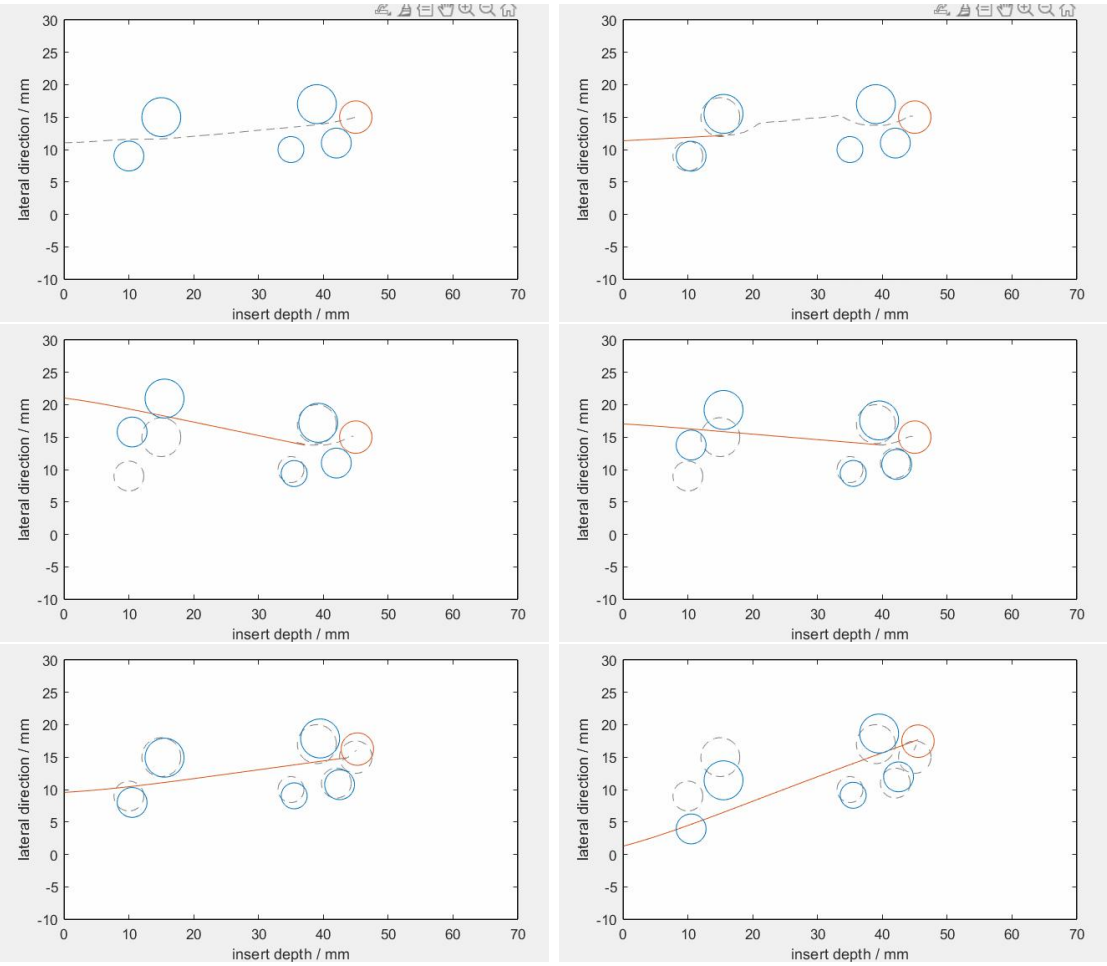

Figure 5.6: An example of Bubble Bending Replanning in test scene 1 with noise level being 0.

