TOWARDS A 3D ULTRASOUND IMAGING SYSTEM FOR NEEDLE-GUIDED MINIMALLY INVASIVE PROCEDURES

By

COSMAS BYOTARIHO MWIKIRIZE

A dissertation submitted to the

School of Graduate Studies

Rutgers, The State University of New Jersey

In partial fulfillment of the requirements

For the degree of

Doctor of Philosophy

Graduate Program in Biomedical Engineering

Written under the direction of

Dr. Ilker Hacihaliloglu

And approved by

_____

_____

_____

_____

New Brunswick, New Jersey

October 2019

ABSTRACT OF THE DISSERTATION

Towards A 3D Ultrasound Imaging System for Needle Guided Minimally Invasive Procedures

By COSMAS BYOTARIHO MWIKIRIZE

Dissertation Director: Ilker Hacihaliloglu

Needle-based minimally invasive procedures such as percutaneous biopsies, regional anesthesia and peripheral vascular interventions are ubiquitous in healthcare. During these procedures, accurate real-time needle localization is vital to minimize risk and improve efficacy. For this purpose, image guidance is utilized, and ultrasound is the desirable modality because of its ease of use, low-cost, non-ionizing radiation and real-time imaging capabilities. However, with ultrasound, needle visualization is inhibited by non-axial specular reflection, hyperechoic artifacts and noise. This dissertation describes the development of accurate, automatic, real-time and robust enhancement and localization techniques for the needle shaft and tip in challenging ultrasound-guided procedures. The developed methods employ both traditional computer vision and advanced machine learning approaches for modeling ultrasound signal transmission, feature detection and pixel-wise classification of needles from 2D and 3D ultrasound data. In extensive *ex vivo* imaging studies on realistic phantoms across a broad range of imaging settings and scenarios, the methods accurately localize needles that have low contrast or are imperceptible to the naked eye. The results of these studies will inform future clinical trials for evaluating feasibility of our methods. Once translated, this work will provide means for a new ultrasound imaging platform to support real-time enhancement and localization of needles and will be applicable to commercially available 2D/3D cart-based and portable ultrasound systems thus benefitting clinical practice, research and industry.

PREFACE

This dissertation is based on several manuscripts resulting from work done by the author of this dissertation who was responsible for: 1) developing the described algorithms, 2) designing and conducting the described experimental studies, 3) generating the presented data and figures, 4) writing, editing, and revising the original text, and 5) addressing reviewers' comments as the corresponding author of the original manuscript submissions. The author received guidance on algorithm development, experiment protocols and editing the manuscripts from the co-authors.

Chapter 2 describes a version of the published work in the following publications:

- **Mwikirize C**, Nosher JL, Hacihaliloglu I (2018) Signal attenuation maps for needle enhancement and localization in 2D ultrasound. *Int J CARS, 13(3): 363-374*.
- **Mwikirize C,** Nosher JL, Hacihaliloglu I (2016) Enhancement of Needle Tip and Shaft from 2D Ultrasound Using Signal Transmission Maps. *MICCAI LNCS*, 9900: 362-369

Chapter 3 describes a version of the published work in:

- **Mwikirize C,** Nosher JL, Hacihaliloglu I (2018) Convolutional Neural Networks for Real-Time Needle Detection and Localization in 2D Ultrasound. *Int J CARS*, 13 (5), 647-657

Chapter 4 describes a version of the published work in:

- **Mwikirize C**, Nosher JL, Hacihaliloglu I (2019) Learning needle tip localization from digital subtraction in 2D ultrasound. *Int J CARS,* 14(6), 1017-1026.

Chapter 5 describes a version of the unpublished work in:

- **Mwikirize C**, Nosher JL, Hacihaliloglu I (2019) Single shot needle tip localization in 2D ultrasound. *MICCAI 2019* (In Press).

Chapter 6 describes a version of the published work in:

- **Mwikirize C**, Nosher JL, Hacihaliloglu I (2017) Local Phase-Based Learning for Needle Detection and Localization in 3D Ultrasound. In: Cardoso M. et al. (eds) Computer Assisted and Robotic Endoscopy and Clinical Image-Based Procedures. *CARE-CLIP 2017. MICCAI LNCS*, 10550: 108-115.

ACKNOWLEDGEMENTS

This work wouldn't have been possible without the immense support of my advisor, Prof. Ilker Hacihaliloglu who provided mentorship and guidance throughout the course of my research. Words are not enough to express my appreciation for his contribution to nurturing me into a better scientist. I am also grateful to members of my dissertation committee: Dr. Mark Pierce, Dr. Nada Boustany and Dr. John L. Nosher for their valuable time, advice and feedback. Special thanks go to Prof. Troy Shinbrot for his friendship and guidance. I also owe gratitude to the Fulbright program and Greg Costalas, my program advisor at Rutgers, as well as Larry Stromberg, our Department administrator for the unwavering assistance.

I'm grateful to the current members and alumni of the Computer Assisted Surgery and Therapy (CompAST) lab: Ahmed Alsinan, Xiao Qi, Prajna Desai, Xuxin Zeng, Matthew Richtmyer, Justin Mohabir, Hridayi Patel, Amrut Sarangi, Robert Battikha, Mikhiel Lim, Jaclyn Calia and Idin Jafari. You made the work environment conducive and enjoyable. My friends and colleagues in the Biomedical Engineering Department, including members of the Biomedical Engineering Society: Adhithi Kanthan, Naser Mohammed, Rohit Rao, Fernando Rebolledo, Paulina Krzyszczyk, Carolina Bobadilla, Ileana Marrero-Berrios, Ayesha Aijaz, Jeremy Anderson, Elliot Dolan, Zachary Fritz, Alison Acevedo, Joseph Molde, Joseph Sheba, AJ Adejare and all my former classmates, thank you for the companionship and for organizing the extra curricula activities which took the weight off our hectic schedule. Estelle Nkoutche, thank you for being a great friend and for the constant encouragement.

Lastly, to my family: I owe it all to you. Thank you for loving me and supporting me from thousands of miles away.

DEDICATION

To my parents: Samuel and Agripina Byotariho

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

2D	Two dimensional
3D	Three dimensional
ADC	Analog to Digital Converter
Adam	Adaptive Momentum Estimation
B-mode	Brightness mode
BN	Batch Normalization
CI	Confidence Interval
CNN	Convolutional Neural Network
CT	Computed Tomography
dB	decibels
DL	Deep Learning
ED	Euclidean Distance
ELU	Exponential Linear Unit
FC	Fully Connected
FCN	Fully Convolutional Network
FF	Feed Forward
FOV	Field of View
FPS	Frames Per Second
GPU	Graphics Processing Unit
HOG	Histogram of Oriented Gradients
HT	Hough Transform
Hz	hertz
k-NN	k Nearest Neighbor
mHz	megathertz
MIP	Maximum Intensity Projection
MLESAC	Maximum Likelihood Sampling Consensus
mAP	mean average precision
mm	millimeter
MRI	Magnetic Resonance Imaging
PIP	Parallel Integral Projection

PoC	Point of Care
PS	Phase Symmetry
RANSAC	Random Sampling Consensus
R-CNN	Region Based Convolutional Neural Network
ReLU	Rectified Linear Unit
RF	Radiofrequency
RMSE	Root Mean Square Error
ROI	Region of Interest
RPN	Region Proposal Network
RT	Radon Transform
SGD	Stochastic Gradient Descent
SIFT	Scale Invariant Feature Transformation
SVM	Support Vector Machine
TEE	Transesophageal Echo
TF	Top-hat filter
TRUS	Transrectal Ultrasound
TV	Total Variation
TVR	Total Variation Regularization
US	Ultrasound

Chapter 1 -Introduction

1.1 Clinical challenges in ultrasound-guided minimally invasive procedures

Minimally invasive procedures involve insertion of a surgical tool and guiding it towards an anatomical target. This necessitates high precision handling of the surgical tool, so that the tool and its tip are accurately localized. In this work, we focus on guidance of needles which are used in a myriad of diagnostic and therapeutic procedures such as percutaneous biopsies [1] (kidney, liver, pancreas, breast, prostate, lung), regional anesthesia [2] (peripheral nerve blocks, spinal/epidural anesthesia), peripheral vascular interventions [3,4] (central venous cannulation, angioplasty, stenting, atherectomy) and prenatal interventions [5].

The key to improving efficacy of these procedures while minimizing risk of injury to the patient is the use of image guidance, which should ideally enable real-time needle visualization throughout the procedure. Ultrasound (US) has emerged as the most popular imaging modality in these interventions because it facilitates real-time intra-procedural visualization of the needle, has no radiation exposure risk, and is widely more available than alternative technologies such as X-Ray, Computed Tomography (CT) and magnetic resonance imaging (MRI) [6,7]. Further, US gives flexibility to insert the needle in multiple anatomical planes and is less time consuming. US has been shown to improve efficacy and safety of central venous access [8] and peripheral nerve blocks [9,10].

The gold standard in US guided procedures is two-dimensional (2D) US. The transducer is placed on the patient's skin in the anatomical region of interest

(ROI), and the needle is advanced toward the target via an appropriate trajectory. Two approaches can be used: In-plane insertion, in which the transducer axis is colinear with the needle or out-of-plane insertion, in which the transducer axis and the shaft are orthogonal [11]. These approaches are illustrated in Figure 1-1.

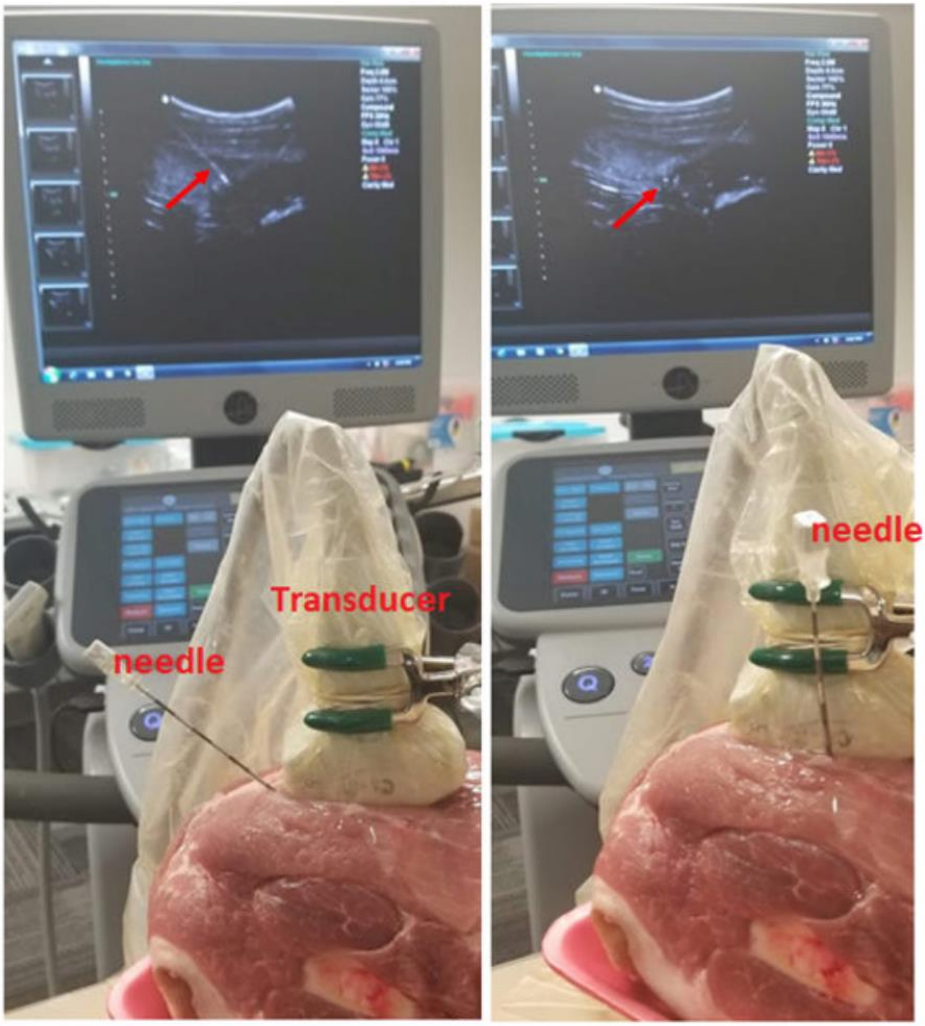


Figure 1-1. Illustrating in-plane (left) and out-of-plane insertion (right) for a 17G needle inserted in a porcine shoulder phantom. The red arrows on the US monitor points to the visible needle features. For in-plane insertion, we can see the whole needle as a bright hyperechoic line while for out-of-plane insertion, only the tip is visible as a bright dot.

Preference for either technique usually depends on the radiologist's skill and type of procedure. In-plane insertion is suitable for shallow targets and where there is less interfering anatomy between the insertion point and the target, for instance during a liver biopsy. Ideally, it should provide full visualization of the entire needle shaft and tip, resulting from reflection of the US beam off the entire needle length and subsequent reception of the reflected beam by the transducer. This is often not the case. First, 2D US has a narrow field of view (FOV). The US beam has a narrow width (as narrow as 1mm at the focal zone of the transducer). Thus, it is difficult to keep the whole needle aligned with the imaging plane, even for experienced users.

Out-of-plane insertion on the other hand minimizes the distance through which the needle must travel to reach the target. This reduces tissue trauma and is usually useful if in-plane trajectory is untenable due to interfering anatomy. The out-of-plane technique is typically used for accessing vascular and nerve targets (these are often easier to visualize in cross section). However, only a small section of the needle is visible: a cross section of the shaft may be mistaken to be the shaft, and the needle tip only becomes visible when it intersects with the US beam. Therefore, radiologists usually rely on observing tissue distortion to localize the tip.

In addition, steep orientation of the needle with respect to the US beam (required for deep targets) causes nonaxial specular reflection off the needle surface, where the beam is reflected away from the probe [12]. When this happens, the needle shaft will appear discontinuous and/or the tip will be invisible. In both techniques, when the needle trajectory contains interfering dense anatomy such

as bone or highly echogenic soft tissue interfaces, these may be mistaken for the needle. Furthermore, the US beam suffers attenuation which degrades image quality as image depth increases, and US images are generally affected by speckle noise. All these limitations contribute to reducing needle visibility in US.

To overcome this clinical challenge, radiologists use several methods to recover needle visualization. These include transducer manipulation by translation or rotation, needle jiggling, needle rotation to improve reflection off the bevel, stylet movement if the needle has one, and hydrolocation. Despite these maneuvers, needle visibility is still a challenge. An invisible needle can have detrimental effects on procedures, such as reduced efficacy, secondary morbidity or even mortality. When multiple failed needle passes are made, there is increased likelihood of inadvertent neural, visceral or vascular puncture, leading to injury, infections and other post-procedure complications. For example, accidental injection of steroids in vasculature during epidural anesthesia has been associated with neurologic complications [13]. Accidental arterial puncture leading to embolic stroke has been reported during ultrasound-guided central venous catheterization [14]. Post-procedure complications lead to increased hospital stays, and hence increased healthcare costs.

To comprehend the technical challenges associated with needle visibility in US, it is important to first understand the interaction phenomena between US and the imaging media. We present this in the next section.

1.2 Basics of ultrasound

Sound waves are mechanical impulses which exist as vibrations of pressure in a medium. Sound is characterized by its frequency in hertz (Hz), wavelength in millimeters (mm) and amplitude in decibels (dB). The speed of sound c in a medium is related to the medium's adiabatic elastic bulk modulus, a quantity that describes how resistant a substance is to compressibility:

$$c = \sqrt{\frac{K}{\rho}}, \tag{1.1}$$

where K is the adiabatic elastic bulk modulus and ρ is the density of the medium. This would suggest that sound should travel slower in dense media, but this is often not the case because dense media are also usually stiff.

US is sound whose frequency is beyond the hearing ability of humans (> 20 kHz). Medical US is typically in the range of 1 megahertz (MHz) to 20 MHz. The central piece of the US system is the transducer, also known as a probe. The transducer surface consists of an array of elements which transmit and receive US signals to and from tissue. The transmitted signals are known as pulses while the reflected signals are echoes. In modern transducers, the elements are typically piezoelectric crystals, which produce pulses when excited by an electrical impulse and convert the backscattered echoes into an electric signal. Each transducer has a characteristic frequency.

When US pulses are sent into tissue, they are either *reflected*, *refracted*, *scattered*, *absorbed*, or *refracted*. Reflection is the most important phenomenon

for US image formation. The reflected US (echo) energy received by the transducer is what is converted into the US image. Each medium is associated with *acoustic impedance* Z, a US imaging parameter related to the speed at which pulses travel in that medium:

$$Z = \rho c, \tag{1.2}$$

where c is the speed of sound and ρ is the medium's density. Acoustic impedance varies between different biological tissue. The average soft tissue speed of US is $1540\ ms^{-1}$.

Reflection occurs at boundaries of media with different acoustic impedances. When US pulses encounter a medium with greater acoustic impedance, reflection will occur. If the boundary encountered has a relatively larger dimension than the US wavelength, specular (regular) reflection will occur. Figure 1-2 illustrates specular reflection. The reflection coefficient relates the amount of incident US energy to the amount of reflected US energy. With reference to Figure 1-2 where US crosses an interface between media of acoustic impedances $\boldsymbol{Z_1}$ and $\boldsymbol{Z_2}$, the reflection coefficient is given by:

$$\frac{I_r}{I_i} = \frac{(Z_1 - Z_2)^2}{(Z_1 + Z_2)^2} \tag{1.3}$$

Non-specular reflection occurs when the wavelength of the incident US waves is of the order of or greater than the size of the particles in the tissue. This effect leads to scattering, in which the incident beam is reflected in many directions. The scattered waves undergo interference, leading to granular noise

known as speckle, which reduces the signal to noise ratio of the backscattered waves [15].

Absorbed acoustic energy is converted into heat and does not contribute to image formation. Absorption increases with viscosity of the medium, as well as frequency of US. For this reason, we expected metallic surfaces and bone to have high absorption capability. In case there is total absorption, no acoustic energy reaches the region below that surface. We then say the surface has created an acoustic shadow in its wake.

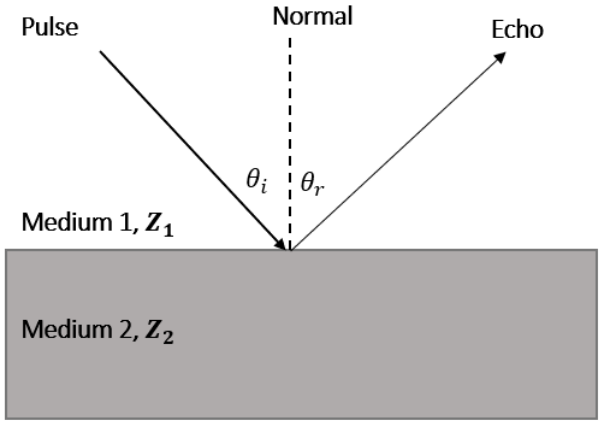


Figure 1-2. Specular reflection at an interface between two media of different acoustic impedances Z_1 and Z_2. The angle of incidence θ_i is equal to the angle of reflection θ_r. This type of reflection occurs if the structure encountered has dimension greater than the US wavelength. If $\theta_i = 90^\circ$, maximum energy from the echo will be received by the transducer. If $\theta_i < 90^\circ$, there is a likelihood of the echo not being received by the transducer. When this happens, we say non-axial specular reflection has occurred.

Refraction on the other hand causes a change in wavelength of the US beam as it travels across an interface, with different velocities on either side. It

doesn't contribute to US image formation, although it leads to a reduction in beam intensity. Previous reflections, refraction, scattering and absorption contribute to attenuation; the loss in intensity of the US pulses as they travel deeper into tissue. Attenuation occurs for both forward waves and backscattered waves. It is directly proportional to frequency of the US signal and the distance travelled by the signal till it returns to the transducer from the target. For this reason, low frequency transducers are preferable for imaging deep anatomy.

The reflected US signal undergoes several processing steps to yield the image observed on the US screen. This processing pipeline is described in the next section.

1.3 The ultrasound image formation process

In clinical practice, radiologists usually use brightness mode (B-mode) images. These are formed from a processing pipeline that starts with radiofrequency (RF) data generated from backscattered acoustic waves received by the US transducer. On the transducer, the angle between the extreme transducer elements; the transducer angle, determines the FOV. The FOV is that portion of the transducer angle which is effective in a scan, and its angular extent is the sector angle. The sector angle is divided by scan lines, which are narrow beams for propagation paths of the US signal, along which the reflected signals are measured.

Since anatomical structures have different acoustic impedances, and each will produce a pulse of commensurate amplitude, there are several reflected

signals of different amplitude over time for each scan line. These analog signals are converted into electrical signals by the transducer and used to generate RF data for each scan-line through sampling and discretization. The B-mode image is generated from aggregating and processing RF data associated with all the coplanar scanlines. In the B-mode image, anatomical structures which are strong reflectors are brighter than weak reflectors. Figure 1-3 shows a generic pipeline for generation of a B-mode frame, consisting of the following processes:

Time gain compensation: Since interfaces far from the transducer receive a generally weaker pulse due to the previous reflections, absorption and scattering, they produce weaker reflections. Time-gain compensation is applied to correct this phenomenon by proportionately amplifying signals from distant interfaces.

Beamforming: Echoes from a reflector in the scanned medium may hit the transducer elements at different times since they travel through different distances. Beamforming on each scan line applies appropriate time delays on the signals to establish synchronism before they are aggregated. The analog signals received by each transducer element are first digitized by an analog-to-digital converter (ADC) before being fed into a beamforming channel. Consequently, the resulting signal has a higher signal to noise ratio, enhanced angular localization and better spatial resolution. The result is the *post beamformed RF data*.