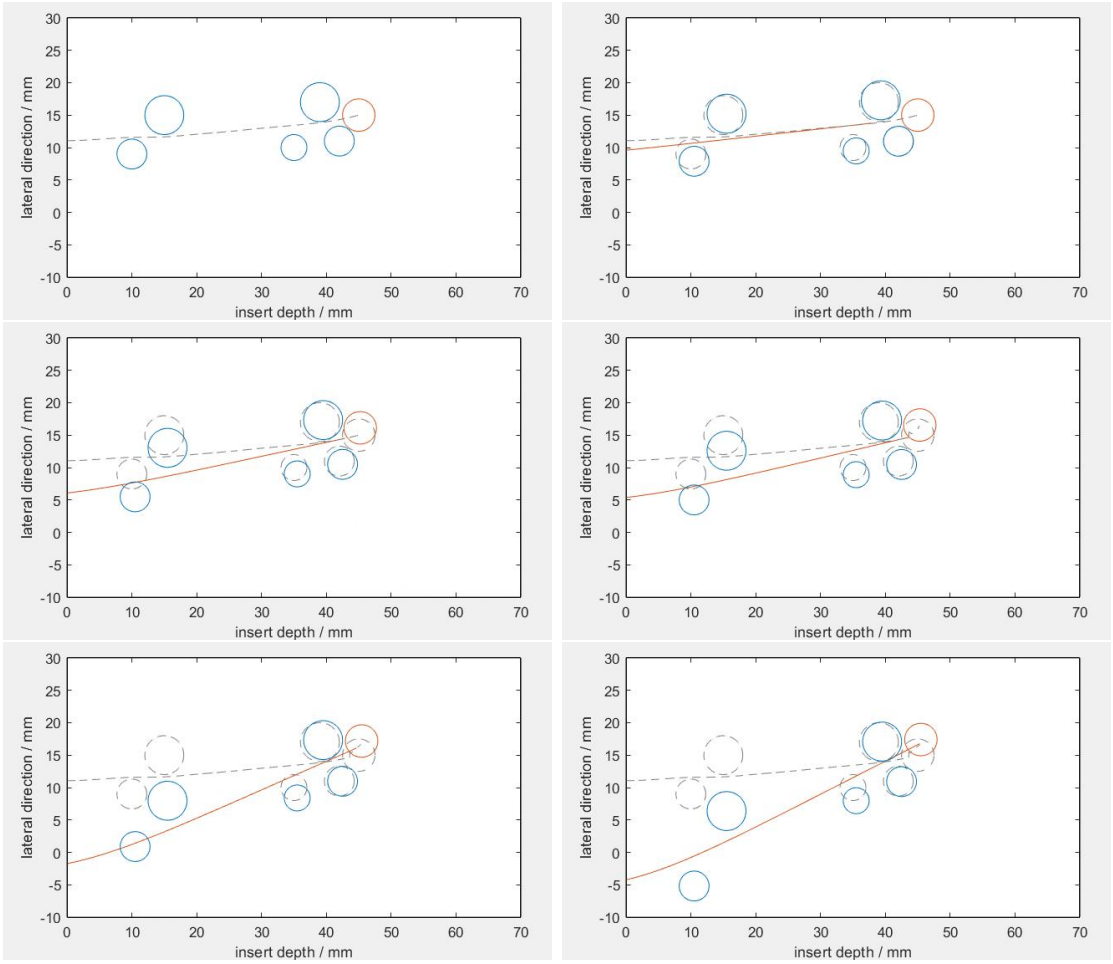

Figure 5.7: An example of PF-RRT replanning in test scene 1 with noise level being 0.

add Gaussian noise on the y_{base} with different levels of standard deviation. The noise levels on y_{base} were 0.05mm, 0.1mm and 0.15mm. We also add Gaussian noise on the curvature coefficients with different levels of standard deviation, with the cut-off that the noise should not exceed one standard deviation. The noise levels on the curvature coefficients were $0\%, 1\%, 5\%, 10\%, 15\%, 20\%$. We used the Bubble Bending replanning algorithm, and the results are shown in Figure 5.8. The error in visual measurement mainly influences the distance to the targets but does not show a clear trend on influencing the needle curvature, which is directly related to the needle strain.

5.5 Conclusions

In this chapter, we use the cantilever beam model with a virtual-spring needle-tissue interaction model to simulate the needle bending and use the PF-RRT and Bubble Bending planning algorithm to plan the external lateral motion of a robot to steer an ordinary clinical needle towards a moving vessel. Future work could include testing with more arbitrary motions of the targets and critical structures as well as the combination with image-based feedback to better steer the needle towards an internal moving target.

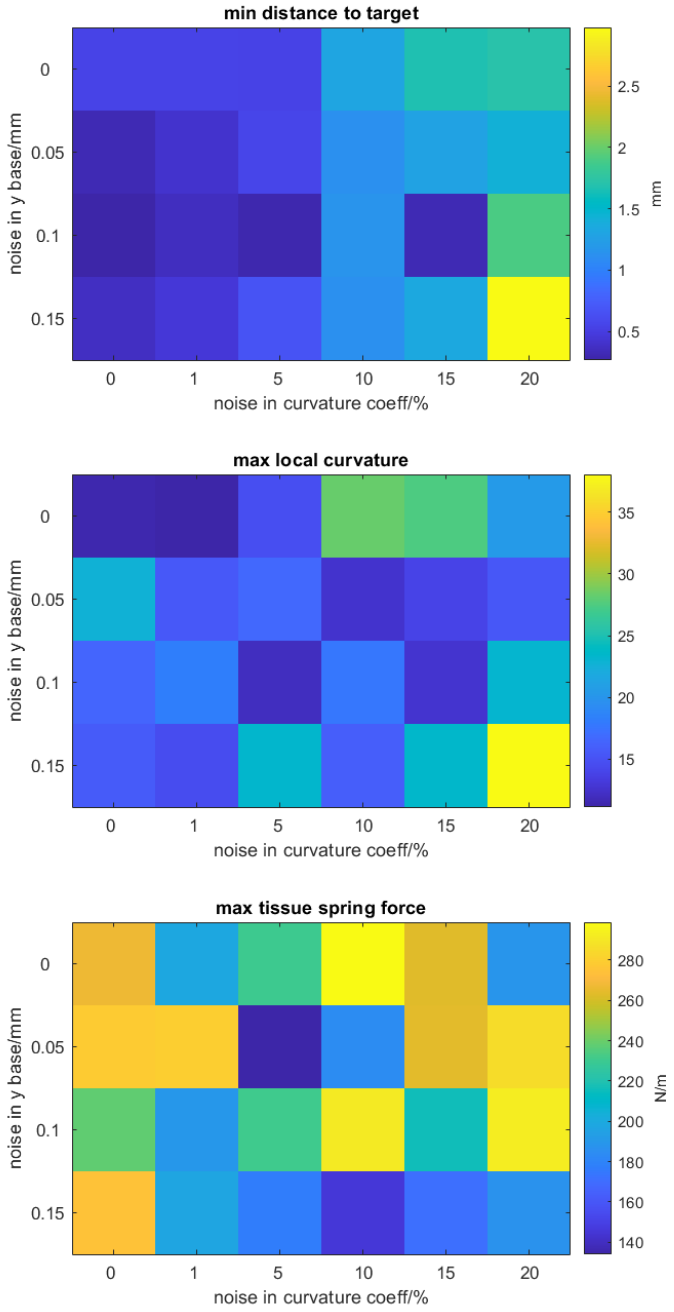

Figure 5.8: Sensitivity of the needle shape to the visual feedback.

Chapter 6

Conclusions

This thesis focuses on solving the challenges brought by needle-tissue interaction during autonomous needle insertion. The needle-tissue interaction can introduce needle bending, which brings difficulties to track a needle inside 2D ultrasound and challenges when planning lateral manipulation for needle steering. The bending needle requires a computer vision algorithm to segment a curved needle which is often only partially visible. It also requires mechanical modeling of the needle bending and dynamic replanning of the steering path to adapt to the motion of the critical structures and the targets inside the tissue introduced by the needle-tissue interaction.

The thesis develops several algorithms to target the problems stated above. Chapter 3 describes a new algorithm to segment and track the curved needle robustly even if the needle is partially visible in 2D ultrasound images. The proposed algorithm is able to localize the partially visible curved needle more robustly than both classical computer vision algorithms and a U-Net model, both in phantom experiments and in in-vivo experiments. It can rely only on ultrasound images, or it can additionally rely on a combination of external sensors. Chapter 4 proposes a robust online algorithm to track the needle utilizing both information from computer vision and robot kinematics, which has the potential for providing real-time tracking and image feedback for autonomous needle insertion. A new algorithm is introduced that localizes the needle based on the analysis of the tissue micro-motion introduced by the needle insertion, which outperforms the state-of-the-art micro motion-based needle localization algorithm. In Chapter 5, needle bending inside the tissue is mechanically

modeled to develop two replanning algorithms to enable robotic needle steering towards a moving target. The method is a fully autonomous replanning algorithm without the need for a human to select a first insertion point or select from preplanned paths.

Potential future work includes:

First, combine the curved needle tracking algorithm with the information fusion-based tracking algorithm. The curved needle tracking algorithm is running at around 10 fps, which is slower than the kinematics-based estimation. Our proposed adaptive Kalman filter has the potential to take the observation from the curved needle to track the needle shape. However, the Kalman filter currently models the dynamic procedure of inserting a straight needle. Therefore, a new model needs to be designed to track a curved needle using Kalman filtering. One possible approach is to use the parameters of the polynomials in the state of the Kalman filter.

Second, in the curved needle tracking, a U-Net-based segmentation algorithm was also proposed. While the U-Net does not segment a curved needle sufficiently accurately, it shows the potential of localizing the needle tip accurately. Therefore, combining the U-Net output with other classical computer vision algorithms using our novel Kalman filter has the potential to achieve higher accuracy of needle tip localization.

Third, the replanning algorithms are only validated in the simulation environment. The next step will be combining the tracking algorithm with the replanning algorithms on an actual robot to achieve a fully autonomous needle insertion. One of the difficulties is that our ultrasound probe has a narrow image width, so it is hard to segment the shape of the whole needle shaft inside opaque phantoms. Also, new algorithms are needed to track the moving critical structures and target(s). Another difficulty is to measure the mechanical parameters inside the real tissue. Other future directions include extending the mechanical model and the replanning algorithms to 3D environments.

Appendix A

Sensitivity Analysis of Critical Structure Configuration and Model Parameters

Appendix A is adapted from the manuscript: **Wanwen Chen**, Blake Buchanan, Evan Harber, Howie Choset, and John Galeotti. "Planning Robotic Lateral Manipulation with Ordinary Deflection Clinical Needles by Potential Field-guided RRT."

We provide a sensitivity analysis of the needle strain and the tissue tearing force with respect to the insertion depth, the needle stiffness and the noise in tissue elasticity. In the appendix, the model parameters are the same with Chapter 5 excepts that the outer diameter of the needle is 0.64mm, and the inner diameter is 0.337mm. The replanning algorithm is PF-RRT.

First is the sensitivity with respect to the insertion depth and the needle stiffness. Fig. A.1 shows that the needle strain increases when the insert depth increases or when the needle is more flexible. However, there might exist singularity locations where the needle strain suddenly increases or decreases with respect to the insertion depth. This problem can be caused by the tissue layout since the singularity points are scene-dependent. Fig A.2 shows that the maximum tissue force also follows a similar pattern.

We then provide a sensitivity analysis with respect to the error in the tissue parameter K, which is difficult to measure in live heterogeneous tissue. We divide the