Filters: Band pass filters are applied to the post beamformed RF signal to remove noise and limit it to an appropriate bandwidth appropriate to the transmit frequency and the bandwidth of the transducer.

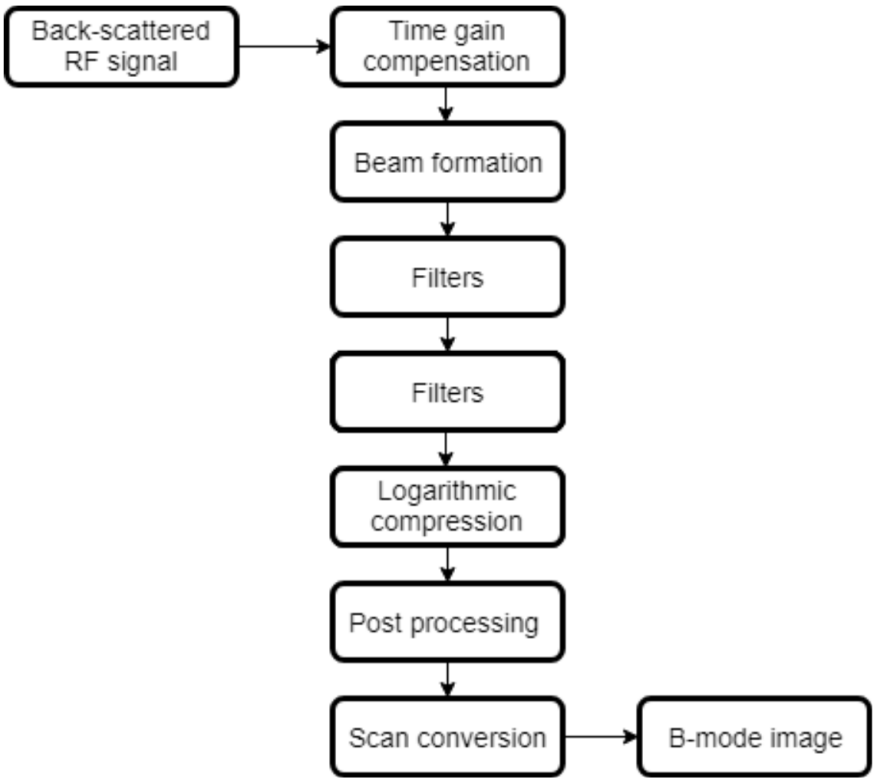


Figure 1-3. Pipeline of B-mode image formation in the pulse-echo method. The analog signals sampled along scanlines are subjected to pre-processing to generate the RF data. After envelope detection, RF data is compressed, post-processed and scan converted to generate the B-mode image. It is possible to intervene at any of the processing stems in order to enhance a specific anatomical feature or surgical instrument.

Envelope detection: The envelope of the RF signal is computed to generate the amplitude of the signal. This is analogous to a rectification operation in which the negative values of the signal are inverted.

Logarithmic Compression: The dynamic range of the human eye may not be able to perceive the range of the envelope detected RF signal (usually up to 80dB). This would make establishing proper contrast between different structures difficult.

To improve contrast, the amplitude of the signal is logarithmically compressed. Mapping functions are utilized to improve contrast by limiting the difference between the highest and lowest values in the signal. Depending on the feature of interest, the mapping function may be tweaked to either suppress or emphasize it. For example, needles which are highly echogenic may benefit from a mapping function that emphasizes the upper echelons of the signal. B-mode images utilize an 8-bit grayscale color map (0-255). The values of signal are translated to this color map.

Post-processing: This is performed to improve image quality or enhancement through noise reduction. Post-processing targets removal of speckle and other image artifacts. The filters of choice are applied on individual scan lines.

Scan conversion: The image at this stage is contained in a parallel grid, which must be converted to the correct geometrical representation cognizant of the transducer type. This is achieved through scan conversion. In 2D US, B-mode images are typically acquired at a frame rate between 5 and 80 frames per second (fps). The frame rate for three-dimensional (3D) US is much less (< 5 fps).

Different features will display different contrast in the US image depending on their location in the imaging medium, their orientation to the US beam, and their physical properties. In the next section, we describe the factors influencing appearance of needles in the US image.

1.4 Appearance of needles in ultrasound images

As pulses from the US transducer travel through the tissue, they encounter changes in acoustic impedance. Regions of low acoustic impedance allow pulses to pass with minimum reflection and will correspond to low intensity regions in the B-mode image. On the other hand, regions of high acoustic impedances will cause a high amplitude signal to be reflected. Such regions like fat, bones and surgical instruments are said to be highly echogenic. Thus, a needle will create an echogenic boundary within tissue, leading to a high intensity feature in the B-mode image.

In addition to the difference in acoustic impedance of the two media forming a boundary which determines the intensity of echoes, we are interested in whether the echoes will hit the transducer to enable detection. The latter consideration is governed by the angle of incidence of the US beam. Strong echoes are collected by the transducer when the needle surface and the pulses are orthogonal. In this case, the angle of incidence is zero. It follows that the flatter the insertion angle with respect to the US transducer is, the better the reflection. Figure 1-4 illustrates the relationship between needle insertion angle and the US transducer. Insertion angle is the angle between the needle shaft and the skin surface at the point of needle entry. If the target is deep, the needle may have to be inserted at a steeper angle, in which case the angle of incidence increases. In this scenario, there is a likelihood of non-axial specular reflection, where some of the reflections are not collected by the transducer.

Since the US beam travels through multiple tissue regions before hitting the needle interface, we expect that it will have previously undergone reflections, absorption and refractions. Indeed, the backscattered waves again undergo reflections before they reach the transducer. Attenuation due to reflection, absorption and refraction is depth dependent. As a result, the intensity of the waves that reach the transducer decreases as depth of the object of interest increases. Scattering of the US beam, when it encounters boundaries with smaller dimensions than its wavelength is yet another effect that reduces intensity of the US beam and causes speckle. There is also a possibility of soft tissue interfaces with the same echogenicity as the needle interfering with needle visualization.

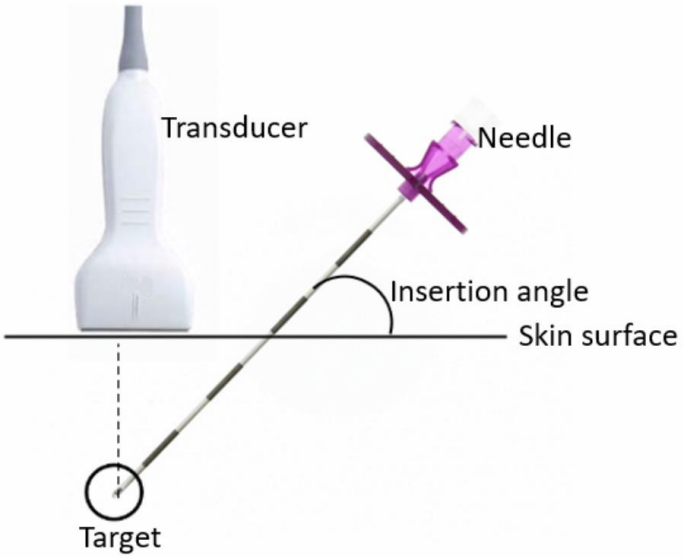


Figure 1-4. Relationship between the US transducer, needle and target during minimally invasive procedures. A shallow insertion angle is possible when the target is superficial, but deeper targets require steep insertion angles which reduce orthogonality between the US beam and the needle surface, hence impeding needle visualization.

Figure 1-5 shows non-axial specular reflection of US waves sent out by a linear US transducer. Note that because of the relatively steep insertion angle, most of the reflected US energy may not be received by the transducer. Figure 1-6 illustrates typical 2D B-mode scans collected using a curvilinear transducer in an *ex vivo* experiment involving animal tissue. Needle shaft and tip visibility is dependent on the insertion angle and depth. Steeper insertion angles and larger depths lead to loss of shaft and tip visibility, and this impedes efficiency of procedures. In order to improve the visibility of the needle and provide automatic localization of the needle tip, various method have been proposed. These will be explained in the next section.

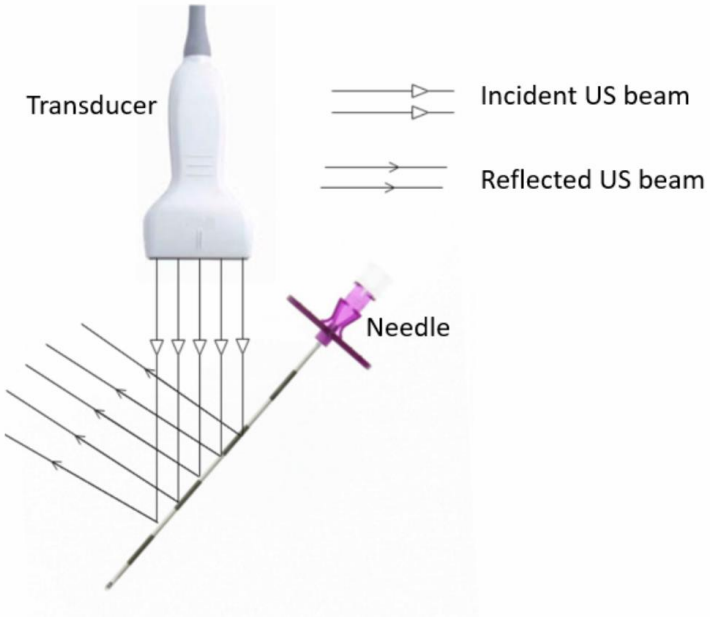


Figure 1-5. Illustrating nonaxial specular reflection from needle surfaces. At each point of incidence on the needle, the angle of incidence is equal to the angle of reflection. However, most of the reflected signal from the needle surface is not received by the transducer, because of the steep insertion angle.

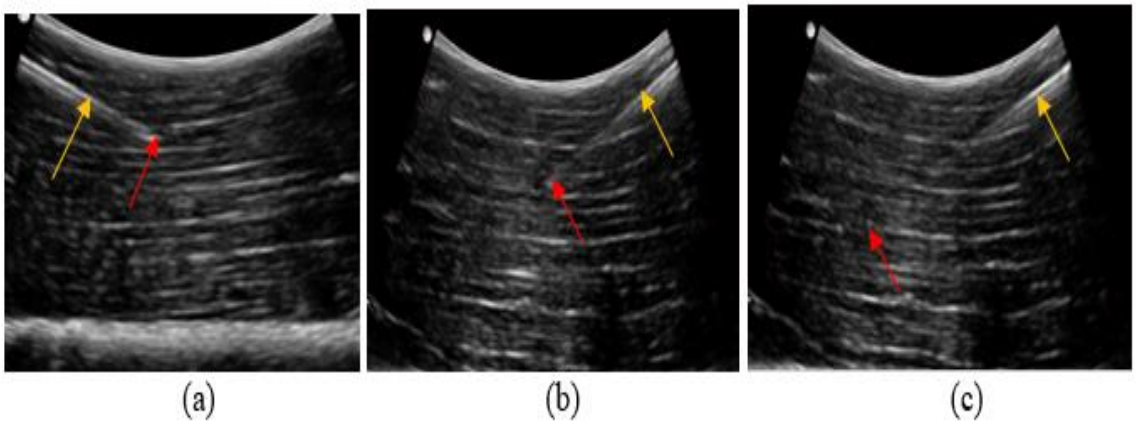


Figure 1-6. Effect of insertion angle on needle visibility. The images were acquired with a 2D C2-5/60 curvilinear transducer on bovine tissue. The red arrow points to the tip and the yellow arrow points to the shaft. (a) The shaft and tip are visible. This is the ideal scenario in a clinical setting and typically occurs at low depth and small insertion angles. (b) The needle tip is visible, but the shaft is discontinuous as we go deeper into the tissue. (c) At even deeper depths, the shaft is discontinuous, and the tip has disappeared. In such a scenario, the surgeon would have to jiggle or rotate the needle to restore visibility, but even then, visibility is not guaranteed. The situation is worse at steeper insertion angles.

1.5 Technological advances to improve needle localization in ultrasound

There is wealth of literature on methods for improving needle visualization/localization in US. These approaches can be broadly grouped into two categories: Hardware enhancements which involve modification or incorporation of additional hardware in the US system, and software improvements in the RF signal or image processing cycle to enhance the needle.

1.5.1 Ultrasound system hardware improvements

Mechanical needle guides: Detachable mechanical needle guides can be attached to transducers to constrain the needle shaft to the path of the US beam [16,17]. However, needle guides limit the degrees of freedom for the needle and

transducer and are therefore not suitable if intraprocedural fine adjustments in the needle trajectory might be required, for example due to patient movement. Furthermore, most needle guides are disposable, and thus increase procedure cost.

Tracking systems: External optical and electromagnetic tracking systems have been reported [18-22]. Optical tracking systems require that line of sight is maintained between the needle (target) and the tracking system. Line of sight is not a limitation in electromagnetic tracking systems. Examples of commercial US systems with electromagnetic tracking include SonixGPS (Analogic, Peabody, Massachusetts, USA), AIM (InnerOptic, Hillsborough, North Carolina, USA), AxoTrack (Soma Access Systems, Greenville, South Carolina, USA), TrigGuide (TrigMedical, Nesher, Israel) and PercuNav (Philips, Andover, Massachusetts, USA). Although these systems ease needle trajectory planning and continuously display the needle tip, they require specialized needles, transducers and electromagnetic transmitters. These hardware enhancements substantially increase the overall system cost. Additionally, electromagnetic transmitters are usually bulky, and needle localization accuracy is affected by other metal objects in the operating environment.

Optical tracking is commercially available on the Clear Guide ONE (Clear Guide Medical, Baltimore, Maryland, USA). The system includes stereo imaging camera-based guidance system attached on the US transducer. When standard needles are used, the system displays a prediction of the needle trajectory but doesn’t enhance the needle visualization. Visualization is only enhanced if special

PercepTIP needles are used. Some optical techniques involve embedding optical fibers in the needle to offer spectroscopic contrast in tissue under US [23-24]. Generally, optical tracking systems also necessitate additional equipment, require specialized skill to operate them and bring additional cost to the US system.

Needle tip design: Technical changes to the tip have been made to facilitate enhancement in the US image. For example, a piezo crystal at the tip has been used to induce vibration, which is then detected using color Doppler US [25]. Other approaches have explored positioning sensors at the needle tip and calculating their position relative to the transducer from the delay in receiving US signals. Sensors made from copolymers, piezoelectric crystals and a fiber-optic hydrophone have been used for this purpose [26-28]. However, the specialized needles and probes are costly.

Echogenic needles: Other researchers have developed echogenic needles which physically enhance the reflection of the US beam off the needle surface [29-34]. Echogenic needles have a special coat or contain material that improves the angular range of the backscattered US beam, for example, a polymer coating that traps microbubbles (Echo-Coat, STS Biopolymers, Henrietta, NY), and a dimpled distal shaft (Echotip, Cook, Bloomington, IN). Some, such as the EchoStim® echogenic insulated US needles (Hakko™ Medical Co. LTD Japan) have been adopted in clinical practice. However, their effectiveness in enhancing the needle tip and shaft is dependent on the angle of inclination of the needle to the US beam. Overall, specialized needles are only a small fraction of the needles in clinical use.

Robotic guidance: There is ongoing work towards investigating the efficacy of robotically guided needle steering and tracking [35-39]. Robotic systems facilitate autonomous or semi-autonomous needle guidance, while utilizing tactile and image assisted feedback from the needle tip. Studies have demonstrated that robots can achieve sub-millimeter accuracy while reducing the degree of invasiveness compared to manual insertions. However, robotic systems generally focus on shallow needle insertions. Perhaps their biggest impediment is cost; robotic systems are expensive and require additional technical expertise to operate and maintain. Therefore, robotic systems have not gained traction in practice.

3D imaging: 2D US renders tools and anatomy in 2D space, yet the actual imaging scene is in 3D. Therefore, the radiologist must mentally construct the 3D scene to properly advance the needle through anatomy. Further, 2D US has a narrow FOV which makes it difficult to align the needle with the imaging plane. In fact, slight motion from the optimum plane usually leads to partial or complete loss of needle visibility.

3D US solves this challenge. Unlike 2D US, 3D US provides image depth information. 3D US permits a wider volumetric FOV without probe adjustment, hence orientation of the needle with respect to the scan plane need not be perfect. This reconstructed volume, when rendered, provides good context for needle movement with respect to patient anatomy. Each volume consists of a spatial sequence of 2D US planes, which can be visualized from any plane. Some US systems provide multi-planar displays, facilitating viewing of the same scene from

different angles [40,41]. With this, radiologists can view arbitrary planes of the reconstructed volume and a panoramic view of the ROI [42].

Traditionally, 3D US consists of 3 stages: acquisition, reconstruction, and visualization. In acquisition, 2D frames are acquired using conventional 2D probes or 3D images are acquired with specialized 3D transducers. During reconstruction, the images are inserted into a volumetric grid to create a voxel array. Visualization entails rendering the built voxel array, for example by arbitrary plane slicing, surface rendering, or volume rendering [43]. This sequential and protracted process makes real-time 3D (4D US) difficult to achieve.

One approach for achieving real-time 3D US is by using a mechanically steered 3D transducer which utilizes a motorized 2D transducer assembly. The 2D transducer is motored to rotate, tilt, or translate over the ROI, and the acquired images are reconstructed into a 3D volume [44,45]. Therefore, the 3D probe must be held statically throughout the acquisition process. Matrix array transducers on the other hand facilitate electronic beam steering which allows direct acquisition of volumetric scans [46,47]. In so doing, they achieve higher frame rates than their motorized counterparts.

Despite these advances in 3D/4D US, the high cost of 3D transducers, bulkiness of transducers, low resolution and low framerate [48] have so far limited widespread adoption of 3D imaging in clinical practice. In addition, *c*ompared to the 3D volume, the needle occupies a small dimension and may be hard to visualize in context. Moreover, 3D US is also affected by signal attenuation, high intensity artifacts and speckle noise.

1.5.2 Software-based methods

Unlike hardware-based methods, software-based approaches are usually implemented to either improve image quality through altering the image formation process or enhance/localize the needle in the final image while maintaining the use of conventional needles and transducers. In the former category, the following examples are found in literature: Spatial compound imaging, in which multiple images acquired from different angles in the same plane are combined to yield a single image has been found to improve needle visibility [49,50]. This technique requires electronic steering of the US beam. Relatedly, with beam steering, the US beam can be guided to be orthogonal to the needle, and this has shown good results in enhancing needle visibility [51,52]. However, its effectiveness is limited to a small range of insertion angles. Further, beam steering is not supported on most US hardware. Some other scholars have proposed the use of power doppler US to localize the needle tip [53,54], but this requires switching between B-mode and color doppler imaging modes, which is cumbersome.

Image processing methods focus on enhancing, segmenting and localizing needles utilizing information in 2D/3D US frames. These can be grouped in three categories: intensity-based methods, intensity invariant methods and learning based methods.

Intensity-based methods: Early works in needle enhancement and localization focused on the characteristic that needles are often present as high contrast linear features in US. Consequently, methods based on the Hough transform (HT) gained prominence [55-60]. Similarly, methods based on the radon transform (RT) have

been reported [61-62]. Other approaches include: the use of orthogonal 2D image projections [63], Maximum Intensity Projection (MIP) [64], and the Random Sample Consensus (RANSAC) [65-67].

Although the above-mentioned methods may differ in design and implementation strategies, they all rely on the assumption that the needle core will appear as a high intensity feature in the US image. This is rarely the case due to the previously mentioned limitations of US: non-axial specular reflection, attenuation and shadowing, speckle noise and hyperechoic interfaces. In fact, in practice, the shaft usually appears discontinuous and doesn't display a conspicuous high intensity feature, and the needle tip may have very low contrast. In such scenarios, all the above-mentioned methods will be inefficient.

Intensity-independent methods: Recently, methods that overcome low needle contrast have been reported. For instance, one can extract local phase information from US images using orientation tuned Log-Gabor filters and use the resulting image to localize the needle tip [68]. Although good tip localization results were reported, this method assumes that part of the needle shaft is visible, and the tip appears as a characteristic high intensity. Localization of needles from spectral analysis of tremor arising from hand-held needle insertion has also demonstrated good performance albeit with computational complexity [69].

Learning-based methods: Instead of handcrafting needle features from US using custom filters, there is a growing trend towards using machine learning for enhancing and localizing needles in US. Pioneer work in this area utilized traditional machine learning approaches, for instance with statistical boosting [70],

and spatiotemporal motion descriptors [71] to train a pixel-wise classifier. More recently, deep learning (DL) based on convolutional neural networks (CNNs) has demonstrated good performance when used for needle segmentation in US [72,73]. Deep learning facilitates segmentation of needle features from an end-to-end learned model; the model learns hierarchical abstractions from image data and uses this information to perform needle extraction, more accurately and faster than traditional approaches. Nevertheless, although deep learning-based approaches have gained traction in other areas of medical image processing [74], there is a paucity of work related to enhancement and localization of needles in challenging imaging scenarios where shaft and tip intensity are low. Moreover, none of the methods reported hitherto provides a real-time solution.

From this review of state-of-the-art, and despite significant advances in US technology and computer vision, there is an unmet need of a clinically usable platform that facilitates real-time enhancement and localization of needles in challenging sonographically guided procedures without altering clinical workflow and increasing hardware cost and complexity.