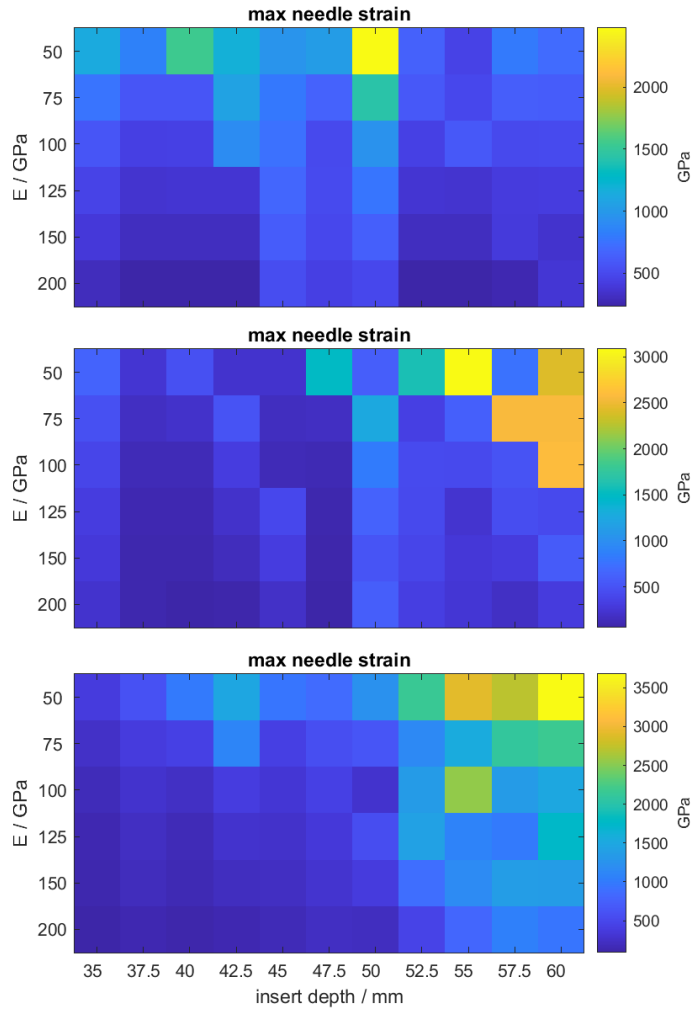

Figure A.1: The sensitivity analysis of the max needle strain with respect to the needle stiffness (y-axis) and the insertion depth (x-axis) in three different test scenes.

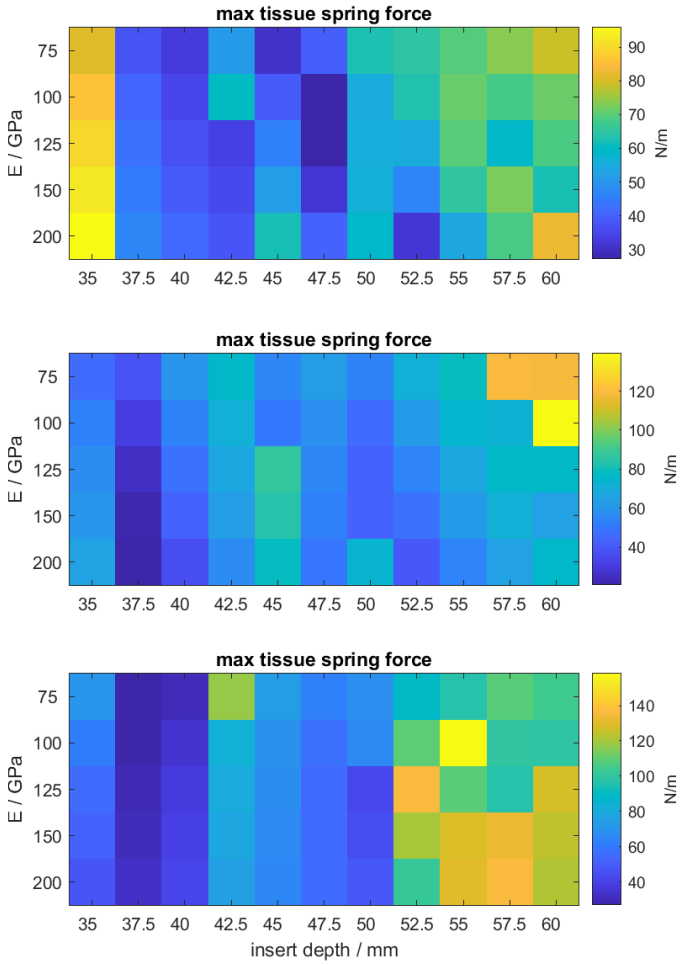

Figure A.2: The sensitivity analysis of the max tissue strain with respect to the needle stiffness (y-axis) and the insertion depth (x-axis) in three different test scenes.

tissue into several layers and add Gaussian noise on the K in different layers. In the simulation, we set the layer's width as 10mm and the standard deviation of the noise as 2kPa and 5kPa. Fig. A.3 shows that the max needle strain has more uncertainty and being different from the original pattern with the noise in K. Fig. A.4 shows that the max tissue spring force increases with added noise in the tissue elasticity but following a similar pattern in that the force increases with increasing insertion depth.

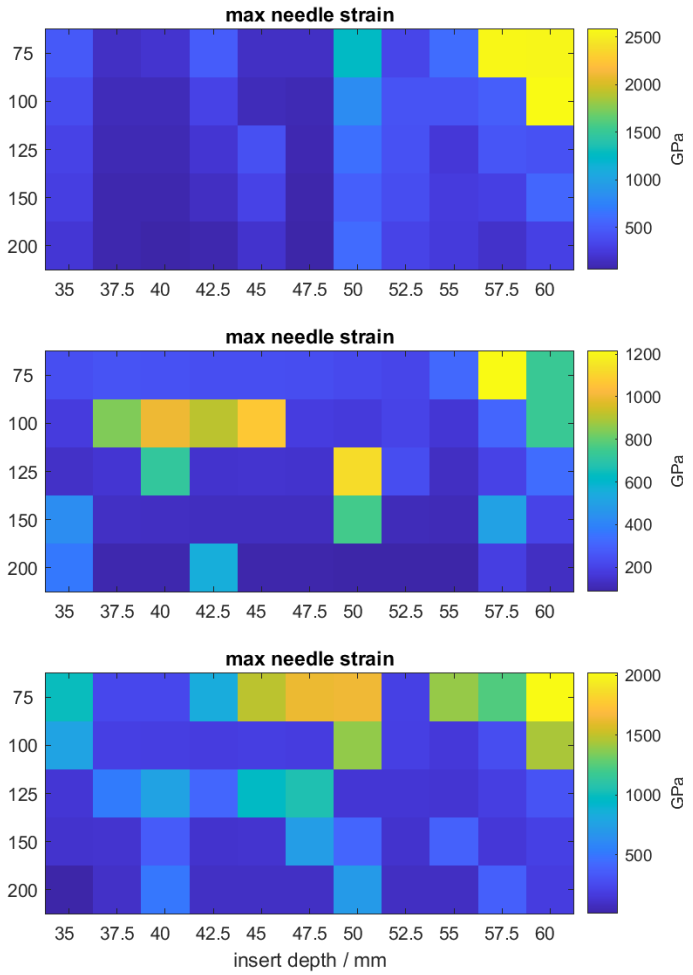

Figure A.3: The sensitivity analysis of the max needle strain with respect to the needle stiffness (y-axis) and the insertion depth (x-axis) with three different level of noise in K. The standard deviation of the noise (from top to the bottom) is 0kPa, 2kPa and 5kPa.

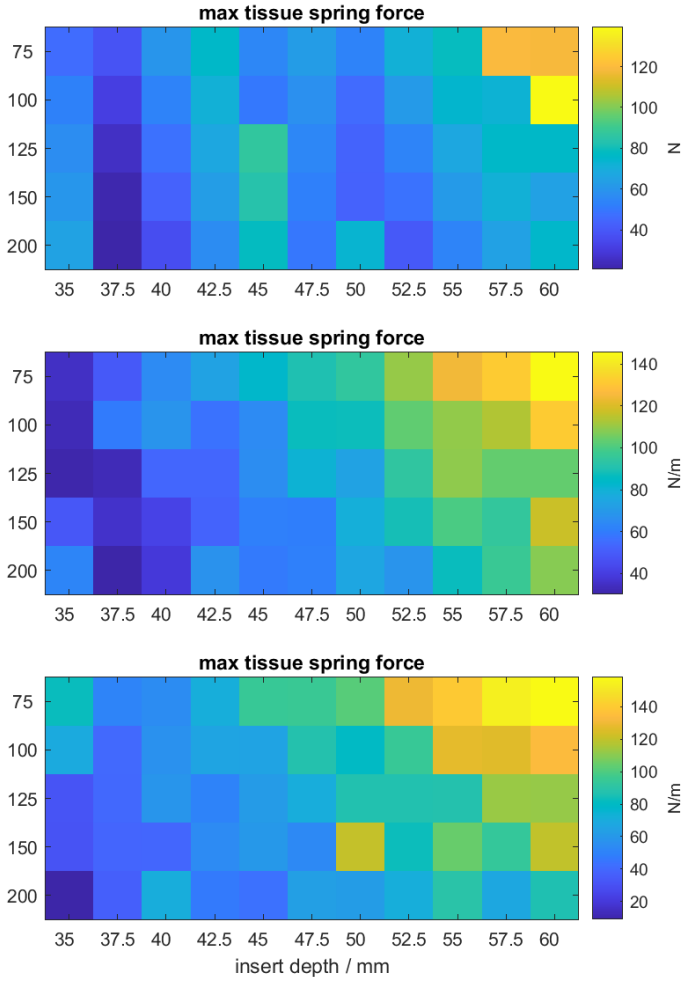

Figure A.4: The sensitivity analysis of the max tissue strain with respect to the needle stiffness (y-axis) and the insertion depth (x-axis) with three different level of noise in K. The standard deviation of the noise (from top to the bottom) is 0, 2kPa and 5kPa.

Bibliography

[1] Niki Abolhassani, Rajni Patel, and Mehrdad Moallem. Needle insertion into soft tissue: A survey. *Medical engineering & physics*, 29(4):413–431, 2007. 2.1.2, 2.2

[2] Yinoussa Adagolodjo, Laurent Goffin, Michel De Mathelin, and Hadrien Courtecuisse. Inverse real-time finite element simulation for robotic control of flexible needle insertion in deformable tissues. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2717–2722. IEEE, 2016. 2.2

[3] Ron Alterovitz, Ken Goldberg, and Allison Okamura. Planning for steerable bevel-tip needle insertion through 2d soft tissue with obstacles. In *Proceedings of the 2005 IEEE international conference on robotics and automation*, pages 1640–1645. IEEE, 2005. 2.2

[4] Ron Alterovitz, Andrew Lim, Ken Goldberg, Gregory S Chirikjian, and Allison M Okamura. Steering flexible needles under markov motion uncertainty. In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1570–1575. IEEE, 2005. 2.2

[5] A Ayadi, S Nicolau, B Bayle, P Graebling, and J Gangloff. Fully automatic needle calibration for robotic-assisted puncture on small animals. In *2007 IEEE/NIH Life Science Systems and Applications Workshop*, pages 85–88. IEEE, 2007. 2.1.4

[6] Ahmed Ayadi, Bernard Bayle, Pierre Graebling, and Jacques Gangloff. An image-guided robot for needle insertion in small animal. accurate needle positioning using visual servoing. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1453–1458. IEEE, 2008. 1

[7] Elif Ayvali and Jaydev P Desai. Accurate in-plane and out-of-plane ultrasound-based tracking of the discretely actuated steerable cannula. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5896–5901, Hong Kong, China, 2014. IEEE. 2.1.2

[8] Parmida Beigi, Robert Rohling, Tim Salcudean, Victoria A Lessoway, and Gary C Ng. Detection of an invisible needle in ultrasound using a probabilistic svm and time-domain features. *Ultrasonics*, 78:18–22, 2017. 2.1.3