1.6 Hypothesis and dissertation objective

The goal of this dissertation is to develop robust, accurate, real-time and automatic image enhancement and localization methods for needle shaft and tip in US guided interventional procedures. *The central hypothesis is that modeling US signal transmission coupled with advanced reconstruction methods and learning-based approaches data will improve appearance of the needle in 2D and 3D US, leading to improved guidance for minimally invasive procedures.* The

rationale for this work is that such methods would be integrated in a computational US imaging platform that facilitates effective localization of needles during minimally invasive procedures thus: 1) improving efficacy and efficiency of procedures such as biopsies, peripheral vascular interventions and regional anesthesia, and 2) optimizing patient safety by minimizing the risk of injury. This would be achieved without substantially increasing US system cost or altering the standard clinical workflow.

1.7 Contributions

In this dissertation, we address the challenge of ultrasonic needle visibility while using steeply inserted hand-held needles. First, we demonstrate that when the shaft is partially visible, and the tip doesn't have a characteristic high intensity, needles can be localized by modeling US signal transmission in 2D US images. This approach, though efficient is semi-automatic. We then demonstrate the application of deep learning for fully automated needle detection and localization in 2D US. We also show that nearly imperceptible motion of the needle can be utilized to localize both in-plane and out-of-plane inserted needles in 2D US at a real-time processing speed. We proceed to propose a technique for faster needle localization in 2D US based on keypoint regression. As a final contribution, we show that enhancement and localization of needles from 3D US is possible using novel deep learning methods. The proposed methods are evaluated through *ex vivo* imaging studies on realistic phantom experiments, and demonstrate superior performance to prior art. The developed image processing-based methods, once translated, will be applicable to commercially available 2D/3D cart-based and

portable US systems thus benefitting clinical practice, research and industry. In the course of achieving our research objectives, we have made the following contributions:

1. We introduce a semi-automatic and accurate method for enhancing steeply in-plane inserted needles based on modeling transmission of the 2D US signal. The derived transmission map is integrated in an image restoration model for enhancement of the needle shaft and tip while considering US specific signal propagation constraints. Needle localization is achieved by extracting intensity-invariant local phase needle features from the enhanced image using Log-Gabor filters.
2. We present a fully automatic deep learning framework for detection of steeply in-plane inserted low contrast needles in 2D US. The detection results are utilized for automatic determination of the needle trajectory and subsequently, tip localization from trajectory constrained features.
3. We present a novel tip localization technique for both in-plane and out-of-plane inserted needles based on logical differencing of consecutive frames to detect subtle spatiotemporal variations in intensity. From the difference image, the tip is augmented by solving a spatial total variation regularization problem using the Split Bregman method. This is followed by filtering irrelevant motion events with a deep learning-based end-to-end data-driven method that models the appearance of the needle tip in US images.

4. We evaluate the feasibility of using deep learning for single-shot tip localization in 2D US using a novel learning framework which performs single shot classification of the tip and regression onto the tip location at real-time speed.
5. We present learning-based approaches for needle enhancement and localization in 3D US. The original volume is shrunk to only slices containing needle information using a deep learning framework. Localization of the needle in 3D space is achieved using MIP and local phase features.

The techniques we have developed follow increasing complexity of needle appearance in the US image and contribute to a comprehensive framework for needle localization in 2D/3D US during challenging procedures.

1.8 Outline of the dissertation

The outline of this dissertation is illustrated in Figure 1-7. The rest of the dissertation is subdivided into six chapters outlined below:

Chapter 2: Modeling signal transmission in 2D ultrasound

This chapter describes our work towards achieving enhancement and localization of steeply in-plane inserted needles in 2D US by modeling US signal transmission maps that depict the signal attenuation probability in the image. The maps are then utilized in an image restoration model to recover needle shaft and tip information by proportionately compensating for signal losses. This approach is semi-automatic. The method is qualitatively and quantitatively validated using scans

collected in *ex vivo* experiments on animal tissue phantoms using different needle types and imaging conditions. We evaluate the efficiency and limitations of this approach compared to state-of-the-art.

Chapter 3: Detection and localization of needles in 2D US

This chapter describes a fully automatic deep learning approach utilizing CNNs, for enhancing and localizing steeply in-plane inserted needles in 2D US. The proposed framework achieves accurate needle detection in 2D US images collected in ex vivo experiments. The detection output is used for needle trajectory estimation from intensity invariant features, and thereafter, the needle tip is localized from an intensity search along the needle trajectory. The method achieves accurate needle localization at a faster speed than the framework in Chapter 2. We present quantitative and qualitative results for needle detection, shaft and tip localization and computational efficiency.

Chapter 4: Needle localization from digital subtraction in 2D ultrasound

Here, we present an approach for continuous imperceptible needle tip localization from dynamic (time series) 2D US, for both in-plane and out-of-plane inserted needles. The method combines a novel digital subtraction algorithm for enhancement of low-level intensity changes caused by tip movement in the US image, and a state-of-the-art deep learning scheme for tip detection.

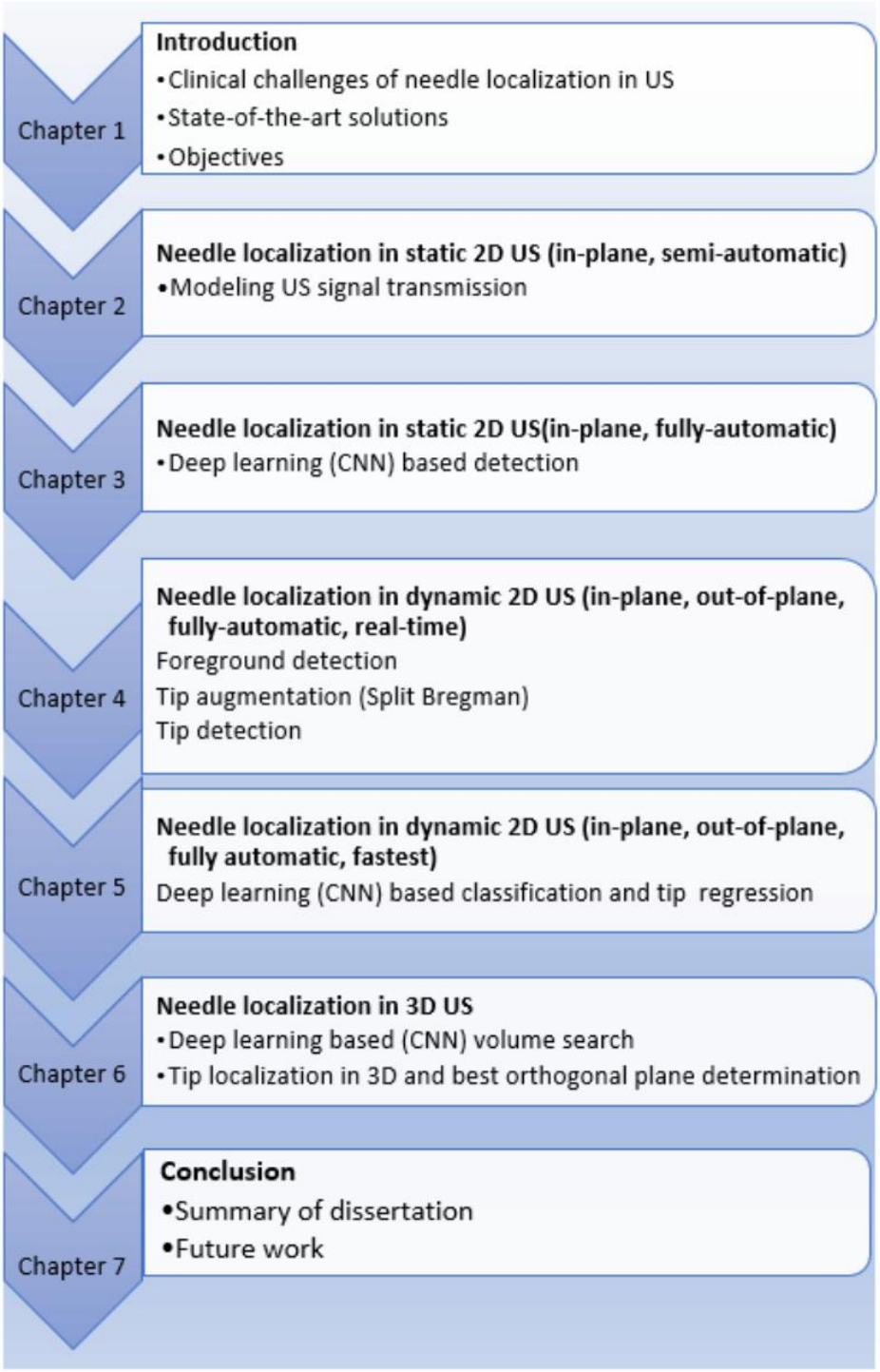


Figure 1-7. Outline of dissertation

We present evaluation results of the method on an extensive ex vivo dataset, collected by inserting different needles in different realistic phantoms. We demonstrate that this is a much faster (10 fps) approach for localizing both in-plane and out-of-plane needles, discuss limitations and opportunities for improving its efficiency.

Chapter 5: Single shot needle detection in 2D US

This chapter presents a real-time approach for needle tip localization in dynamic 2D US. We use a digital subtraction algorithm like the one in Chapter 4 to enhance the needle tip. The enhanced tip is then fed into a novel DL model which performs both classification (determines whether a needle tip is present) and regression to automatically output the spatial tip location. Our approach achieves accurate results and at 67 fps, it is the fastest method for needle tip localization reported so far.

Chapter 6: Learning based needle localization in 3D US

Unlike 2D US which is limited to scan plane visualization and thus only a cross-section of the needle, 3D US produces volume data, facilitating needle visualization at all needle axis orientations. As such, 3D US has the potential to overcome the challenge of needle misalignment by integrating volume data along the needle path.

Techniques in the previous chapters have all focused on 2D US. Since 3D US is essentially a reconstruction of 2D US over a wider FOV, some of these methods can be extended to 3D US. In this chapter, we describe a learning-based

framework for needle enhancement and localization in 3D US to demonstrate this feasibility. Slices (2D frames) containing the needle are extracted from the volume using a DL CNN detection model. The needle is enhanced from the resulting sub-volume using MIP. Intensity invariant needle features are then extracted from the reduced volume using Log-Gabor filters, and used to localize the needle tip in 3D, resulting in localization of the orthogonal slice that provides the best needle visualization. The performance of this method is evaluated on 3D US data collected in ex vivo experiments.

Chapter 7: Conclusion and future work

This chapter summarizes the results and conclusions from this dissertation. We discuss improvements that can be made to our approaches, and how they can be integrated in a computational imaging platform for robust and accurate localization in both 2D and 3D US. We also discuss the potential to extend our work to localizing other surgical instruments such as guidewires and catheters, as well as other imaging systems and modalities such as transrectal US (TRUS), Transesophageal echo (TEE) US, fluoroscopy and endoscopy.

Chapter 2 -Modeling signal transmission in 2D ultrasound

2.1 Overview

Image quality in 2D US is adversely affected by signal attenuation arising from reflections, refraction, scattering and absorption of the US signal. In this chapter, we propose a framework for enhancement and localization of the needle shaft and tip in steeply inserted hand-held needles under in-plane 2D US guidance where signal attenuation is prevalent. This is achieved by modeling the transmission of the US signal and deriving maps that depict the attenuation probability within the image domain. The maps are employed in a contextual regularization framework to recover needle shaft and tip information. The needle tip is automatically localized by line-fitting along the local-phase directed trajectory, followed by statistical optimization. The proposed method was tested on *ex vivo* US scans collected during insertion of various needles into freshly excised porcine and bovine tissue. A tip localization accuracy of 0.55 ± 0.06 mm was achieved at a processing rate of 1.77 s for a 500×500 image.

Our main contribution here is the incorporation of US signal modeling into an optimization problem to estimate an unknown signal transmission map which is used for enhancement of the needle shaft and tip while considering US specific signal propagation constraints. We also address the challenge of high intensity artifacts that may reduce needle localization accuracy. The results provide

evidence of the method's clinical potential in improving needle identification during challenging US guided procedures, such as lumbar blocks in obese patients.

2.2 Background

The need to minimize the effect of signal attenuation in US is an ongoing research problem in medical image computing. Algorithms for US image processing always face the need to overcome low contrast and loss of anatomical features in attenuated or shadowed regions [75]. One approach for overcoming this problem is through modeling of US signal transmission. Confidence maps, a measure of per-pixel confidence, have previously been proposed to estimate uncertainty in US images caused by attenuation and shadowing [76]. Confidence maps are performed within a random walks' framework [77]. They have previously been efficiently used in image segmentation, registration and shadow detection [76,78] as well as bone segmentation [79,80]. In these approaches, confidence maps are incorporated in context specific models cognizant of the features of interest. For example, use of local Phase Symmetry (PS) features is popular for segmentation of bone surfaces since bone surfaces exhibit ridge-like responses in US [81].

Confidence maps can also be utilized in image regularization problems. Image degradation effects such as attenuation, shadowing and noise present us with the need for image restoration in order to obtain an enhanced image. Let us consider a generic case in which the observed 2D US image $US(x, y)$ has been

affected by a gain (or loss) field with a response $A(x, y)$ in the medium, and noise $\mathrm{Z}(x, y)$. We can write:

$$US(x, y) = A(x, y) * US_E(x, y) + Z(x, y) \quad (2.1)$$

Here, the image restoration problem is presented with the observed image being a two-dimensional convolution of the true image intensity $US_E(x, y)$ with the blurring function $A(x, y)$ plus the noise [82]. In discrete terms, (2.1) can be expressed as:

$$\mathbf{US} = \mathbf{A}\,\mathbf{US_E} + \mathbf{Z} \quad (2.2)$$

In (2.2), $\mathbf{US}$, $\mathbf{A}, \mathbf{US_E}$ and $\mathbf{Z}$ are vectors containing all the pixel values in the respective image matrices in lexicographic order. The problem in (2.2) is ill-conditioned, and solving it directly is difficult due to noise sensitivity. This is when regularization methods are called into play. Regularization is achieved using norm functions which assign non-negative lengths or sizes to each vector in a vector space.

For a vector $\boldsymbol{m} \in \mathbb{R}^n$, we are interested in 1-norm (L_1) and the 2-norm (L_2). L_1 seeks to minimize the sum of distances along coordinate axes i.e. the sum of the absolute differences between the target value and the estimated value: $\|\boldsymbol{m}\|_1 = \sum_{k=1}^{n} |\boldsymbol{m}_k|$. L_2 on the other hand, is the Euclidean length or least squares, and seeks to minimize the sum of the square of the differences between the target value and the estimated value: $\|\boldsymbol{m}\|_2 = (\sum_{k=1}^{n} |\boldsymbol{m}_k|^2)^{\frac{1}{2}}$.

For our problem in (2.2), the linear least squares regularization problem formulated according to the method proposed by Tikhonov [83] would take the form:

$$\|\mathbf{US} - \mathbf{A}\,\mathbf{US_E})\|_2^2 + \lambda\|L\,\mathbf{US})\|_2^2 \quad (2.3)$$

The first term in (2.3) is the data-fitting or error term, λ is the regularization parameter, L is the regularization operator, and the second term is the penalty term which accounts for irregularity of the solution. *L* is often chosen to be a first or second order finite difference operator. Depending on the nature of the regularization problem, (2.3) can be modified. Previously, L_1 has found preferable usage in image restoration because it yields more robust solutions than L_2 [84-88]. When we expect features that have edges in our optimization data and hence possible outliers in the second term of (2.3), the use of L_1 for this term is prudent. (2.3) is thus modified to take the form:

$$\|\mathbf{US} - \mathbf{A}\,\mathbf{US_E})\|_2^2 + \lambda\|L\,\mathbf{US}\|_1 \quad (2.4)$$

(2.4) presents an optimization problem which combines L_1 and L_2, and is known as a Least Mixed Norm (LMN) problem [82]. The work in this chapter will focus on modelling an LMN contextual regularization framework for needle tip and shaft enhancement, in which the effects of attenuation are modelled using US confidence maps and Euclidean Distance (ED) maps.

2.3 Methods

The proposed method is based on our prior experience with in-plane US guidance of lumbar injections in vivo, in which (i) the needle is inserted in-plane and the insertion side (left or right) is known, (ii) the needle tip appears as a characteristic but variable intensity, and (iii) a portion of the needle shaft near the transducer surface is visible. In our approach, we focus on enhancement of

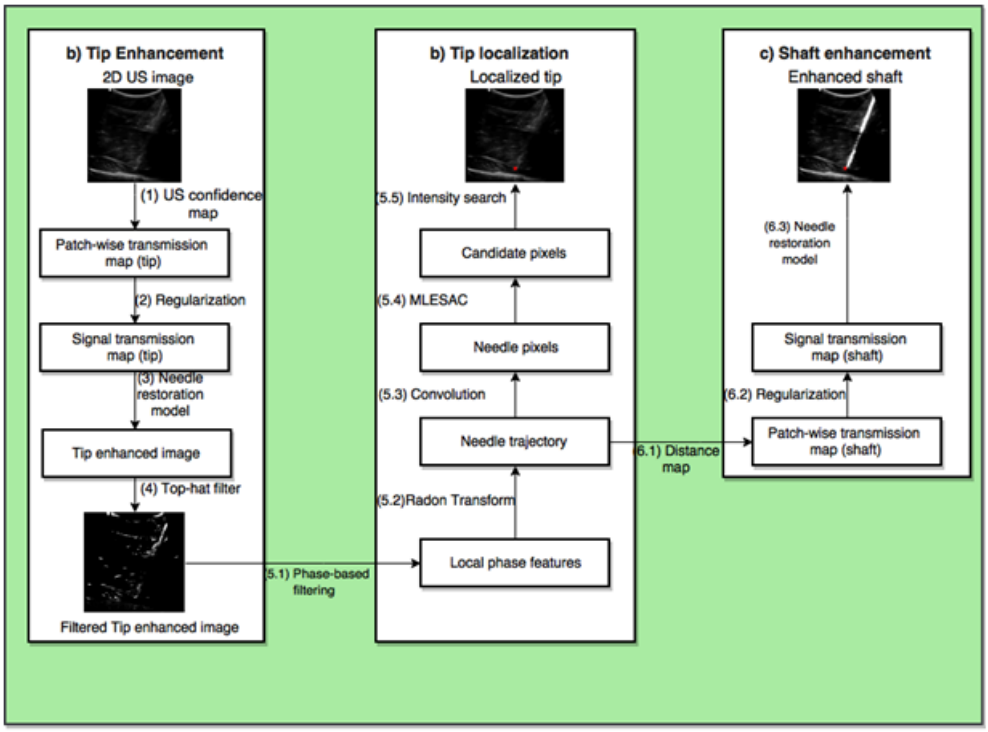
needles under curvilinear 2D US guidance at insertion angles of $40^0 - 80^0$ to the skin surface. Owing to the transducer's convex shape, only part of the shaft is present in the US image. An overview of the proposed is shown in Figure 2-1. In the next section, we describe how these image features are used in our proposed method.

2.3.1 Needle restoration model

Consider a 2D B-mode US image $I(x, y)$ in which the needle tip and shaft are inconspicuous. We propose a linear interpolation model to account for the effect of attenuation:

$$I(x, y) = t(x, y)I_e(x, y) + \big(1 - t(x, y)\big)v, \tag{2.5}$$

where $t(x, y)$ is the depth dependent signal transmission map function (representing response of a loss field in the transmission medium), $I_e(x, y)$ is the US image intensity to be recovered and v is a constant intensity equal to echogenicity of the tissue confining the needle. If ρ is the arc-length parametrization of the signal path, then, for each short distance $d\rho$ along this path, the fraction of US signal lost is given by $\propto \times d\rho$, where $\propto$ is the attenuation coefficient. A spatially varying $\propto$ yields $t(x, y) = exp\left(-\int_0^{d(x,y)} \propto \big(\rho(\tau)\big)d\tau\right)$. Therefore, $t(x, y)$: $0 \leq t(x, y) \leq 1$ denotes the fraction of US signal that reaches a target point $d(x, y)$ without attenuation.



Input: US image $I(x,y)$
Output: Tip location (x_t, t_t), Enhanced shaft image $I_e(x,y)_{shaft}$

(1.1) $G(v,e) \leftarrow I(x,y)$ ▷ $G(v,e)$ is an 8-connected graph
(1.2) $W_h, W_v, W_d \leftarrow e_{ij}$ ▷ e_{ij} are edges, W_h, W_v, W_d are edge dependent weights
(1.3) $I_c(x,y) \leftarrow W$ ▷ confidence map $I_c(x,y)$ from graph weights
(1.4) $\psi(x,y)_{tip} \leftarrow I_c(x,y)$ ▷ ψ is patch-wise transmission function
(2.1) $W_\rho \leftarrow I(x,y)$ ▷ W_ρ are contextural weights
(2.2) $t(x,y)_{tip} \leftarrow \psi(x,y)_{tip}$ ▷ regularization, $t(x,y)_{tip}$ is signal transmission map (tip)
(3.1) $\nu \leftarrow I(x,y)$ ▷ ν is echogenicity of tissue confining the needle
(3.2) $I_e(x,y) \leftarrow I(x,y)$ ▷ needle restoration model for tip enhancement
(4) $I_e(x,y) - (I_e(x,y) \ominus L) \oplus L \leftarrow I_e(x,y)$ ▷ Top-hat filter, L is a linear element
(5.1) $I(\theta) \leftarrow I_e(x,y)_{tip}$ ▷ Local phase-based filtering
(5.2) $l_{\theta+\theta_0} \leftarrow I(\theta)$ ▷ l_θ is a linear trajectory mask
(5.3) $I(\theta,l) \leftarrow l_{\theta+\theta_0}$ ▷ pixels in $I_e(x,y)_{tip}$ lying along trajectory
(5.4) $I(\theta,l) - I(\theta_0,l_0) \leftarrow I(\theta,l)$ ▷ remove outliers to leave only candidate pixels
(5.5) $(x_t,t_t) \leftarrow$ *candidate pixels* ▷ geometrical optimization and intensity search
(6.1) $\psi(x,y)_{shaft} \leftarrow l_{\theta+\theta_0}$ ▷ patch-wise transmission map for shaft
(6.2) $t(x,y)_{shaft} \leftarrow \psi(x,y)_{shaft}$ ▷ regularization
(6.6) $I_e(x,y)_{shaft} \leftarrow t(x,y)_{shaft}$ ▷ needle restoration model for shaft enhancement

Figure 2-1. Overview of the proposed framework for needle enhancement and localization. It consists of three main processes: (1-4) tip enhancement using a linear interpolation model that utilizes US signal transmission maps, (5) local phase-based tip localization, and (6) shaft enhancement. Inset is the pseudocode, whose parameters are described in detail in the text that follows.