[9] Thomas Brox, Andrés Bruhn, Nils Papenberg, and Joachim Weickert. High accuracy optical flow estimation based on a theory for warping. In *European conference on computer vision*, pages 25–36. Springer, 2004. 4.1.3

[10] Jay Carriere, Carlos Rossa, Nawaid Usmani, Ronald Sloboda, and Mahdi Tavakoli. Needle shape estimation in soft tissue based on partial ultrasound image observation. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2277–2282, Seattle, WA, USA, 2015. IEEE. 2.1.2

[11] GA Chapman, D Johnson, and AR Bodenham. Visualisation of needle position using ultrasonography. *Anaesthesia*, 61(2):148–158, 2006. 2.1.3

[12] Alvin I Chen. *Image-guided robotics for autonomous venipuncture*. PhD thesis, Rutgers University-Graduate School-New Brunswick, 2016. 2.1, 2.1.4, 3.1, 4.1.2

[13] Alvin I Chen, Max L Balter, Timothy J Maguire, and Martin L Yarmush. Real-time needle steering in response to rolling vein deformation by a 9-dof image-guided autonomous venipuncture robot. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2633–2638. IEEE, 2015. 1, 2.1

[14] Alvin I Chen, Max L Balter, Timothy J Maguire, and Martin L Yarmush. Deep learning robotic guidance for autonomous vascular access. *Nature Machine Intelligence*, 2(2):104–115, 2020. 2.1

[15] Joanna Czajkowska, Bartłomiej Pyciński, Jan Juszczyk, and Ewa Pietka. Biopsy needle tracking technique in us images. *Computerized Medical Imaging and Graphics*, 65:93–101, 2018. 2.1.3, 4.2

[16] Simon P DiMaio and Septimiu E Salcudean. Needle steering and motion planning in soft tissues. *IEEE Transactions on Biomedical Engineering*, 52(6):965–974, 2005. 2.2, 5.1, 5.2

[17] SP DiMaio, S Pieper, K Chinzei, N Hata, SJ Haker, DF Kacher, G Fichtinger, CM Tempany, and R Kikinis. Robot-assisted needle placement in open mri: system architecture, integration and validation. *Computer Aided Surgery*, 12(1): 15–24, 2007. 1

[18] Mingyue Ding and Aaron Fenster. A real-time biopsy needle segmentation technique using hough transform. *Medical physics*, 30(8):2222–2233, 2003. 2.1.1

[19] Yan Duan, Sachin Patil, John Schulman, Ken Goldberg, and Pieter Abbeel. Planning locally optimal, curvature-constrained trajectories in 3d using sequential convex optimization. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5889–5895. IEEE, 2014. 2.2

[20] Bita Fallahi, Carlos Rossa, Ron S Sloboda, Nawaid Usmani, and Mahdi Tavakoli. Sliding-based image-guided 3d needle steering in soft tissue. *Control Engineering*

Practice, 63:34–43, 2017. 1

[21] Gregory S Fischer, Iulian Iordachita, Csaba Csoma, Junichi Tokuda, Simon P DiMaio, Clare M Tempany, Nobuhiko Hata, and Gabor Fichtinger. Mri-compatible pneumatic robot for transperineal prostate needle placement. *IEEE/ASME transactions on mechatronics*, 13(3):295–305, 2008. 1

[22] Mengyu Fu, Alan Kuntz, Robert J Webster, and Ron Alterovitz. Safe motion planning for steerable needles using cost maps automatically extracted from pulmonary images. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4942–4949. IEEE, 2018. 2.2

[23] Mengyu Fu, Oren Salzman, and Ron Alterovitz. Toward certifiable motion planning for medical steerable needles. *arXiv preprint arXiv:2107.04939*, 2021. 2.2

[24] Charles Galamhos, Jose Matas, and Josef Kittler. Progressive probabilistic hough transform for line detection. In *Proceedings. 1999 IEEE computer society conference on computer vision and pattern recognition (Cat. No PR00149)*, volume 1, pages 554–560. IEEE, 1999. 3.1

[25] Ilker Hacihaliloglu, Parmida Beigi, Gary Ng, Robert N Rohling, Septimiu Salcudean, and Purang Abolmaesumi. Projection-based phase features for localization of a needle tip in 2d curvilinear ultrasound. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 347–354. Springer, 2015. 2.1.3

[26] Cheng Huang and Yong Lei. A novel model-based path planning method for robot-assisted flexible needle insertion. In *2017 13th IEEE Conference on Automation Science and Engineering (CASE)*, pages 1414–1419. IEEE, 2017. 1, 2.2, 5.2

[27] Kiros Karamanidis, Artemis Travlou, Peter Krauss, and Uwe Jaekel. Use of a lucas–kanade-based template tracking algorithm to examine in vivo tendon excursion during voluntary contraction using ultrasonography. *Ultrasound in medicine & biology*, 42(7):1689–1700, 2016. 4.1.3

[28] Mert Kaya and Ozkan Bebek. Needle localization using gabor filtering in 2d ultrasound images. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4881–4886. IEEE, 2014. 2.1.1

[29] Mert Kaya, Enes Senel, Awais Ahmad, and Ozkan Bebek. Visual needle tip tracking in 2d us guided robotic interventions. *Mechatronics*, 57:129–139, 2019. 2.1.1, 2.1.2

[30] MERT KAYA, ABDURRAHMAN ENES ŞENEL, and ÖZKAN BEBEK. Gabor filter-based localization of straight and curved needles in 2d ultrasound images. *TURKISH JOURNAL OF ELECTRICAL ENGINEERING & COMPUTER SCIENCES*, 28(5):2940–2955, 2020. 2.1.2

[31] Mohsen Khadem, Carlos Rossa, Ron S Sloboda, Nawaid Usmani, and Mahdi Tavakoli. Mechanics of tissue cutting during needle insertion in biological tissue. *IEEE Robotics and Automation Letters*, 1(2):800–807, 2016. 2.2

[32] Chunwoo Kim, Doyoung Chang, Doru Petrisor, Gregory Chirikjian, Misop Han, and Dan Stoianovici. Ultrasound probe and needle-guide calibration for robotic ultrasound scanning and needle targeting. *IEEE Transactions on Biomedical Engineering*, 60(6):1728–1734, 2013. 2.1.4

[33] Nahum Kiryati, Yuval Eldar, and Alfred M Bruckstein. A probabilistic hough transform. *Pattern recognition*, 24(4):303–316, 1991. 4.1.2

[34] Bardia Konh, Blayton Padasdao, Zolboo Batsaikhan, and Seong Young Ko. Integrating robot-assisted ultrasound tracking and 3d needle shape prediction for real-time tracking of the needle tip in needle steering procedures. *The International Journal of Medical Robotics and Computer Assisted Surgery*, 2021. 2.1.2

[35] Ting-Yu Lai, Hsiao-I Chen, Cho-Chiang Shih, Li-Chieh Kuo, Hsiu-Yun Hsu, and Chih-Chung Huang. Application of a novel kalman filter based block matching method to ultrasound images for hand tendon displacement estimation. *Medical physics*, 43(1):148–158, 2016. 4.1.3

[36] Xiaodong Lan and Stefano Di Cairano. Continuous curvature path planning for semi-autonomous vehicle maneuvers using rrt. In *2015 European control conference (ECC)*, pages 2360–2365. IEEE, 2015. 5.2

[37] Isabelle Leang, Stéphane Herbin, Benoît Girard, and Jacques Droulez. On-line fusion of trackers for single-object tracking. *Pattern Recognition*, 74:459–473, 2018. 4.1.4

[38] Thomas Lehmann, Carlos Rossa, Nawaid Usmani, Ronald Sloboda, and Mahdi Tavakoli. Deflection modeling for a needle actuated by lateral force and axial rotation during insertion in soft phantom tissue. *Mechatronics*, 48:42–53, 2017. 5.4.2

[39] Thomas Lehmann, Ronald Sloboda, Nawaid Usmani, and Mahdi Tavakoli. Model-based needle steering in soft tissue via lateral needle actuation. *IEEE Robotics and Automation Letters*, 3(4):3930–3936, 2018. 2.2, 5.1, 5.1, 5.1

[40] Zihao Li, Shuang Song, Li Liu, and Max Q-H Meng. Tip estimation method in phantoms for curved needle using 2d transverse ultrasound images. *Applied Sciences*, 9(24):5305, 2019. 2.1.2

[41] Michael H Loser and Nassir Navab. A new robotic system for visually controlled percutaneous interventions under ct fluoroscopy. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 887–896. Springer, 2000. 1

[42] Kim Mathiassen, Diego Dall'Alba, Riccardo Muradore, Paolo Fiorini, and Ole Jakob Elle. Robust real-time needle tracking in 2-d ultrasound images using statistical filtering. *IEEE Transactions on Control Systems Technology*, 25 (3):966–978, 2016. 2.1.1, 2.1.4

[43] Davneet S Minhas, Johnathan A Engh, Michele M Fenske, and Cameron N Riviere. Modeling of needle steering via duty-cycled spinning. In *2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 2756–2759. IEEE, 2007. 2.2

[44] Cosmas Mwikirize, John L Nosher, and Ilker Hacihaliloglu. Convolution neural networks for real-time needle detection and localization in 2d ultrasound. *International journal of computer assisted radiology and surgery*, 13(5):647–657, 2018. 2.1.1

[45] Cosmas Mwikirize, John L Nosher, and Ilker Hacihaliloglu. Signal attenuation maps for needle enhancement and localization in 2d ultrasound. *International journal of computer assisted radiology and surgery*, 13(3):363–374, 2018. 2.1.1

[46] Cosmas Mwikirize, John L Nosher, and Ilker Hacihaliloglu. Single shot needle tip localization in 2d ultrasound. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 637–645. Springer, 2019. 2.1.1

[47] Zipi Neubach and Moshe Shoham. Ultrasound-guided robot for flexible needle steering. *IEEE Transactions on Biomedical Engineering*, 57(4):799–805, 2009. 1, 2.2, 3.1, 4.1.3

[48] Akiva Novoselsky, Shlomo E Sklarz, and Miya Dorfan. Track to track fusion using out-of-sequence track information. In *2007 10th International Conference on Information Fusion*, pages 1–5. IEEE, 2007. 4.1.4

[49] Allison M Okamura, Christina Simone, and Mark D O'leary. Force modeling for needle insertion into soft tissue. *IEEE transactions on biomedical engineering*, 51(10):1707–1716, 2004. 2.2

[50] Stephen H Okazawa, Richelle Ebrahimi, Jason Chuang, Robert N Rohling, and Septimiu E Salcudean. Methods for segmenting curved needles in ultrasound images. *Medical image analysis*, 10(3):330–342, 2006. 2.1.2

[51] Wooram Park, Kyle B Reed, Allison M Okamura, and Gregory S Chirikjian. Estimation of model parameters for steerable needles. In *2010 IEEE International Conference on Robotics and Automation*, pages 3703–3708. IEEE, 2010. 2.2