Referring to (2.5), the enhanced image is obtained using:

$$I_e\,(x,y) = \frac{I(x,y)-\nu}{[max\,(t(x,y),\ \kappa)]^{\propto}} + \nu, \tag{2.6}$$

where $\kappa = 0.001$, a small constant prevents division by zero. The proposed solution is severely under constrained because the number of unknowns is greater than the number of equations. The approach we devise to achieve a solution is to first model a patch-wise transmission map, $\psi(x,y)$, that reflects the boundary constraints imposed on the needle by attenuation and the needle trajectory, for all $(x,y) \in \Omega$, the set of pixel locations in $I(x,y)$. $\psi(x,y)$ is then optimized to generate $t(x,y)$ through a regularization framework we will describe later. Next, we describe how we compute different $\psi(x,y)$ and thereafter, $t(x,y)$ for tip and shaft enhancement, and localization of the needle tip.

2.3.2 Patch-wise transmission maps for tip enhancement

Direct derivation of the patch-wise transmission map would involve partitioning the image into grids specified by a patch size, and knowledge of image depth information. This approach would be computationally expensive. Confidence maps have previously been used to estimate uncertainty in US images caused by depth dependent attenuation and shadowing [76]. Leveraging this approach to estimate the patch-wise transmission map automatically includes depth information and ensures that the transmission in a local image patch, at the same depth, will always be almost constant. Therefore, we do not have to specify image patch sizes.

A confidence map, $I_c(x,y)$ results from a probability density function that assigns to each pixel in $I(x,y)$ a probability that a random walk [77] emanating from that pixel would be able to reach virtual transducer elements at the top of the image, given US specific constraints. To calculate the confidence map, $I(x,y)$ is represented as a weighted graph in an 8-connected lattice. Random walks originating from virtual transducers at the top of the image are used to calculate expected signal strengths throughout the image [76]. The weighting function for the random walks is given by:

$$\chi_{ij} = \begin{cases} {\chi_{ij}}^{H} = exp\left(-\beta\left|c_i - c_j\right| + \gamma\right), & e_{ij} \in E_H \\ {\chi_{ij}}^{V} = exp\left(-\beta\left|c_i - c_j\right|\right), & e_{ij} \in E_V \\ {\chi_{ij}}^{D} = exp\left(-\beta\left|c_i - c_j\right| + \sqrt{2}\gamma\right), & e_{ij} \in E_D \\ 0, & otherwise. \end{cases} \tag{2.7}$$

Here, e_{ij} represents the edge between nodes i and j, $E_H, E_V,$ and E_D are the horizontal, vertical and diagonal edges on the graph, $c_i = I(x,y)_i \exp(-\propto l_i)$, where $I(x,y)_i$ is the image intensity at node i, and l_i is the normalized closest distance from the node to the virtual transducer elements. The behavior of the confidence map is controlled by three free parameters: $\propto, \beta$, and γ. $\propto$, the attenuation coefficient controls the depth-dependent attenuation rate. It is expected that pixels close to the top of the image will exhibit high confidence and vice versa. β is an algorithmic constant which affects the robustness and accuracy of segmentation. γ models the beam width, imposing a penalty on random walks crossing a horizontal/diagonal edge in the graph with increasing corresponding distance from the starting scanline. Recalling that $\boldsymbol{I_c}(x,y)$ expresses probability

that a random walk emanating from a pixel in the image reaches the top of the image, yet we are interested in the likelihood that a signal sent from the transducer will reach a point of interest in the image (needle shaft and tip), we accordingly model the patch-wise transmission map as $\psi(x,y) = I_c(x,y)^*$, where * denotes the complement. The parameters of $\psi(x,y)$ are determined as follows: $\propto= 2$, and $\beta = 90$ for all the experiments [76,77]. γ should achieve a balance between a distinct map and minimal horizontal discontinuities. In Figure 2-2, we show the effect of $\gamma \in [0\ 1]$ on the derived $\psi(x,y)$. Note that $\gamma = 0.03$ yields the best balance; this value was determined from several test images and kept constant throughout the validation experiments.

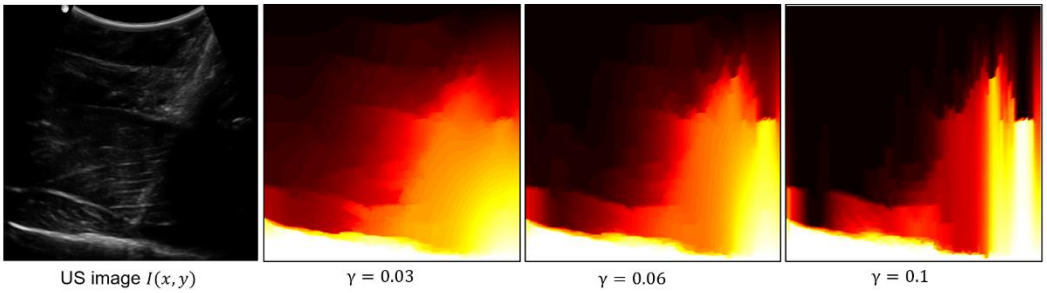


Figure 2-2. Effect of γ on the patch-wise transmission map, $\psi(x,y)$. With $\propto = 2$ and $\beta = 90$ fixed, $\gamma = 0.03$ achieves a distinct function with minimum horizontal discontinuities.

2.3.3 Signal transmission maps for tip enhancement

As mentioned in Section 2.2, to obtain a solution that is less sensitive to noise and abrupt changes in image intensity, we formulate a LMN regularization problem for deriving the signal transmission map, $t(x,y)$. Our objective function is:

$$\frac{\lambda}{2}\|t(x,y) - \psi(x,y)\|_2^2 + \sum_{i\in\Omega}\left\|W_i \circ \left(G_i * t(x,y)\right)\right\|_1 \quad (2.8)$$

This equation has two components: the first is the data which measures the closeness of $t(x,y)$ to $\psi(x,y)$, while the second introduces additional contextual constraints on $t(x,y)$. The regularization parameter, λ, is used to balance the two parts. G_i is a bank of high order differential operators consisting of 8 Kirsch filters [89] and a Laplacian operator. The 8 Kirch filters consist of the same kernel mask rotated in 45° increments. Combining the first-order derivative Kirch filters with a second-order derivative Laplacian mask preserves edge features associated with the needle. In (2.8), we introduce a weighting function W to further constrain $t(x,y)$ in a local image patch. For two neighboring pixels, the weighting function is such that $W(t_2(x,y) - t_1(x,y)) \approx 0$. If the two pixels are far apart, then W should be small, and vice versa. Consequently, we compute the weighting function from: $W_i(\varrho) = exp\,(-\left|(G_i * I(x,y))_{\varrho}\right|^2)$, where ϱ is a given location in the image. When $t_2(x,y) = t_1(x,y)$, $W = 0$. Conversely, when $W = 0$, the constraint on $t(x,y)$ between neighboring pixels is eliminated.

One approach to solving (2.9) involves using variable splitting where several auxiliary variables are introduced to construct a sequence of simple sub-problems, the solutions of which finally converge to the optimal solution of the original problem [90]. In Figure 2-3, we show the result of deriving the signal transmission map $t(x,y)$ from $\psi(x,y)$ using various λ. In all cases, $t(x,y)$ exhibits low intensities at the top, with depth dependent increase. However, we desire $t(x,y)$ with the smoothest attenuation density estimate for use in the needle restoration model. Setting $\lambda = 1$ ensured that the needle shaft and tip were enhanced mutually on several test images, and this value was kept constant throughout the validation

experiments. Knowing $t(x,y)$, the tip enhanced image is determined using (2.6). Notice that the mean intensity of the local region in $t(x,y)$ is always less than the echogenicity of the tissue confining the needle. From (2.6), the tip will be represented by a local average of the surrounding points, yielding a high intensity feature corresponding to the tip in the enhanced image.

Figure 2-4(c) illustrates the result of tip enhancement in a 2D US image where the needle tip doesn't originally have a characteristic high intensity (Figure 2-4(a)). The transmission map $t(x,y)$ is shown in Figure 2-4(b).

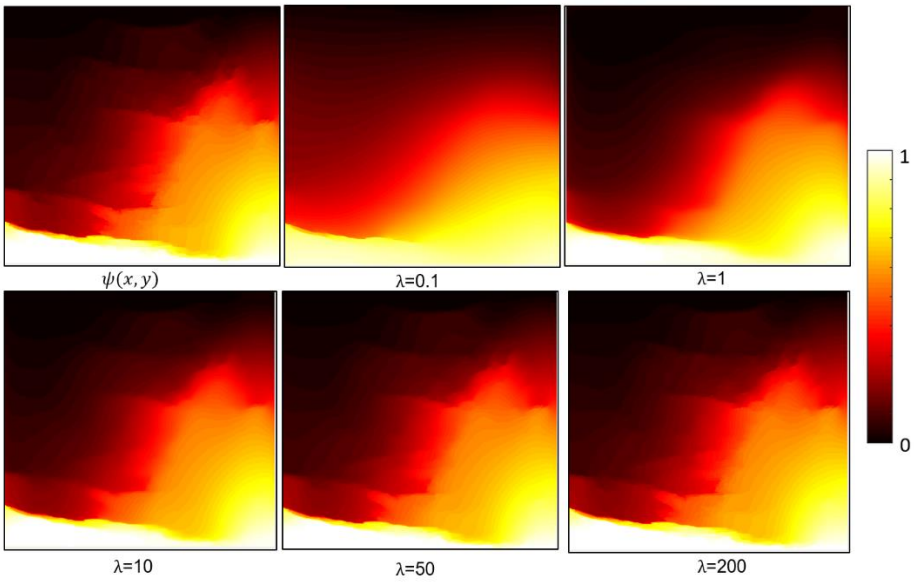


Figure 2-3. Effect of the regularization parameter, λ on derivation of $t(x,y)$ from $\psi(x,y)$. $\psi(x,y)$ is derived with $\propto = 2, \beta = 90$ and $\gamma = 0.03$. $t(x,y)$ displays lower intensities near the top of the image where attenuation and scattering are minimal, and higher intensities with increasing depth. As λ becomes larger, $t(x,y)$ tends to $\psi(x,y)$. λ=1 ensures $t(x,y)$ with the smoothest attenuation density estimate and the best mutual enhancement of the shaft and tip.

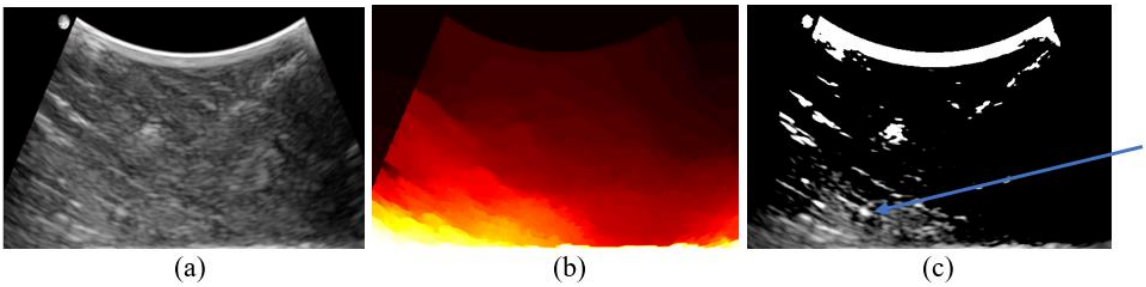


Figure 2-4. Needle tip enhancement process. (a) B-mode US image at an insertion angle of 45^0. The shaft is discontinuous, and the needle tip is not easily visible. (b) The signal transmission map function $t(x, y)$. The map displays lower intensities near the top of the image where attenuation is minimal, and higher intensities with increasing depth. (c) The enhanced needle tip (blue arrow) along the needle trajectory. The tip is conspicuous and easier to localize.

In (2.6), the choice of v affects the accuracy of the enhancement result. It is imperative that v results in partial enhancement of the shaft alongside the tip. As we will see later, shaft information is essential for the tip localization process. If v is brighter than most pixels in $I(x, y)$, the output of (2.6) will produce no needle information. Alternatively, a low v creates more high intensity artifacts. To maximize contrast of the needle, $v = 0.3 \times max\ (I(x, y)$ where $max\ (I(x, y)$ denotes maximum intensity in $I(x, y)$, is chosen.

Minimizing effect of high intensity artifacts: The tip enhancement process may also enhance artifacts along the needle trajectory. With such artifacts in the enhanced image, it is difficult to distinguish needle features. Therefore, we need a mechanism to remove them. When these artifacts appear linearly, it is easy to eliminate them using morphological filters. Specifically, we use a Top-hat filter (TF), which computes the morphological opening of $I_e(x, y)$ and subtracts it from

$I_e(x,y)$: $TF\big(I_e(x,y)\big) = I_e(x,y) - D_L[E_L(I_e(x,y))]$. Here, L is a linear structuring element while D_L and E_L denote dilation and erosion operations respectively. The output of the TF yields the final tip enhanced image, $I_e(x,y)_{tip}$. In Figure 2-5, we show the output of tip enhancement process for various ν. It is observed that in the optimum case, the high intensity artifact at the bottom of the image is suppressed.

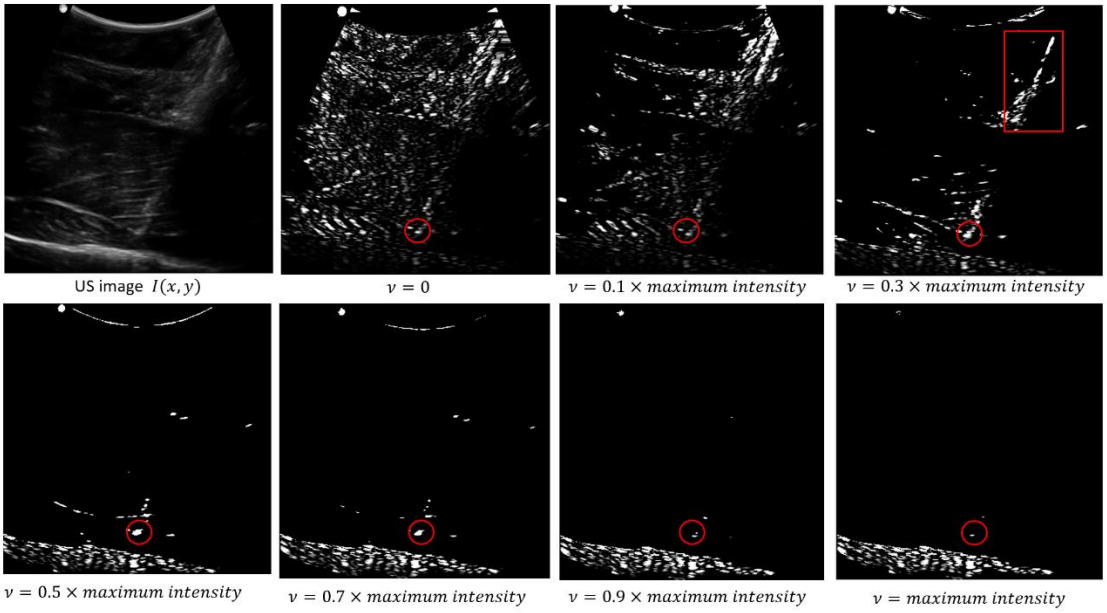


Figure 2-5. Example of needle tip enhancement at various ν. Note that ν influences whether enough tip and shaft information is present in the tip enhanced image, $I_e(x,y)_{tip}$ to facilitate the tip localization process. In the desired case, the tip (enclosed by red circle) is enhanced together with part of the shaft (enclosed by red rectangle).

2.3.4 Tip localization

The first step in the tip localization process is estimation of the needle trajectory. To achieve this, we need to extract shaft information from the enhanced image, $I_e(x,y)_{tip}$. For a 500×500 $I_e(x,y)_{tip}$ image, a fixed 100×100 region of interest (ROI), $I_eROI(x,y)$, is defined on the insertion side of the needle as shown

in Figure 2-6(a). $I_e ROI(x, y)$ must contain part of the shaft, although it need not contain the tip. To extract shaft information from $I_e ROI(x, y)$, we use a framework previously proposed in [68]. This framework demonstrated good localization accuracy when the tip had adequate characteristic high intensity compared to the surrounding tissue, a scenario we achieve after the tip enhancement process.

We begin by enhancing the needle shaft near the entry point into the tissue using a bank of orientation tuned band-pass 2D Log-Gabor filters, with the transfer function:

$$LG(\omega, \theta) = exp\left(\frac{-log\left(\frac{\omega}{k}\right)^2}{2\log {\sigma_\omega}^2}\right) exp\left(\frac{-(\theta-\theta_m)^2}{2{\sigma_\theta}^2}\right) \quad (2.9)$$

Here, ω is the filter frequency, θ is the filter orientation, k the center frequency, σ_ω is the bandwidth on the frequency spectrum, σ_θ is the angular bandwidth and θ_m is the filter orientation. The filter parameters are selected automatically as proposed in the pioneer work [68]. These filters have a Gaussian distribution on the log scale and are excellent for extraction of local-phase features, yielding intensity-invariant image descriptors (which are insensitive to US brightness variations). After applying the filters, the resulting phase symmetry image, $PS(x, y)$, contains distinct local phase features for the needle shaft. Limiting the calculation to $I_e ROI(x, y)$ minimizes the effect of any residual high intensity artifacts that may be present elsewhere in $I_e(x, y)_{tip}$. In Figure 2-7, we show the result of extracting $PS(x, y)$ for scans obtained from different tissue types. Independence of our method from tissue type is augmented by calculating the image from a binary tip enhanced image.

From $PS(x,y)$, the needle trajectory is estimated using the RT with an angular range of $0^{\circ} - 179^{\circ}$. To obtain a region where the needle shaft and tip certainly lie, the estimated trajectory is expanded over the whole $I_e(x,y)_{tip}$ image as shown in Figure 2-6(b). Knowledge of the trajectory region helps us to extract only data lying along the trajectory in $I_e(x,y)_{tip}$ by convolution. This data is trimmed using the Maximum Likelihood Estimation Sample Consensus (MLESAC) algorithm [91], which performs inlier detection and geometrical optimization. The resulting image, $I_{MLESAC}(x,y)$ is shown in Figure 2-6(c).

The resulting co-linear candidate intensities lying along a line L are distributed among a set of line segments, each defined by a set of points or knots denoted as $\mu_1 \ldots\ldots \mu_n$. The needle tip is extracted using:

$$I_{needle}\left(I_{BP}(x,y)\right) = \frac{\int_{\mu_i}^{\mu_i+1} I_{BP}(x,y)\, d\mu}{\left\| L_{(\mu_i+1)} - L_{\mu} \right\|_2}; \mu \in [\mu_i, \mu_{i+1}] \tag{2.10}$$

In (2.10), $I_{BP}(x,y)$ is obtained by applying a Log-Gabor filter without orientation selectivity to $I_e(x,y)_{tip}$, whereas μ_i and μ_{i+1} are successive knots [68]. The function in (2.10) assigns to pixels between knots μ_i and μ_{i+1} a mean intensity value along L. The result of this operation, $I_{needle}(x,y)$ is shown in Figure 2-6(d). From $I_{needle}(x,y)$, the needle tip is localized as the farthest maximum intensity pixel at the distal end of the needle trajectory (Figure 2- 6(e)).

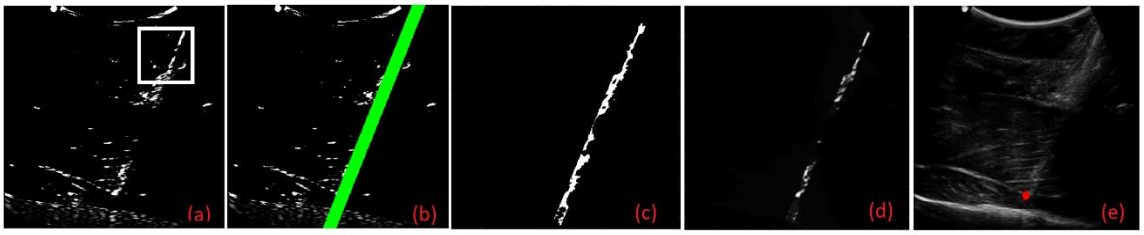


Figure 2-6. Framework for Framework for automatic tip localization. (a) Region of interest $I_eROI(x,y)$ on enhanced tip image $I_e(x,y)_{tip}$, indicated by white rectangle. (b) Estimated trajectory (green) overlayed on $I_e(x,y)_{tip}$. (c) $I_{MLESAC}(x,y)$, the output of the MLESAC algorithm. (d) $I_{needle}(x,y)$ image obtained from (5). From $I_{needle}(x,y)$, needle tip localization is achieved by selecting the first maximum intensity pixel lying along the calculated needle trajectory. (e) Enhanced needle tip (red dot) marked on the US image $I(x,y)$.