[52] Sachin Patil, Jessica Burgner, Robert J Webster, and Ron Alterovitz. Needle steering in 3-d via rapid replanning. *IEEE Transactions on Robotics*, 30(4): 853–864, 2014. 2.2

[53] Marlene Pinzi, Thomas Watts, Riccardo Secoli, Stefano Galvan, and Ferdinando Rodriguez y Baena. Path replanning for orientation-constrained needle steering. *IEEE Transactions on Biomedical Engineering*, 68(5):1459–1466, 2021. 1, 2.2, 5.3.2

[54] G Reusz, P Sarkany, J Gal, and A Csomos. Needle-related ultrasound artifacts and their importance in anaesthetic practice. *British journal of anaesthesia*, 112 (5):794–802, 2014. 2.1.3

[55] Joseph M Romano, Robert J Webster, and Allison M Okamura. Teleoperation of steerable needles. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 934–939. IEEE, 2007. 2.2

[56] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 3.2

[57] Yun-Hsuan Su, Kevin Huang, and Blake Hannaford. Real-time vision-based surgical tool segmentation with robot kinematics prior. In *2018 International Symposium on Medical Robotics (ISMR)*, pages 1–6. IEEE, 2018. 2.1.4

[58] Wen Sun, Jur van den Berg, and Ron Alterovitz. Stochastic extended lqr for optimization-based motion planning under uncertainty. *IEEE Transactions on Automation Science and Engineering*, 13(2):437–447, 2016. 2.2

[59] Chenliang Tang, Gaosheng Xie, Olatunji Mumini Omisore, Jing Xiong, and Zeyang Xia. A real-time needle tracking algorithm with first-frame linear structure removing in 2d ultrasound-guided prostate therapy. In *2019 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 1240–1245. IEEE, 2019. 2.1.1

[60] Daniel Tenbrinck, Sönke Schmid, Xiaoyi Jiang, Klaus Schäfers, and Jörg Stypmann. Histogram-based optical flow for motion estimation in ultrasound imaging. *Journal of mathematical imaging and vision*, 47(1):138–150, 2013. 4.1.3

[61] Ryosuke Tsumura and Hiroyasu Iwata. Trajectory planning for abdominal fine needle insertion based on insertion angles. *IEEE Robotics and Automation Letters*, 2(2):1226–1231, 2017. 4.1.4

[62] Ryosuke Tsumura, Jin Seob Kim, Hiroyasu Iwata, and Iulian Iordachita. Preoperative needle insertion path planning for minimizing deflection in multilayered tissues. *IEEE robotics and automation letters*, 3(3):2129–2136, 2018. 2.2

[63] Gustaaf J Vrooijink, Momen Abayazid, and Sarthak Misra. Real-time three-dimensional flexible needle tracking using two-dimensional ultrasound. In *2013 IEEE International Conference on Robotics and Automation*, pages 1688–1693, Karlsruhe, Germany, 2013. IEEE. 2.1.2

[64] Michael Waine, Carlos Rossa, Ronald Sloboda, Nawaid Usmani, and Mahdi Tavakoli. 3d shape visualization of curved needles in tissue from 2d ultrasound images using ransac. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4723–4728, Seattle, WA, USA, 2015. IEEE. 2.1.2

[65] Adam C Waspe, H Jason Cakiroglu, James C Lacefield, and Aaron Fenster. Design, calibration and evaluation of a robotic needle-positioning system for small animal imaging applications. *Physics in Medicine & Biology*, 52(7):1863, 2007. 2.1.4

[66] Robert J Webster, Jasenka Memisevic, and Allison M Okamura. Design considerations for robotic needle steering. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 3588–3594. IEEE, 2005. 2.2

[67] Robert J Webster III, Jin Seob Kim, Noah J Cowan, Gregory S Chirikjian, and Allison M Okamura. Nonholonomic modeling of needle steering. *The International Journal of Robotics Research*, 25(5-6):509–525, 2006. 2.2

[68] Andreas Wedel, Annemarie Meißner, Clemens Rabe, Uwe Franke, and Daniel Cremers. Detection and segmentation of independently moving objects from dense scene flow. In *International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 14–27. Springer, 2009. 4.1.3

[69] Fei Xu, Dedong Gao, Shan Wang, and A Zhanwen. Mlesac based localization of needle insertion using 2d ultrasound images. In *Journal of Physics: Conference Series*, volume 1004, page 012037. IOP Publishing, 2018. 2.1.1

[70] Jijie Xu, Vincent Duindam, Ron Alterovitz, and Ken Goldberg. Motion planning for steerable needles in 3d environments with obstacles using rapidly-exploring random trees and backchaining. In *2008 IEEE international conference on automation science and engineering*, pages 41–46. IEEE, 2008. 2.2

[71] Yupei Zhang, Zhen Tian, Yang Lei, Tonghe Wang, Pretesh Patel, Ashesh B Jani, Walter J Curran, Tian Liu, and Xiaofeng Yang. Automatic multi-needle localization in ultrasound images using large margin mask rcnn for ultrasound-guided prostate brachytherapy. *Physics in Medicine & Biology*, 65(20):205003, 2020. 2.1.2

[72] Baoliang Zhao, Long Lei, Liang Xu, Shibo Li, Ying Hu, Jianwei Zhang, Xiaojun Yang, and Yue Zhang. Needle deflection modeling and preoperative trajectory planning during insertion into multilayered tissues. *IEEE/ASME Transactions on Mechatronics*, 2020. 2.2