2.3.5 Shaft enhancement

The final step in our proposed framework is needle shaft enhancement using (2.6) and (2.8). Here, we desire different patch-wise and signal transmission maps relevant to the shaft restoration problem. Suppose along the needle trajectory, *N* represents a set of pixels belonging to the needle and *B* represents a set of pixels belonging to the background in $I(x,y)$. The patch-wise transmission map $\psi(x,y)_{shaft}$ is derived using the ED transform of all *N*, i.e. a measure of the minimal distance between *N* and *B.* If we denote the ED by $\sigma(x,y)$, then:

$$\psi(x,y)_{shaft} = \frac{|\sigma(x,y) - max\ (\sigma(x,y))|}{max\ (\sigma(x,y))} \tag{2.11}$$

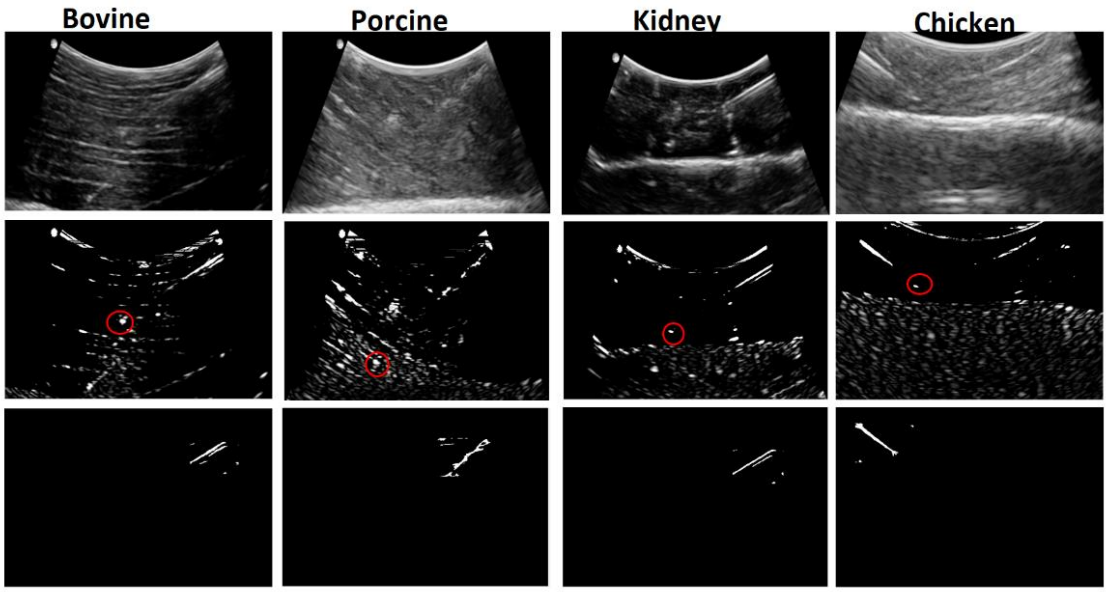


Figure 2-7. Illustrating tissue independence of $PS(x,y)$ image for different tissue types (bovine, porcine, kidney and chicken). Row 1: US image $I(x,y)$. Row 2: Tip enhanced image $I_e(x,y)_{tip}$. The enhanced tip is surrounded by a red circle. Row 3: $PS(x,y)$ image. The use of $PS(x,y)$ in calculating needle trajectory ensures accuracy through orientation tuning and independence from tissue type.

Since we know the tip location from the previous step, we constrain $\psi(x,y)_{shaft}$ not to exceed the tip position. The signal transmission map for the shaft, $t(x,y)_{shaft}$ is obtained from $\psi(x,y)_{shaft}$ using (2.8), and shaft restoration is performed using (2.6), with $v = max\ (I(x,y)$. The result of shaft enhancement is illustrated in Figure 2-8. Notice that $t(x,y)_{shaft}$ has low intensities along the needle axis and higher intensities for image regions away from the axis. The enhanced shaft arises from a local average of pixels belonging to the shaft along the trajectory.

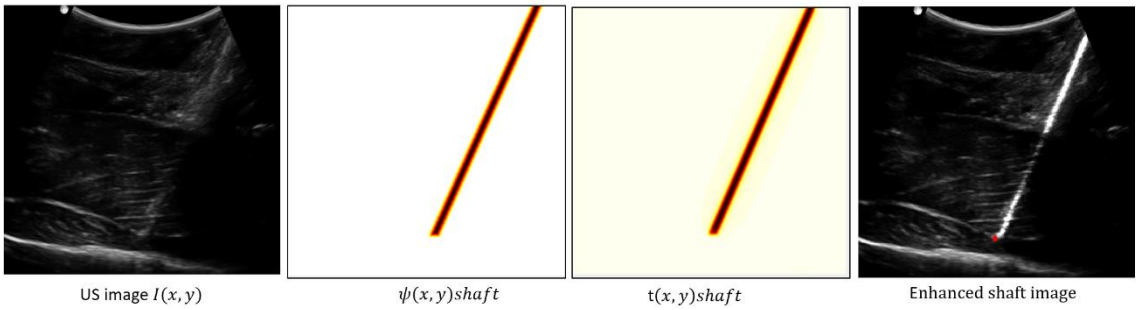


Figure 2-8. Needle shaft enhancement process. The signal transmission map $t(x,y)_{shaft}$ derived from the patch-wise transmission map $\psi(x,y)_{shaft}$, shows higher intensities with increasing distance from the central axis of the estimated trajectory. In the shaft enhanced image, the automatically localized tip is indicated by the red dot.

2.4 Experimental validation

US images for validation were obtained using a SonixTouch US system (Analogic Corporation, Peabody, MA, USA) equipped with a C5-2/60 curvilinear probe. A 17-gauge (1.5 mm diameter, 90 mm length) Tuohy epidural needle (Arrow International, Reading, PA, USA) was inserted by hand into freshly excised bovine and porcine tissue at varying angles ($40^{\circ} - 80^{\circ}$) and depths (up to 9 cm). 300 scans (150 from each tissue type) were collected with the needle in-plane.

We performed an additional set of experiments following a similar procedure but with different needles: 22GA BD Spinal Needle (BD Medical, Franklin Lakes, New Jersey, USA), and the Biopince Full Core Biopsy Needle (Argon Medical devices, Athens, Texas, USA). By using different needle types, we sought to determine the robustness of our method to needle size and length. Figure 2-9 illustrates a typical experimental setup for data collection.

The proposed method was implemented in MATLAB 2017a (Mathworks, Natick, Massachusetts, USA) on a 4.2 GHz Intel(R) CoreTM i7 CPU, 16GB RAM Windows PC (DELL, Round Rock, Texas, USA). After automatically determining the tip localization accuracy, an expert manually localized the tip on corresponding images. We then determined localization error as the ED between the automatically localized tip and the manually localized position.

Figure 2-9. Data collection setup. 2D data is collected using the SonixGPS system and C5-2/60 curvilinear transducer. The needle depth and insertion angle are altered as static image frames are collected. Here, the Biopince Full Core Biopsy Needle (Argon Medical devices, Athens, Texas, USA) is inserted in Porcine tissue.

The Root Mean Square (RMSE) and 95% Confidence Interval (CI) for localization error were also calculated. The same dataset used in evaluating the proposed method was used to assess the performance of a state-of-the-art method [68]. Similarly, the RMSE and the associated 95% CI were calculated. Further, we performed a one tailed paired t-test to determine whether the differences between localization errors from the two methods were statistically significant.

2.5 Results

2.5.1 Qualitative results

Qualitative results for tip and shaft enhancement of the 17G epidural needle (Figure 2-10) show that the proposed method efficiently enhances the tip and shaft at steep insertion angles when the shaft is broken and the tip inconspicuous. Our method works for different visibility profiles of the shaft and tip in the US image. Figure 2-11 shows a qualitative comparison of the proposed method and the methods in [68]. In the top row, we notice that when the tip intensity is high compared to surrounding tissue and there are no high intensity artifacts along the trajectory, both methods give accurate tip localization. However, investigating the middle row shows that low tip intensity affects tip localization by the method in [68]. The last row illustrates a case where high intensity artifacts lead to inaccurate localization by the method in [68]. These artifacts do not affect the proposed method.

Figure 2-12 illustrates qualitative results for tip and shaft enhancement for B-mode images collected with insertion of a full core biopsy needle (Figure 2-12(a-

b)) and a spinal needle (Figure 2-12(c-d)). Again, we notice substantial recovery of needle shaft and tip information in the enhanced images.

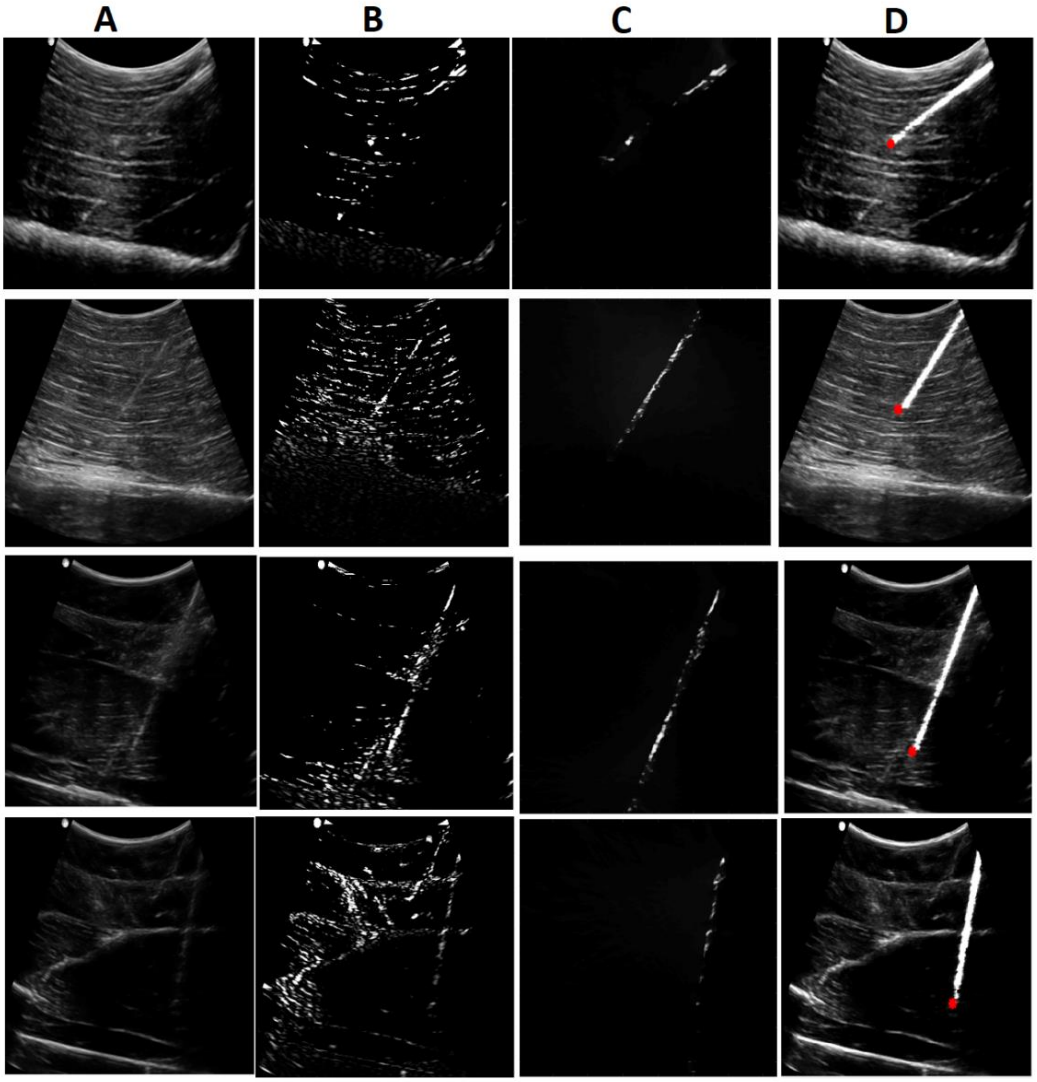


Figure 2-10. Qualitative results showing shaft enhancement and tip localization at steep insertion angles of the 17G Tuohy epidural needle. A) US image $I(x,y)$. B) Tip enhanced image $I_e(x,y)_{tip}$. C) $I_{needle}(x,y)$ image, from which the needle tip is determined as the first bright intensity pixel at the distal end of the needle. D) Shaft enhanced image. The automatically localized tip is indicated by a red dot. The proposed method restores needle information when the shaft is discontinuous, the tip is inconspicuous or occluded by high intensity artifacts.

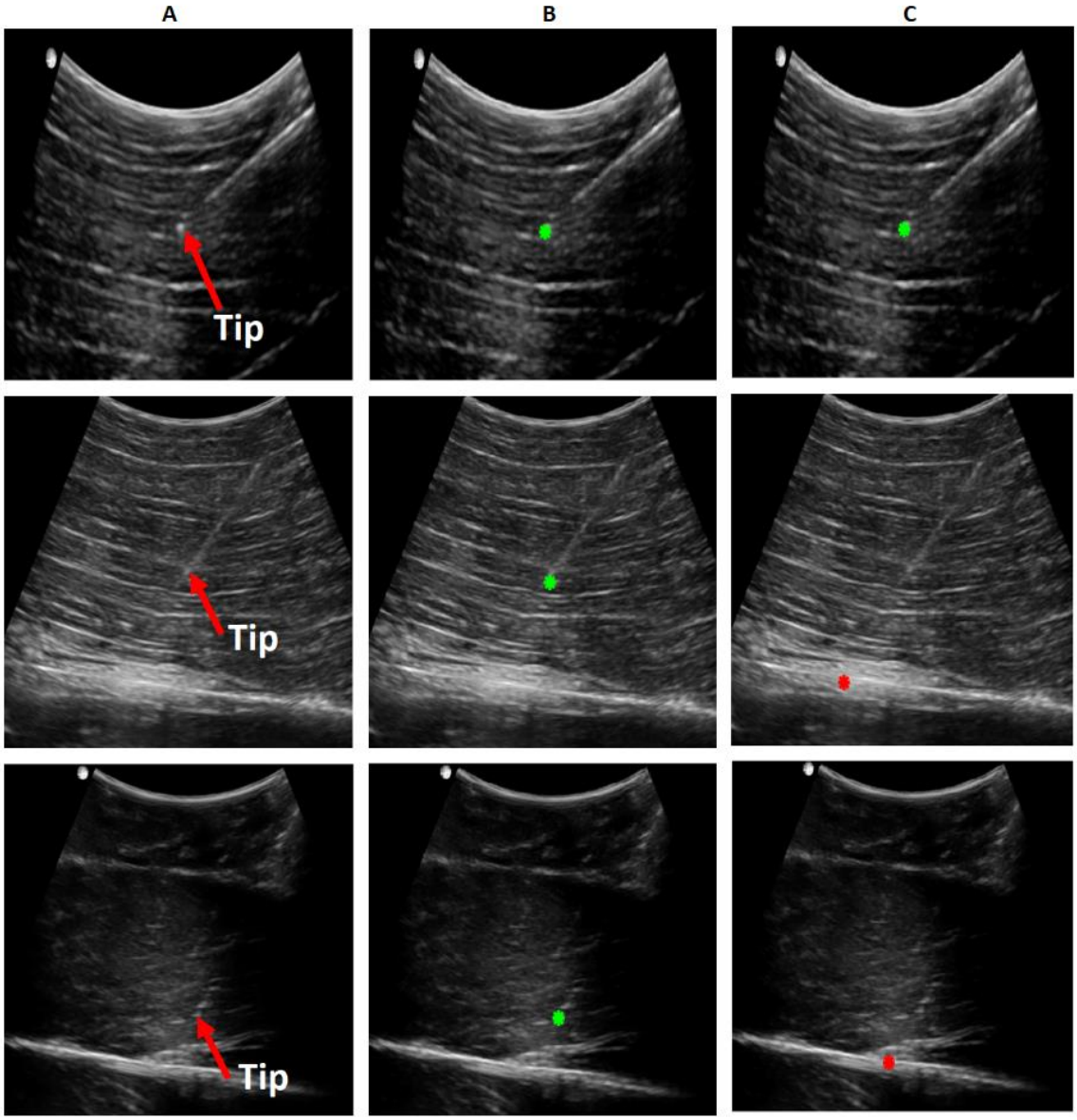


Figure 2-11. Comparing the proposed method with the method in [68]. Column A: US image $I(x, y)$. The arrow points to the expert localized tip. B: Tip localization from the proposed method. C: Tip localization by method in [68]. A green dot shows correct tip localization while a red dot shows inaccurate localization.

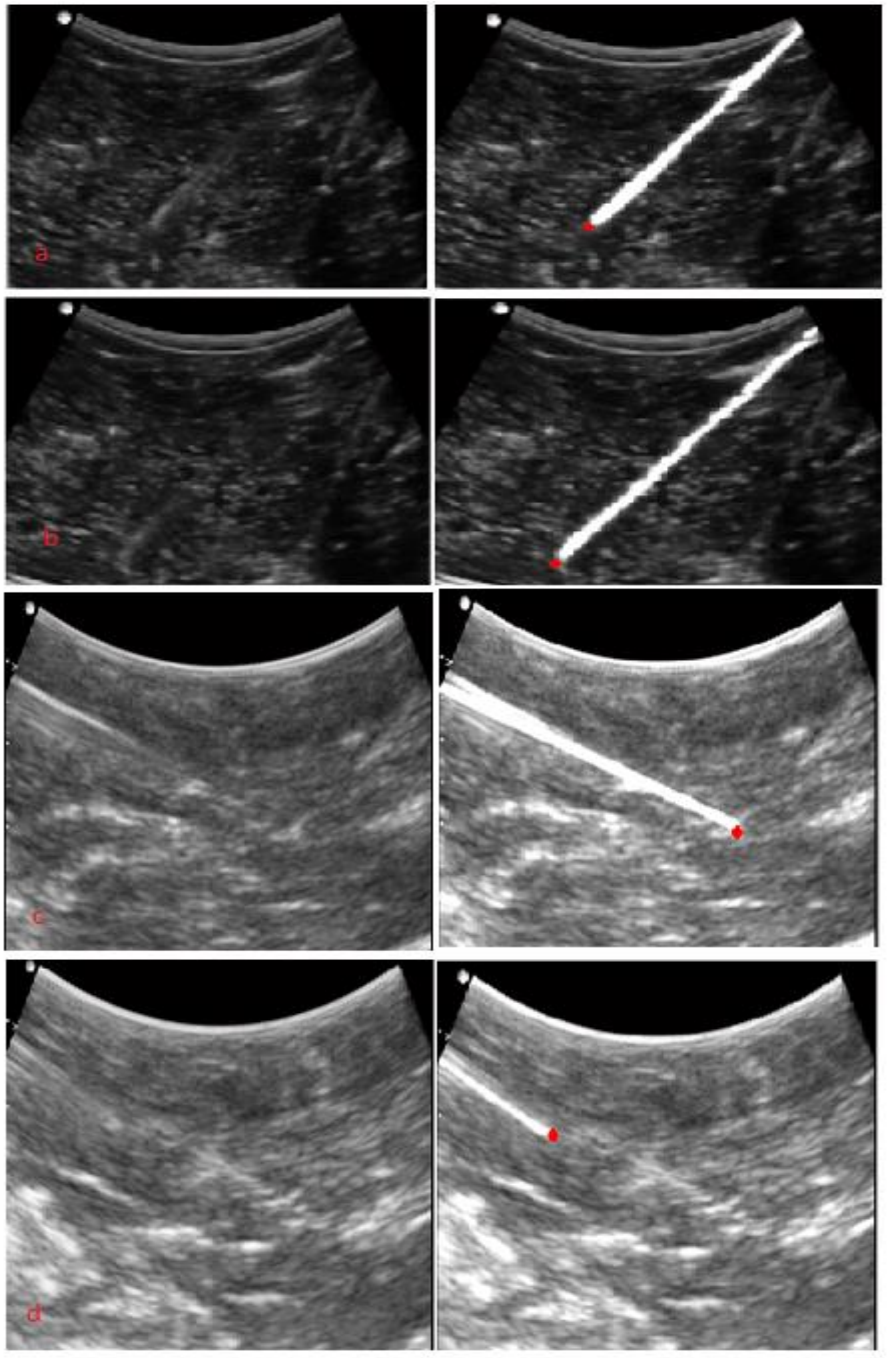


Figure 2-12. Qualitative results showing enhancement of the Biopince Full Core biopsy needle (a-b) and the 22GA BD spinal needle (c-d). Colum 1: 2D B-mode US images in which the shaft and tip are not easily identifiable. Colum 2: Enhancement of the shaft and tip using the proposed method.

2.5.2 Tip localization accuracy

Quantitative results from the proposed method are shown in Table 2-1. The localization error is invariant to tissue type, so we report aggregate errors for bovine and porcine tissue. The overall localization error from the proposed method was 0.55 ± 0.06 mm. In comparison, the localization error from the method in [68] was 0.88 ± 0.08 mm. For the methods in [68], we excluded outliers (localization error >2 mm). For the proposed method, all localization errors were less than 2mm. With the method in [68], only 56% of the dataset was retained. A one-tailed paired t −test shows that the differences between the localization errors reported in [68] and the proposed method are statistically significant ($p < 0.005$).

Table 2-1. Tip localization error (mm) from the proposed method and the method in [68].

Parameter	Proposed method	Method in [68]
Mean ED	0.55	0.82
RMS	0.62	0.89
SD	0.28	0.35
95% CI	0.06	0.08
Maximum Error	1.48	1.95

Further, recall that ν is optimally fixed at $0.3 \times max\,(I(x,y)$. We show the effect of ν on tip localization accuracy in Table 2-2. Values of $\nu \geq 0.5 \times max\,(I(x,y)$ are excluded since they result in complete loss of shaft information. The results are in consonance with qualitative results (Figure 2-5) regarding optimal choice for ν. In Table 2-3, we show effect of the regularization parameter, λ, on tip localization accuracy. It is observed that minimum error occurs at $\lambda = 1$, in agreement with qualitative results in Figure 2-3 where $\lambda = 1$ gave the

smoothest estimate of the signal transmission map. Also, recall that we chose a 100×100 ROI during the tip localization process. In Table 2-4, we show the effect of ROI size on tip localization accuracy. Note that small and larger ROI sizes result in inaccurate results because they lead to inclusion of inadequate shaft information or interfering artifacts respectively, thus reducing clarity of trajectory estimation.

2.5.3 Execution time

For a 500×500 2D image, the tip and shaft enhancement processes executed for 0.4 and 0.27 seconds respectively, while tip localization took 1.1 seconds. This gives an overall enhancement and localization time of 1.77 s ($\sim$0.5 fps).

Table 2-2. Effect of ν on tip localization error. % values are of $max\,(I(x,y))$, the maximum intensity in $I(x,y)$. Minimum error is obtained at 30%.

ν	0%	10%	20%	30%	40%
Mean ED	1.46	1.2	0.88	0.55	0.72
RMS	1.62	1.46	0.94	0.62	0.78
95% CI	0.44	0.24	0.14	0.06	0.08

Table 2-3. Effect of λ on localization error. Minimum error is obtained at $\lambda = 1$. For $\lambda \geq 10$, the increase in localization error with increase in λ is not statistically significant, because $t(x,y) \rightarrow \psi(x,y)$.

λ	0.1	0.4	0.8	1	10	50	200
Mean ED	0.86	0.74	0.65	0.55	0.96	0.98	0.99
RMS	0.65	0.67	0.66	0.62	0.85	0.96	0.94
95% CI	0.21	0.18	0.15	0.06	0.24	0.46	0.51

Table 2-4. Effect of ROI size on tip localization error. The optimum ROI was 100×100 pixels.

ROI size	220x20	40x40	60x60	80x80	100x100	120x120	140x140
Mean ED	.1.94	1.16	0.71	0.56	0.55	0.88	1.22
RMS	1.56	1.14	0.77	0.64	0.62	0.68	1.04
95% CI	0.88	0.48	0.14	0.09	0.62	0.12	0.56

2.6 Discussion

In this chapter, we have presented a framework that addresses the challenge of poor needle tip and shaft visibility in 2D curvilinear US at steep insertion angles. The proposed method evaluates the novel use of signal transmission maps and LMN contextual regularization to achieve needle shaft and tip enhancement in 2D US. The tip is localized automatically using spatially distributed image statistics from the estimated shaft trajectory, derived from local phase-oriented features. The proposed method achieves restoration of shaft and tip information at varying depths and insertion angles. The proposed method also provides better tip localization accuracy than a previously reported state-of-the-art method [68].

Considering the diameter of the needles used in this study and the resolution of state-of-the-art US machines, a tip localization error of < 1 mm is clinically acceptable. Different than [68], we demonstrate needle enhancement at steeper insertion angles ($40^{\circ} - 80^{\circ}$) and minimize the effect of high intensity artifacts along the needle trajectory.

The work in this chapter applies only to localization of straight in-plane inserted needles. Therefore, in procedures where bending needles are used, this approach would not work. Further, our method requires shaft and tip information to be partially available. In Figure 2-13, we show cases where these assumptions are violated, leading to failure of the method. In Figure 2-13 (top row), there is hardly any shaft information in the original US image; the enhanced image also

contains insufficient shaft information. As such, we can't proceed with the tip localization process. In Fig. 2-13 (middle and bottom rows), shaft information is available, but tip information is absent. Therefore, although the enhanced image facilitates the tip localization process, the tip location obtained is inaccurate. Nevertheless, for instances where the shaft and tip are somehow visible, the achieved tip localization accuracy and shaft enhancement makes our method appropriate for further investigation in vivo.

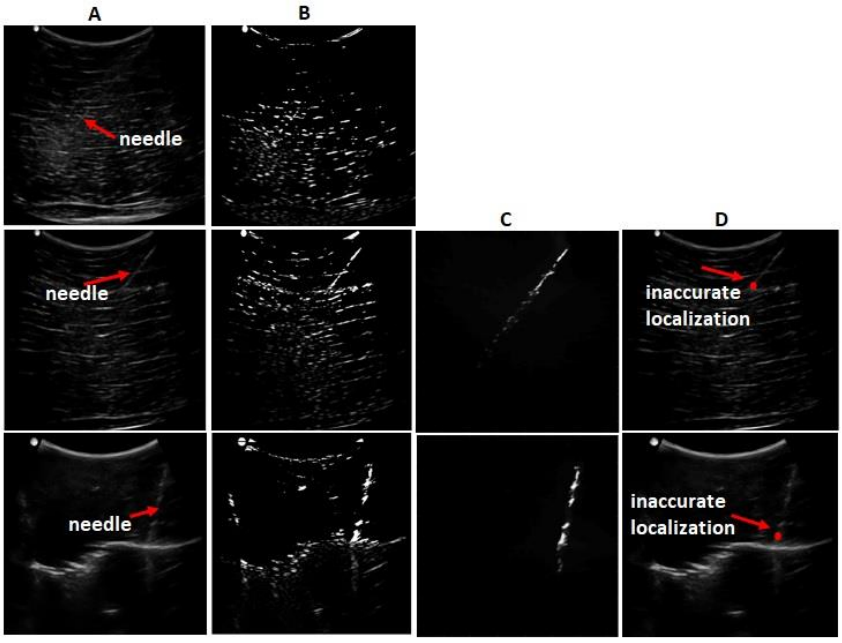


Figure 2-13. Examples of cases where the proposed method fails. A) US image $I(x,y)$. B) tip enhanced image $I_e(x,y)_{tip}$. C) $I_{needle}(x,y)$ image. D) Localized tip (red) marked on original image. In the top row, $I(x,y)$ contains insufficient shaft information, and thus $I_e(x,y)_{tip}$ contains no conspicuous shaft information. Therefore, we can't proceed with the tip localization process. In the middle and bottom rows, $I(x,y)$ contains enough shaft information but tip information is unclear. Presence of an enhanced shaft means we can attempt tip localization, but absence of an enhanced tip leads to incorrect tip localization.

We should note that the overall computation time for this approach is less than 1 fps. This can be improved by code optimization and use of a more powerful computer, although it would most likely still be untenable for a real time imaging. Further, the localization process is not semi-automatic since we must choose a fixed ROI in order to determine the trajectory. These limitations motivate the development of learning-based methods for needle localization in 2D US, described in Chapters 3, 4 and 5. These methods incorporate aspects of the enhancement and localization schemes described in this chapter.

Chapter 3 -Detection and localization of needles in 2D US

3.1 Overview

The use of deep artificial neural networks, often dubbed deep learning, has recently become prominent in computer vision applications. This trend has been driven by the availability of large image datasets and improvements in computational power supported by Graphical Processing Units (GPUs). One popular application of deep learning is computer-aided detection of objects in images. Such tasks are usually solved by deep CNNs.

In this chapter, we develop a CNN based framework for automatic and accurate detection of steeply inserted needles in 2D US. We demonstrate its application in needle trajectory estimation and tip localization. Our approach consists of a unified network, comprising a Fully Convolutional Network (FCN) and a Fast Region-based Convolutional Neural Network (R-CNN). The FCN proposes candidate regions for the needle, which are then fed to a Fast R-CNN for finer needle detection. We leverage a transfer learning paradigm, where the network weights are initialized by training with non-medical images and fine-tuned with *ex vivo* US scans collected during insertion of a 17G epidural needle into freshly excised porcine and bovine tissue at depth settings up to 9cm and $40^{\circ} - 80^{\circ}$ insertion angles. Needle detection results are used to accurately estimate needle

trajectory from intensity invariant needle features and perform needle tip localization from an intensity search along the needle trajectory.

The needle detection model achieves a frame rate of 25 fps on a GPU, 99.6% precision, 99.78% recall rate and an F_1 score of 0.99. Investigation of needle localization accuracy was performed on scans collected over a bovine/porcine lumbosacral spine phantom. Shaft localization error of 0.82 ± 0.4 degrees, tip localization error of 0.23 ± 0.05 mm, and a total processing time of 0.58 seconds were achieved. The proposed method is automatic and provides robust needle localization results in challenging scanning conditions. The accurate and robust results coupled with real-time detection and sub-second total processing make the proposed method promising in applications for needle detection and localization during challenging minimally invasive ultrasound-guided procedures.

3.2 Background

In machine learning, the general objective is to implement mathematical models which can be trained to produce useful outputs given input data. In so doing, computers learn to perform tasks from experience provided in the form of training data. Many machine learning models are of the supervised learning type, i.e., the computer is given a set of labelled or annotated training data and tasked with generating correct labels or annotations on previously unseen data [92]. Machine learning methods have for a long time been used in medical image processing, for instance, the use of SVM classifiers and clustering algorithms such

as k-nearest neighbor (k-NN) [93,94]. These learning models usually rely on handcrafted features; manually extracted features from raw data or features extracted by other models. However, it is not always easy to define features for a given task.

Recently, there has been an upscale of deep learning (DL) techniques for medical image processing. As opposed to traditional machine learning approaches, DL models perform automatic feature extraction. Although there are many DL strategies, CNNs have become most popular in image related tasks because they provide a powerful way to learn features in images [95]. Before we delve into CNNs, it is important to first understand the concepts underlying neural networks.

3.2.1 Artificial Neural Networks

Designed to mimic the functioning of the human neuronal system, a neural network comprises several connected computational units (neurons) which communicate by sending signals to each other via weighted connections. Each neuron can accept data from multiple inputs and then transmit the data to other neurons. Figure 3-1 shows a typical neural network which consists of two building blocks: 1) the nodes, akin to cell bodies in biological neurons, and 2) synaptic weights, which implement connections between the various nodes. The network consists of an input layer through which data enters the network, one or more hidden layers which learn non-linear features from the input data, and the output layer which delivers predictions. For our case, we are interested in the Feed-Forward (FF) network configuration, where informational flow is unidirectional,

from the input, through the hidden layers, to the output [96]. If nodes in a layer are fully connected to all nodes in the previous layers, and nodes in that layer do not have any connections, that layer is said to be fully connected (FC).

In the simplest case of a FF network, the relationship between the input $\mathbf{x}$ and output $\mathbf{y}$ at each layer is represented by the mathematical function:

$$\mathbf{y} = g(\boldsymbol{\theta}^{\boldsymbol{T}}\mathbf{x}) = \mathrm{g}\left(\sum_j \theta_j x_j\right) \tag{3.1}$$

Here, *g* denotes a non-linear transformation and $\boldsymbol{\theta}$ are the weights. Each layer computes a weighted sum of all the outputs from the previous layer and passes it through a non-linear transformation to yield the layer's activations. This process is repeated until the output layer. What results is a hierarchical transformation of the input, with the first layers extracting general features like edges while later layers extract finer features pertinent to the input. We shall highlight the commonly used non-linearities shortly. During training the process, the weights of the network are iteratively adjusted until the network makes good predictions for the training data. If this is achieved, we get the trained model, which is expected to similarly make good predictions on unseen data.

3.2.2 Convolutional Neural Networks

If we were to use a regular neural network for tasks involving images, we would be overwhelmed by computational complexity. Consider a $300 \times 300 \times 3$ image. First, we would need to flatten the image matrix into a $270{,}000 \times 1$ input vector. Then, a single neuron in the first fully connected layer of such a network would require $270{,}000$ weights! Obviously, we would need more neurons in that

layer and more layers, so the computational expense would exponentially increase. Moreover, with this approach, we would lose spatial dependencies, thus reducing accuracy of our predictions.

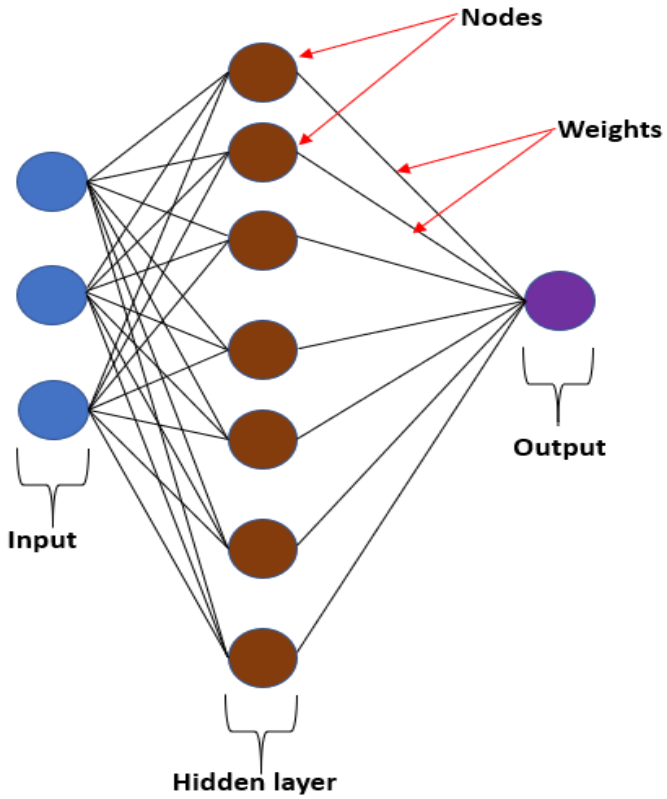


Figure 3-1. A typical neural network.

Convolutional Neural Networks (CNNs) simplify tasks involving images by arranging neurons in 3 dimensions: *width, height* and *depth*. Each layer then performs a transformation of the input volume, while maintaining spatial relationships in the data. Neurons in each layer are only connected to a small region of the previous layer.

3.2.3 Building blocks of CNNs

A typical CNN consists of the following layers:

Convolutional layers: Convolution consists of sliding a small filter (kernel) of size $K \times K \times n_c$ over the whole input $W \times H \times n_c$ corresponding to activations from the previous layer. Note that the depth n_c of the kernel corresponds to the depth of the input. At each filter position also known as the receptive field, the pixel values in the input are multiplied with pixel values in the filter in an elementwise manner and then summed. In so doing, each filter becomes a feature identifier, and the various convolutions hierarchically learn the object features. The values in the filter are the weights or parameters, and they are learned automatically during the training process, after which they are shared for computation in each layer. The filter is moved in increments dictated by the stride, S. Before applying the kernel, we can argument the input with zero padding, of dimension P. Padding helps to capture features along the edges during convolution. It has the effect of maintaining or reducing the width and height of the output. The output's width W_o and height H_o are given by:

$$W_o = \frac{W-K+2P}{S} + 1, \;\; H_o = \frac{H-K+2P}{S} + 1 \qquad (3.2)$$

The output's depth corresponds to the number of filters used. The output is hence a tensor of feature maps of size $W_o \times \; H_o \times n_c$.

When implementing a CNN, one must think of using the best model hyperparameters: the configurations external to the model which cannot be estimated from the data (unlike parameters, such as weights, which are

estimated/learned from the training data). For example, for convolution layers, hyperparameters include: the number of layers, and within each layer, the kernel size, the number of kernels, the stride and the padding. There are no set standards for these parameters. Usually, tuning these parameters, cognizant of the type of data one has and the task at hand is a critical part of implementing a good CNN model. Typically, a 3×3 kernel is used.

Activation layers: Tensors from the convolutional layers are fed through non-linear activation functions which facilitate learning of complex mappings between inputs and outputs. The most popular of these is the rectified linear unit (ReLU), defined as $f(z) = \max(z, 0)$. ReLU works like a half-wave rectifier circuit: if the input is negative, the output is zero, otherwise, the input is the same as the output. Previously, other non-linearities such as $f(z) = \tanh(z)$ or the sigmoid: $f(z) = 1/(1 + \exp(-z))$ were used. However, ReLU has been found to learn much faster in deep networks [97]. Graphical illustrations of these functions are in Figure 3-2. More recently, other non-linear activation functions such as leaky ReLU [98] and exponential linear unit (ELU) [99] have been found to perform well in some tasks.

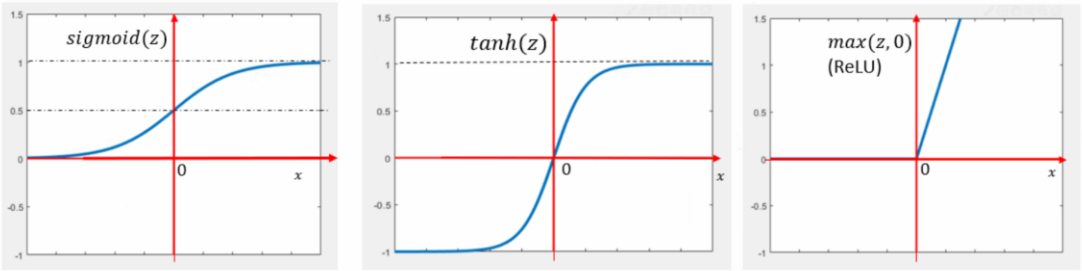


Figure 3-2. Common non-linear activation functions.

Pooling layers: Pooling serves to downsample the activations in a layer and reduce computational cost. Pooling takes a small grid of the volume as an input and returns a single number for that region. Pooling is implemented with a filter, say of size $K \times K$ and stride S. The filter is applied to the input volume, and at each location, it outputs either the maximum number in that sub-region (max pooling) or the average of the numbers (average pooling).

Pooling results in reduction of spatial dimensions. For an input volume $W \times H \times n_c$, the output volume's width W_o and height H_o are given by:

$$W_o = \frac{W-K}{S} + 1, \;\; H_o = \frac{H-K}{S} + 1 \qquad (3.3)$$

Pooling gives CNNs translational invariance because the relative location of features in the output activation maps doesn't change. Like convolutional layers, one must consider the filter size and the stride as the main hyperparameters for the pooling layers. Typically, a 2×2 filter with a stride of 2 is used.

Dropout: Sometimes, the weights of a network are over tuned to the training data such that the network performs poorly on data it hasn't seen. This inability to generalize to data different from training data is called overfitting. One strategy to reduce overfitting is by using a dropout layer, which eliminates a random set of activations in a layer during training [100]. In so doing, the network is 'forced' to give a correct prediction despite missing activations, hence preventing over tuning to the training data.

Overfitting is not that simplistic to eliminate. The performance of deep networks scales with the amount of training data available. This is because a

typical network will have millions of parameters. During the training which we will describe later, these parameters are tuned so that networks match predictions. For a network with many parameters, it follows that we need a proportionate amount of training examples to tune those parameters. Standard datasets in computer vision such as ImageNet contain a million or more images [101]. Unfortunately, it is difficult so have such large datasets for medical images. If a model is trained on little data, it is almost guaranteed to overfit. Therefore, the first remedy to overfitting is always to increase the size of the training dataset if possible.

It has also been found that a model pretrained on images in one domain, say natural images (which are abundant) can be leveraged in implementing a model for another domain, say medical images. The former is retrained to finetune its weights to the new domain. The retraining serves to finetune the weights to the new domain. This technique, called transfer learning [102] has gained prominence in the medical image computing community because of the limited datasets involved.

Yet another approach to reduce overfitting is by synthetically increasing the size of training datasets through augmentation: the images are rotated, translated or scaled to generate ‘new’ training examples which are treated by the neural network as such. The premise of data augmentation lies in the fact that CNNs are invariant to transformations such as translation and rotation.

Batch normalization: Input data for learning tasks is often preprocessed so that it resembles a normal distribution (zero mean, unit variance). This prevents early saturation of non-linear activation functions. During the training process, the

distribution of each layer's inputs keeps changing, and this can slow down the training process. This challenge is called internal covariate shift, and it is solved using a technique called Batch Normalization (BN) [103]. With Batch Normalization, the inputs of each layer are normalized (zero mean, unit variance) for each mini training sequence, in which a small section of the training data, known as a mini-batch is passed through the network. In so doing, it increases the stability and speed of the training process. BN layers are often put after activation layers.

Fully connected layers: One or more Fully Connected (FC) layers are usually put at the output end of the CNN. While convolution layers output high level features from data, FC layers learn non-linear combinations of the features. The output of each FC layer is a $N \times 1$ vector.

3.2.4 Common deep learning tasks in medical image analysis

Deep learning based on CNNs has recently gained momentum in medical image analysis. While this is not a comprehensive review, here is an overview of the most common applications with examples:

Image classification: The learned model is fed with an image and tasked with outputting a class label out of several possible categories, depending on the object that exists in the image. Such models have been developed for classification of malignant and benign breast masses [104], brain tumors [105], focal liver lesions [106] and laparoscopic videos [107]. These models are usually adapted from state-of-the-art models that have been developed for classifying natural images, for example AlexNet [108], VGGNet [109], GoogLeNet [110], and Microsoft ResNet

[111]. For classification tasks, the FC layers are often followed by a classification softmax layer with a cross-entropy loss.

Semantic segmentation: With semantic segmentation, we are not only interested in knowing that an image contains an object of a class; we are also interested in pixel-wise labeling of that object. Because spatial information in the image is lost when CNNs are fed into FC layers, semantic layers usually involve the use of Fully Convolutional Networks (FCNs) [112]. In an FCN, the FC layers of the CNN are replaced by transposed convolutional (deconvolution) layers which progressively up sample the inputs so to recover the original spatial dimensions of the input image while performing pixel-wise segmentation [113]. The most prominent architectures for semantic segmentation of medical images are U-Net [114], and V-Net [115].

Object localization: Localization or detection tasks focus on determining that an object contains an object (classification) and spatial localization of that object using a bounding box. To efficiently localize objects in images, the localization step is usually facilitated by some form of region proposal framework, so that the region in which the network should look to find an object is narrowed. State-of-the-art architectures for this task include Region Based CNNs (R-CNN [116], Fast R-CNN [117] and Faster R-CNN [118]) and YOLO [119,120]. In this dissertation, we are most interested in this task category: we want to use CNNs for detection of needles in US images.

Regression: Unlike classification where the output of a network is a class label, regression networks solve linear regression problems: we want to output a number or numbers that represent specific attributes of an image, for example, pose estimation [121]. Therefore, the objective is to predict continuous values. Such models have been applied in image registration [122-124].

3.2.5 Training a neural network

Training a neural network consists of altering its weights so that its output (prediction) matches its input. Typically, the network is shown an image or batch of images for which it produces outputs. These outputs are then compared to the groundtruth or expected outputs. An objective function calculates the error between the output and the expected outputs. This function is known as the *loss function* and the error is the loss. Loss functions are usually defined depending on the application. Common loss functions include mean absolute error (L_1), mean squared error (L_2) and cross-entropy loss.

After calculating the loss, the learning algorithm calculates the gradient of the loss with respect to the weights in each node of the network using the chain rule. This task is accomplished by the backward propagation algorithm [125]. This shows us the magnitude by which the loss would increase or decrease if the weights were to be adjusted by a tiny amount, and hence what weights contributed most to the loss. The network then changes its weights to reduce this error (the weights are changed in a direction opposite to that of the gradient). The weight adjustment is usually performed by an optimization algorithm called *Stochastic gradient descent* (SGD) [126]. The process is repeated for small sets of the training data (batches)

until the average the average of the loss function stops decreasing [95]. A complete pass through all the batches in the training dataset constitutes one epoch. The training pipeline is illustrated in Figure 3-3 and consists of 4 main phases: the forward pass for calculating weights, the loss function for calculating the errors, the backward pass for calculating the derivative of errors with respect to the weights, and the weight update.

If we W is the existing weight and L is the loss, then the weight update step is given by:

$$W_i \leftarrow W - \eta \frac{dL}{dW} \tag{3.4}$$

Here, W_i denotes the new weights, while η is a hyper-parameter that controls by how much the weights are adjusted, and it is known as the *learning rate*. Usually, the starting point for a learning rate is configured naively or through experience.

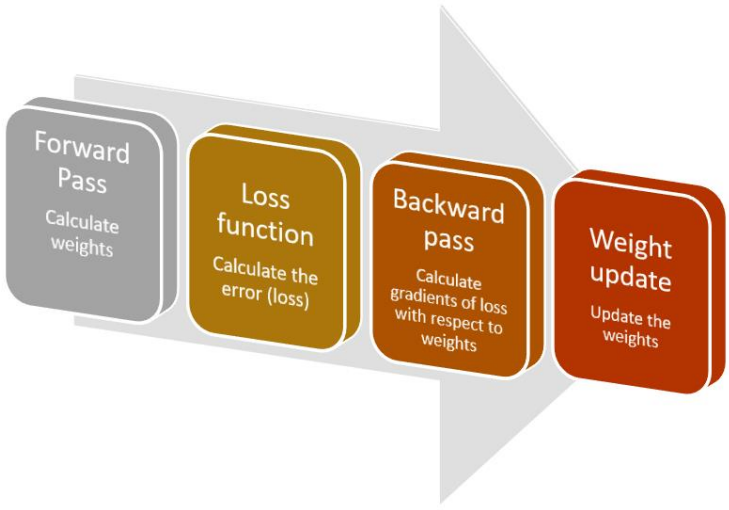


Figure 3-3. The network training process.

There are many versions of SGD optimizers [127], for example, Stochastic Gradient with Momentum (SGDM) [128], Adagrad [129], Adadelta [130], Adaptive Moment Estimation (Adam) [131] and RMSprop. All these allow one to set the initial learning rate, and internally configure it to decrease gradually as the trained model converges to the minimum loss.

A model which performs well on data it has not seen is said to have good generalization ability. This is often estimated during training using a separate dataset, the *validation* set, which is used as feedback to finetune the model. Once the training process is completed, the performance of the network is measured on a set of images the network has not seen before. This is called the *test* dataset. There is no golden rule for stratifying the data between the training, validation and test sets, although a rule of thumb is to use $\geq 80\%$ of the data for training.

3.2.6 Deep learning software

Neural networks are typically built in software frameworks. Common frameworks include Tensorflow [132], Keras [133], PyTorch [134], Caffe [135] and Theano [136] among others. MATLAB also provides libraries for implementing neural networks in its Deep Learning Toolbox. Because training deep neural networks is a compute intensive task, all the frameworks perform when supported by a GPU and NVIDIA's CUDA platform and the cuDNN library. In the next section, we present the design and implementation of our deep learning framework for needle detection in 2D US.

3.3 Deep learning framework for needle detection

To achieve fast and accurate needle detection, we propose the following main steps: 1) Generation of region proposals for the needle using a Fully Convolutional Network. Several regions of various scales and aspect ratios are proposed, and each is assigned a score related to the overlap with the ground-truth needle information in the US image. A multi-loss task is then used to determine final classification and location of the region proposals. 2) The region proposals are utilized for needle detection in a Fast R-CNN. Next, we describe the network architecture, and how it is trained to achieve a unified framework for fast needle detection and localization.

3.3.1 Network architecture

The proposed network architecture depicted in Figure 3-4 is based on the Faster R-CNN framework [118]. A Faster R-CNN is translational invariant. Therefore, needles of various sizes can be inserted at different depths and insertion angles, and the detector will perform accurately, irrespective of the needle's geometrical transformation. The Faster R-CNN consists of two major components: a Fast R-CNN [117] and a Region Proposal Network (RPN), modeled as an FCN. The two networks share convolution layers. This makes region proposal computation almost cost-free. We use convolution, max pooling, ReLU and FC layers as our building blocks. Hereafter, we describe design decisions that are specific to our architecture:

Fast R-CNN model: Our objective is to have a small network that provides a cost-effective solution with high needle detection accuracy. Therefore, we construct a custom Fast R-CNN consisting of only 5 learned layers: 3 convolution and 2 fully connected layers. The network's physical hyperparameters are shown in Table 3-1. Note that the FC layers are implemented using convolutions. The original Faster R-CNN implementation evaluated models with 16 learned layers derived from the VGG network [109] and 8 learned layers from ZF-NET [137]. The more the layers, the more the network parameters and hence increase in computational complexity.

Detection tasks focus on a small region of the input image where the object of interest might be located. The size chosen should be comparable the smallest detectable object. Therefore, we use a $32 \times 32 \times 3$ image input layer.

Table 3-1. Physical hyperparameters of the Fast R-CNN, the object detection part of the Faster R-CNN. The layers are illustrated in Figure 3-4.

N	Layer name	Kernel	Stride	Padding	$\mathbf{W \times H \times D}$
0	input	-	-	-	$32 \times 32 \times 3$
1	conv_1	3	1	1	$32 \times 32 \times 32$
2	relu_1	-	-	-	$32 \times 32 \times 32$
3	Maxpool_1	2	2	-	$16 \times 16 \times 32$
4	conv_2	3	1	1	$16 \times 16 \times 32$
5	relu_2	-	-	-	$16 \times 16 \times 32$
6	maxpool_2	2	2	-	$8 \times 8 \times 32$
7	conv_3	3	1	1	$8 \times 8 \times 64$
8	relu_3	-	-	-	$8 \times 8 \times 64$
9	fc_1	8	1	-	$1 \times 1 \times 64$
10	fc_2	1	1	-	$1 \times 1 \times 2$

In Figure 3-5, we show the strongest feature maps derived from the convolution layers (*conv_1*, *conv_2* and *conv_3*) for the same input US image. Examining the feature maps and comparing them with the input image, note that conv*_1* learns the rough features specific to the needle. On the other hand, *conv_2* and *conv_3* learn finer needle features.

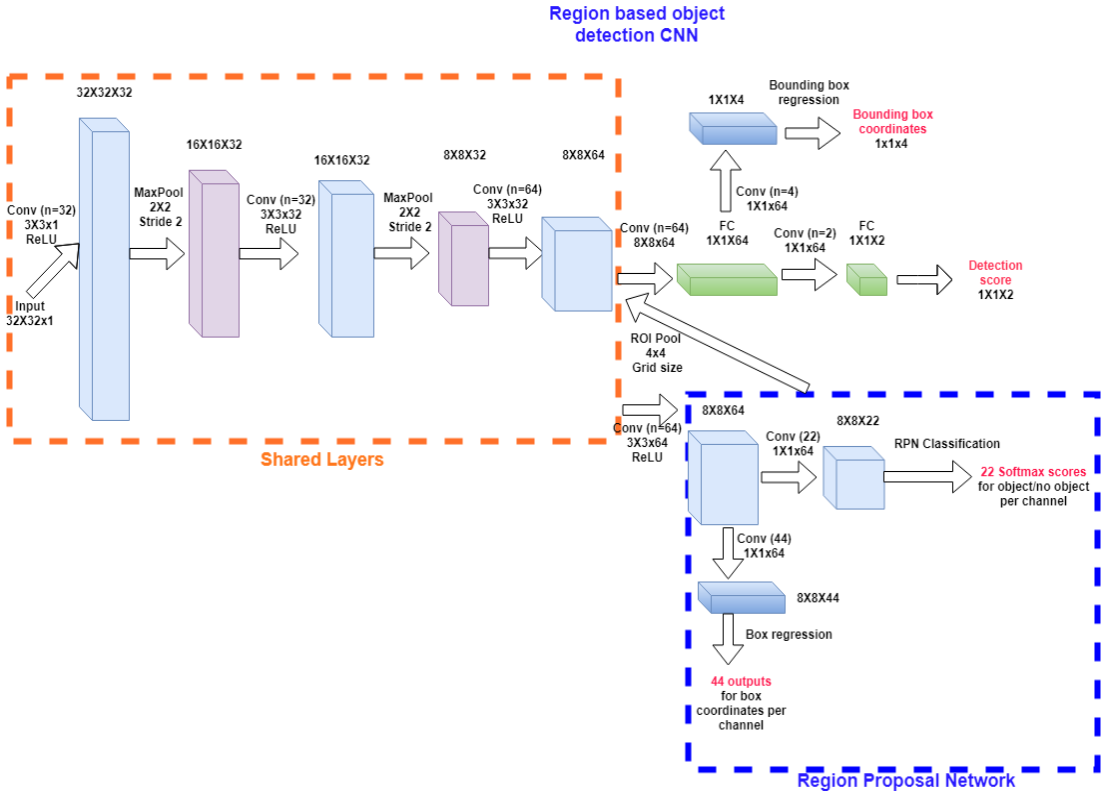


Figure 3-4. Needle detection architecture.

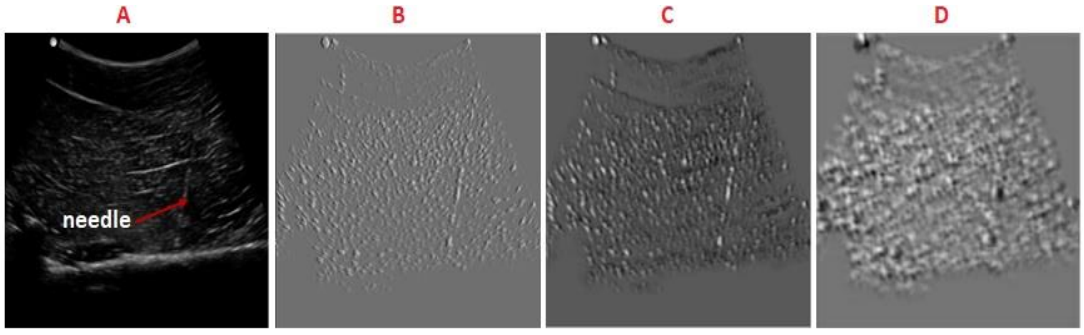


Figure 3-5. Visualization of the strongest feature maps from *conv_1* (B), *conv_2* (C) and *conv_3* (D) for the same input image (A). *conv_1* learns the discriminant linear features of the needle, and finer needle features are refined by the subsequent layers.

Regional Proposal Network (RPN): The RPN generates "areas of interest" for the Fast R-CNN. This is achieved by ranking potential bounding boxes for the needle (called anchors). In our design, the input to the RPN is the feature map of the last convolution layer in the Fast R-CNN (*conv_3*). We apply a sliding 3×3 convolution window over this feature map. At each window location, a maximum of 9 anchor boxes are predicted, generated from 3 scales with a scaling stride of 1.5× $(d_m, 1.5 \times d_m, 2.25 \times d_m)$ and 3 aspect ratios (1:1, 1:2, 2:1), where d_m corresponds the minimum dimension of the bounding boxes in the labeled training images. For a feature map of size $w \times h$, the maximum number of anchors is equal to $w \times h \times 9$. For our case, this is ~600.

To minimize the number of "active" anchor boxes, we eliminate cross-boundary anchors. This leaves ~60 anchors per image. Each anchor box is then assigned a positive class label (needle) if the Intersection-over Union (IoU) overlap with the needle bounding box in the labeled US image is > 0.7. Conversely, a negative class label (no needle) is assigned if IoU < 0.3. In so doing, we detect whether the region centered at each sliding window location contains needle data, and the sliding window location encodes coarse localization information with reference to the input US image. Each window location maps to a 256-dimensional intermediate layer, and lastly, to two sibling 1x1 fully connected layers, shared across all sliding window locations, for box-classification (needle or no needle) and box-regression (for finer localization information). The classification layer outputs a maximum of 22 scores, and the regression layer has a maximum of 36 outputs encoding the coordinates of the 44 anchor boxes, for each sliding window location.

The outputs of these fully connected layers are determined by minimizing the following RPN multi-task loss function [117]:

$$L(p_j, {p_j}^*, t_j, {t_j}^*) = \frac{1}{N_c}\sum_j L_c(p_j, {p_j}^*) + \frac{\lambda}{N_r}\sum_j {p_j}^* L_r(t_j, {t_j}^*). \quad (3.5)$$

Here, the first term describes the box-classifier and the second is the box regressor. j is the anchor index, p_j denotes the associated predicted probability, and t_j is the predicted location. ${p_j}^* \in [0,1]$ is the ground-truth label arising from the IoU scores described earlier, and ${t_j}^*$ is the associated location. L_c is the log loss over two classes (needle or no needle) and L_r is the regression loss. L_r is a smooth L_1 loss [118]:

$$L_r(t, t^*) = \begin{cases} 0.5(t - t^*)^2, \ if \ |t - t^*| < 1 \\ |t - t^*| - 0.5, \ otherwise \end{cases} \quad (3.6)$$

In (3.5), N_c is set to 256, the mini-batch size used in our experiments, while N_r is set to 600, the approximate number of total anchor locations in our derived convolution map of the RPN. The regularization parameter, λ, is set to 10 [108]. Since the derived region proposals may overlap, the use non-maximum suppression based on p_j, with a threshold of 0.8. The top-N ranked proposal regions are then fed into the Fast R-CNN to generate overall classification and tightened bounding boxes.

3.3.2 Training

To achieve a unified network, the RPN and Fast R-CNN are trained using a 4-step alternating process [118]: Step 1) The RPN is trained end-to-end. Step 2) The Fast R-CNN network is trained using region proposals derived from Step 1.

Step 3) The RPN is retrained using shared weights of the Fast R-CNN from Step 2. Step 4) Keeping the shared layers fixed, the Fast R-CNN is retrained using the updated region proposals from Step 3. We utilize stochastic gradient descent with momentum (SGDM), an initial learning rate of 10^{-4}, a momentum of 0.9, a mini-batch size of 256, and 10 epochs for each stage. Fast R-CNN layer weights are initialized by pre-training with the CIFAR-10 dataset [138]. Our implementation uses MATLAB's Deep Learning Toolbox.

3.4 Dataset overview

As earlier mentioned, deep neural networks require datasets comprising tens of thousands of images since they typically have millions of parameters (weights) to optimize. However, when dealing with medical images, data sizes are usually small. A popular technique to circumvent this problem is transfer learning, where a pre-trained network (typically on non-medical images) is fine-tuned using medical data, in our case, the domain specific US images containing needle information. For this purpose, we pretrain our network on the CIFAR-10 dataset.

We collected 2D B-mode US images using a SonixGPS system (Analogic Corporation, Peabody, MA, USA) with a 2D C5-2/60 curvilinear probe. A 17-gauge (1.5 mm diameter, 90 mm length) Tuohy epidural needle (Arrow International, Reading, PA, USA) was inserted into different samples of freshly excised bovine and porcine tissue in plane, at various insertion angles ($40^{\circ} - 75^{\circ}$) and insertion depths (up to 9 cm). We also overlayed bovine/porcine tissue on a lumbosacral spine phantom, and collected images using a 2D hand-held wireless US system

(Clarius C3, Clarius Mobile Health Corporation, Burnaby, British Columbia, Canada). These images were used only for validating needle trajectory and tip localization and were not included during training of the proposed network.

Using 2500 images from SonixGPS system, we performed 10-fold cross-validation: the images were randomly partitioned into 10 subsamples, each of size 250. In turn, 9 of the subsamples (2250 images) were used as training data while the other 250 images were used as a validation set. The cross-validation process was repeated, with each of the 10 subsamples used exactly once as validation data. The training images were labeled by an expert sonographer to indicate needle locations.

Image Preprocessing: US images from minimally invasive procedures such as biopsies and epidural spinal injections may contain high intensity artifacts which increase the likelihood of false positives, thus reducing accuracy of needle detection. Pre-processing aims to reduce the influence of such artifacts. First, the B-mode image $I(x,y)$ is subjected to a Top-hat filter using a linear structuring element $L(x,y)$, and the filtered image $F(x,y) = I(x,y) - [I(x,y) \circ L(x,y)]$. Here, $I(x,y) \circ L(x,y)$ denotes an erosion operation computed by $((I(x,y) \ominus L(x,y)) \oplus L(x,y))$; $\ominus$and $\oplus$ are morphological erosion and dilation operations respectively. The contrast of the filtered image is then stretched to yield $P(x,y) = \max(F(x,y) \times F(x,y) \,/\, \max(F(x,y))$, where $\max(F(x,y)$ is the maximum intensity in the filtered image.

3.5 Detection results

The trained model was evaluated on 2500 US images. Figure 3-6 shows sample qualitative results. In all cases, the needle shaft is detected, despite low shaft intensity. Needle detection is independent of insertion depth, but rather on availability of needle data in the US image to support the learning process. We have focused on demonstrating our approach on cases where the needle is slightly off-plane or there is substantial reflective loss of the backscattered US signal between the needle and the transducer, as is typical for steep insertions. In practice, these are challenging cases because the shaft will be invisible or discontinuous, and the tip will be separate from the shaft, and may also be imperceptible. For this reason, it is difficult for our approach to detect the whole needle despite the use of multiple scales and aspect ratios during the learning process. Indeed, due to needle discontinuity, various regions of the needle may be detected separately. This scenario is illustrated in Figure 3-7. In such cases, we derive the region with the highest confidence score. As we will show in the next section, this is not an impediment to the needle localization process: this detection information is leveraged to automatically estimate the needle trajectory, and henceforth accurately localize the tip.

Training of the Faster R-CNN network took an average of 42 minutes on a single NVIDIA GTX 1060 6GB GPU. The mean needle detection time was 0.04 s. This is a real-time rate and corresponds to 25 fps. Overall, average precision and recall rates of 99.6% and 99.78% respectively, and an F_1 score of 0.99 were achieved. The role of preprocessing in improving detection accuracy cannot be

overstated. When the same dataset is analyzed using the proposed approach, but with the network trained on US images that are not preprocessed, the overall precision score drops to 69.83%, the recall to 73.88% and the F_1 score to 0.72.

We investigate the performance of the proposed Faster R-CNN through ablation studies, where we remove or add features to the network and analyze the corresponding effect on its performance. First, we remove the RPN. This leaves a stand-alone Fast R-CNN network. The network weights are also initialized using the CIFAR-10 dataset and fine-tuned with US data. Next, we keep the RPN and Fast R-CNN intact but train the network with only random weight initialization, without the CIFAR-10 dataset.

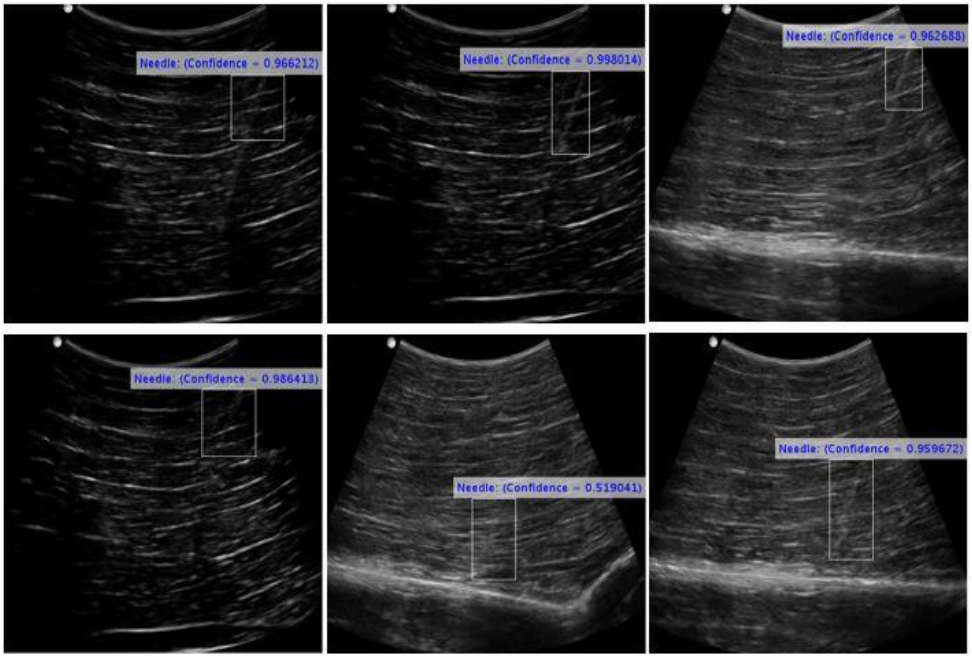


Figure 3-6. Needle Detection results. The needle shaft is accurately localized, despite low (top row) or imperceptible (bottom row) shaft intensity. The numbers on the bounding boxes are detection scores; a measure of the confidence of detection. When multiple detections exist, we choose the highest detection score.

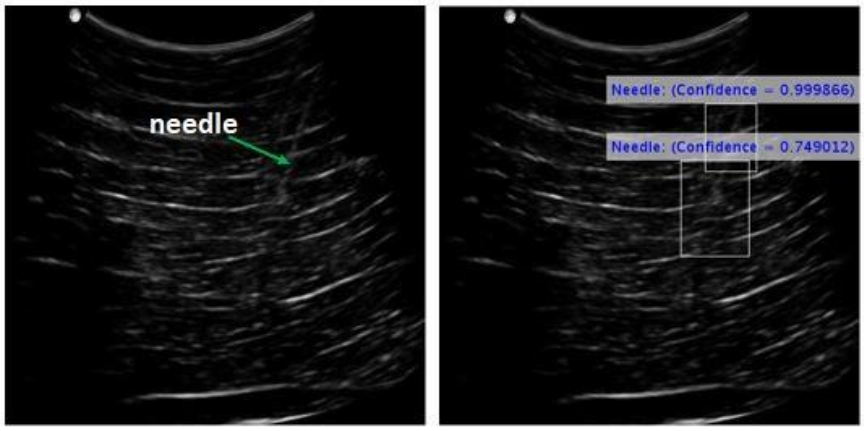


Figure 3-7. Multiple needle detection for the same needle. Left: US image. Right: Needle detection results. In such a case, we extract the feature with the highest detection score.

We call this Faster R-CNN1. Lastly, to show the effect of convolution layers, we train and test the network as previously described, but with one convolution layer removed (Faster R-CNN2), and then by adding two more convolution layers (Faster R-CNN3). We also implement a model based on the VGG network [109], pre-train it with the CIFAR-10 dataset and then fine-tune the weights with US data.

Table 3-2 shows a comparison of the performance of these networks. We performed 10-fold cross validation only with the proposed method, using a dataset of 2500 images as earlier mentioned, after proving its superior performance to the ablation variants. For the other networks, reported results are from training on a dataset of 1500 randomly selected images and testing on 400 images that were not part of the training data. The proposed method gives the best combination of detection time and F_1 score. Addition of 2 convolution layers (Faster R-CNN3) doubles the computational cost and reduces the F_1 score to 0.88. Reducing the number of convolution layers by one reduces the F_1 score to 0.83. Meanwhile, the

VGG16 network is the slowest of all: the detection task is computationally heavy (~1 second on a GPU). This is anticipated since VGG16 has 41 layers (16 learned layers). Next, we present applications of the localization information deduced from the needle detection process.

Table 3-2. Comparing the proposed method with ablation variants. Faster RCNN1 is trained without pretraining with non-medical images. Faster RCNN2 has one less convolution layer than the proposed method, while Faster RCNN3 has 2 more convolution layers.

	Training time (minutes)	**Test time (seconds)**	**Precision (%)**	**Recall (%)**	**F_1 score**
Proposed method	42	0.04	99.6	99.78	0.99
Fast R-CNN	9	0.84	89.2	97.2	0.93
Faster R-CNN1	29	0.04	79.7	89.3	0.84
Faster R-CNN2	24	0.04	84.7	80.8	0.83
Faster R-CNN3	66	0.08	86.3	90.0	0.88
Model based on VGG	56	1.03	88.0	84.0	0.86

3.6 From needle detection to localization

Ultimately, the desired output is an enhanced needle image which can facilitate automatic needle shaft and tip localization. In Figure 3-6 and 3-7, we observe that our method may not give a bounding box that contains the whole needle. While this is a limitation, the detection result is enough for automatic detection of the needle insertion side, estimation of the needle insertion trajectory, and facilitating automatic localization of the tip. This addresses the limitations of previous methods [68, 139,140], where there is need for *a priori* knowledge of the needle insertion side and selection of a fixed ROI close to the transducer surface. In the process, our detection framework makes the entire process of needle

localization fully automatic. Next, we describe how we achieve automatic tip localization.

3.6.1 Needle Trajectory localization

Recall that the detection step yields a bounding region for the needle, which we use as an automatically generated ROI. First, we construct a phase-based image descriptor, called phase symmetry ($PS(x, y)$), using an orientation-tuned 2D Log-Gabor filter bank [68] applied to the ROI on the pre-processed US image (Figure 3-8(B)). Previous approaches utilized a similar technique but extracting the PS image required *a priori* information: an estimate of the needle trajectory or insertion side of the needle (for θ_m) and a fixed ROI containing the shaft, close to the transducer surface [68, 139,140]. Now, we are determining the ROI, the insertion side and the insertion angle automatically. The ROI corresponds to the needle bounding box from the detection process. Figure 3-9(A) shows how we automatically determine the insertion side and an estimate of the insertion angle from this bounding box. We define the bounding box with parameters x_i, y_i, L and W. If (x_c, y_c) is the center of the image, for right-side insertions, $x_i < x_c$, and for left-side insertions, $x_i > x_c$. For the former case, an estimate of the needle trajectory, β is given by $\beta = tan^{-1}(\frac{W}{L})$ and for the latter, $\beta = 90 + tan^{-1}(\frac{W}{L})$. The filter bank is applied with 3 scales and 3 orientations, $\theta_m = [\beta - 10,\ \beta,\ \beta + 10]$, and yields $PS(x, y)$, containing a prominent needle feature.

The $PS(x, y)$ image may contain artifacts not belonging to the needle. We eliminate these by extracting the longest connected component. The output of this

operation, $PS_L(x, y)$ is shown in Figure 3-8(C): a distinct, intensity invariant straight feature. The last step in trajectory estimation involves application of the Hough transform (HT). The automatically estimated trajectory is shown in Figure 3-8(D). We calculate trajectory error by comparing the automatically determined trajectory with the gold standard trajectory estimated by an expert sonographer. The parameters used in computing this error are shown in Figure 3-9(B). If we denote the original US image as $I(x, y)_{m \times n}$, where *m* and *n* are the horizontal and vertical dimensions, then the center of the image is estimated as $(x_c, y_c) = (m/2, n/2)$. We calculate: 1) γ_1, the angle subtended by the automatically detected trajectory on the horizontal axis, 2) γ_2, the angle subtended by the trajectory labeled by an expert, on the horizontal axis, 3) λ_1, the shortest distance between the automatically detected trajectory and the center of the image, and 4) λ_2, the shortest distance between the expert-labeled trajectory and the center of the image. The trajectory error is then quantified using $\gamma_1 - \gamma_2$ and $\lambda_1 - \lambda_2$.

In Figure 3-9(C), we show a qualitative comparison between trajectory localization from the proposed method and the expert-localized trajectory. Quantitative results from analysis of 400 US images collected with the same imaging system as the training dataset, but not part of the training data, are shown in Table 3-3. We also present results from analysis of 400 images collected with the Clarius US system, which were not part of the training data, in Table 3-4. Also shown are corresponding results from applying the state-of-the-art methods in [68] and [140] on the same datasets. The proposed approach gives superior

performance to these methods, with statistically significant improvement in trajectory error ($p < 0.005$), obtained using one-tailed paired t-test.

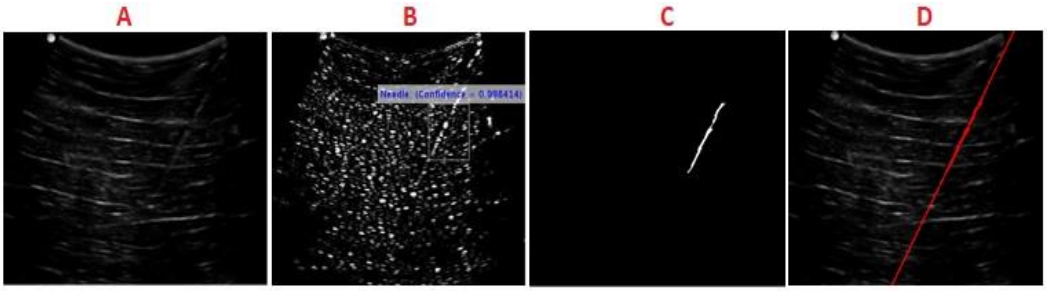


Figure 3-8. Needle trajectory estimation. A) US image, B) Preprocessed image with a marked ROI. The ROI is automatically determined as the needle bounding box from the detection step. C) Image after extraction of local phase features and finding the longest connected component. D) Estimated trajectory (red line) after applying the Hough Transform.

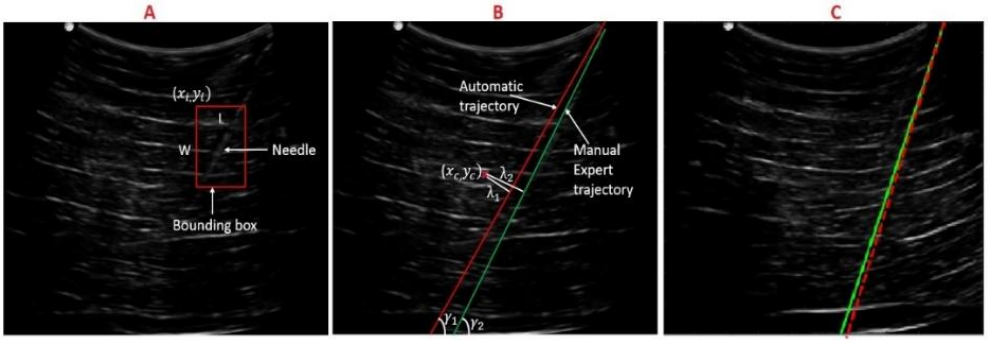


Figure 3-9. Trajectory estimation parameters. A) Estimation of needle insertion side and trajectory. Initial estimate of the trajectory is obtained from the diagonal of the needle bounding box. B) Parameters used in calculating needle trajectory error. (x_c, y_c) is the center of the image. The automatically determined parameters are subtracted from the parameters obtained manually by an expert. Here, the difference between the two trajectories is exaggerated for illustration purposes. C) Trajectory from proposed method (green) and expert-labeled (red, dashed) overlayed on US image. The proposed method gives an accurate prediction of the trajectory.

However, there is no significant difference between the results obtained with the two imaging platforms, over different imaging media. This result emphasizes the clinical relevance of our approach.

Table 3-3. Trajectory Localization errors on data collected with the SonixGPS US system.

Method	$\gamma_1 - \gamma_2$ (degrees)	$\lambda_1 - \lambda_2$ (mm)
Proposed Method	0.8 ± 0.4	0.4 ± 0.1
Method in [68]	1.5 ± 0.6	1.1 ± 0.2
Method in [140]	1.8 ± 0.8	1.4 ± 0.2

Table 3-4. Trajectory Localization errors on data collected with the 2D Clarius C3 system over a bovine/porcine lumbosacral spine phantom.

Method	$\boldsymbol{\gamma_1 - \gamma_2}$ (degrees)	$\boldsymbol{\lambda_1 - \lambda_2}$ (mm)
Proposed Method	0.82 ± 0.4	0.38 ± 0.1
Method in [68]	1.74 ± 0.6	1.31 ± 0.2
Method in [140]	1.86 ± 0.8	1.42 ± 0.2

3.6.2 Tip localization

From the estimated needle trajectory, we create a mask of the trajectory region; a region extending a few pixels on either side of the needle axis, where we are sure the needle axis must lie. Next, we use the approach in Chapter 2 [68,140]. A mask of the trajectory region is convolved with the preprocessed US image. This is followed by line fitting using the MLESAC algorithm [91]. Finally, the needle tip is automatically localized by filtering the resulting image with a 2D Log-Gabor filter, and performing a statistical search along the trajectory. The processing pipeline is illustrated in Figure 3-10. Evaluation of tip localization is performed on 400 US

images collected with the Clarius US system. Overall, the tip localization process (excluding needle detection) executes for 0.54 seconds.

In Figure 3-11, we show qualitative tip localization results for both images collected over the spine phantom and soft tissue. The former demonstrate clinical feasibility for procedures such as lumbar facet joint and medial branch blocks. Note that despite interference from bone and other artifacts, the needle tip is accurately localized.

The tip localization error is determined from the ED between the automatically localized tip and the manually localized tip by an expert sonographer, and yields 0.23 ± 0.05 mm. On the other hand, the methods in [68] and [140] yield localization errors of 0.55 ± 0.12 mm and 0.84 ± 0.25 mm respectively. Comparing the proposed method and these previous methods, there is a statistically significant improvement in tip localization accuracy ($p < 0.005$) obtained using a one-tailed paired t-test.

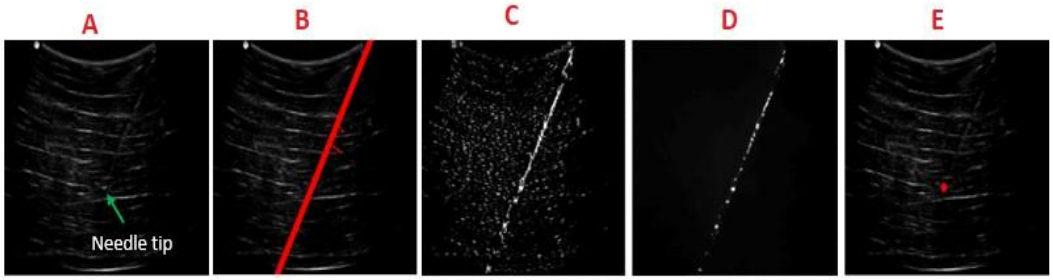


Figure 3-10. The needle tip localization process. A) US image. B) Extended trajectory region (red) computed with the HT and used to form the trajectory mask. C) Output of MLESAC algorithm. D) Enhanced needle image. E) Automatically localized tip (red).

3.7 Discussion

In this Chapter, we have presented a novel method for detection of needles in 2D US data, utilizing convolution neural networks. Our method achieves high precision (99.6%) recall rate (99.78%), and a detection time of 0.04 seconds.

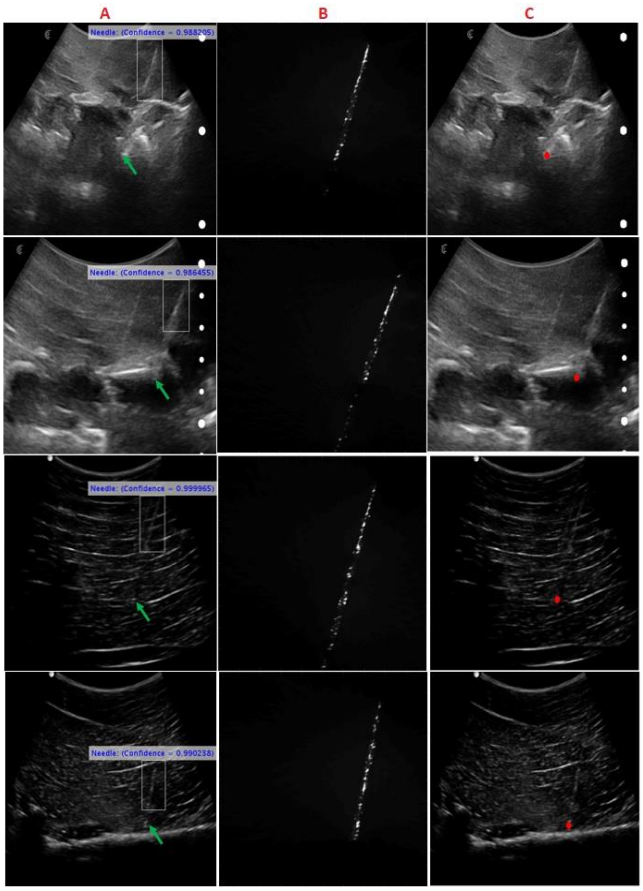


Figure 3-11. Qualitative results for tip localization. A) US image. The green arrow points to the expert-localized tip. B) Enhanced tip image, C) Localized tip (red). Rows 1 and 2 show results for imaging over a bovine lumbosacral spine phantom using Clarius US system while rows 3 and 4 show results for images collected over soft tissue using SonixGPS imaging system. The proposed method achieves automatic and accurate tip localization despite low or inconspicuous needle information, and high intensity interfering artifacts.

The proposed method doesn't always achieve detection of the whole needle, especially if the needle shaft is broken, as is typical at steep insertion angles. However, the detected portion of the needle provides an automatically generated ROI, which is utilized for trajectory estimation and tip localization.

Evaluation of the method on images collected with a different imaging system and not part of the training dataset reveals better trajectory estimation and tip localization accuracy than previously reported methods [68, 140]. The overall detection and localization time of 0.58 seconds (~2fps), though not real-time, is better than state-of-the-art methods, and can further be improved with better computing hardware.

The proposed method is promising for clinical application, and would work seamlessly in any imaging scenario, if the network is trained on a larger clinical dataset incorporating different imaging conditions. Although we have focused on hand-held needles, our method could find use in minimally invasive robotic interventions.

The proposed method achieves needle detection in cases where the needle is imperceptible. However, our current localization approach, based on the Hough transform, only works for non-bending needles.

Chapter 4 -Needle localization from digital subtraction in 2D ultrasound

4.1 Overview

In Chapters 2 and 3, we have addressed the challenge of needle localization in static 2D US frames. Although we have achieved good needle localization accuracy, we still have limitations on processing speed. Moreover, both approaches presented so far suit only in-plane needle insertion and assume that the needle follows a straight trajectory. The approach in this chapter seeks to remedy these limitations.

Here, we propose a novel approach for localizing the needle tip from dynamic 2D US (video) data, with needles inserted both in-plane and out-of-plane. We treat the needle tip as a moving target in each video frame. As the tip moves through tissue, it causes subtle spatiotemporal variations in intensity. Relying on these intensity changes, we formulate a foreground detection scheme for enhancing the tip from consecutive US frames. The tip is augmented by solving a spatial total variation regularization problem using the Split Bregman method. Lastly, we filter irrelevant motion events with a deep learning-based end-to-end data-driven method that models the appearance of the needle tip in US images, resulting in needle tip detection.

The detection model is trained and evaluated on an extensive ex vivo dataset collected with 17G and 22G needles inserted in-plane and out-of-plane in bovine and porcine phantoms. We use 5000 images extracted from 20 video sequences for training and 1000 images from 10 sequences for validation. The overall framework is evaluated on 500 images from 10 sequences not used in training and validation and achieves a tip localization error of 0.83 ± 0.02 mm, and an overall processing time of 0.094 s per frame (10 fps).

4.2 Background

Foreground detection represents a class of problems in computer vision and image processing, in which we are interested in detecting changes in image sequencies. For example, when a needle is inserted in tissue, the image scene before insertion will be disrupted as the needle advances. Foreground detection then consists of treating the needle as the foreground against a background formed by the rest of the image. Then from consecutive frames, the task is to isolate the foreground from the background.

Several works utilizing this approach are reported in literature. For instance, an attempt was made to detect and segment biopsy needles from transrectal ultrasound (TRUS) images [141]. In this clinical scenario, the needle is fired with a biopsy gun and lasts in few frames of the video sequence. There is also definitive contrast between the needle shaft/tip and the background. Changes in the video scene are detected with a background model, and needle segmentation follows by energy minimization using graph cuts [142]. Other works attempt to localize

needles from dynamic intensity changes arising from needle movement in the US image utilize optical flow [69,143]. Optical flow works best if intensity levels associated with needle motion exhibit a smooth transition. Further, optical flow assumes that neighboring points in an image always belong to the same feature and move together. These two assumptions reduce its reliability for dynamic needle localization since US images are sensitive to speckle, and susceptible to artifacts arising from abrupt changes in transducer motion, patient movement, hyper-echoic anatomy or physiological events such as pulsation and breathing.

In our approach, we propose a robust needle tip localization strategy in 2D US that combines a computationally efficient tip enhancement framework and a supervised deep learning approach that captures the expected tip shape and variation. By learning the expected features associated with the needle tip, the model successfully localizes the tip in presence of motion artifacts and abrupt intensity changes. The main contributions of this chapter are: 1) a novel digital subtraction algorithm that performs differencing of consecutive image frames within the US sequence. Thus, we can extract salient motion from temporal relationships in the US sequence with a dynamic background model, 2) an augmentation technique for the needle tip, in which we extend the Split-Bregman approach to solve a spatial total variation (TV) problem for the tip enhanced image, and 3) a deep learning framework optimized for needle tip detection from end-to-end learning. The detector learns contextual patterns associated with the needle tip and outputs static bounding boxes, from which the needle tip position is estimated.

The proposed method achieves both in-plane and out-of-plane needle localization, as well as localization of thin needles since it doesn't depend on full needle visibility. This is achieved at significantly faster computational accuracy than state-of-the-art. Our method could be utilized in a smart computer assisted interventional system to facilitate needle localization in the presence of artifacts from anatomical features and other instruments.

4.3 Methods

The proposed method is designed for hand-held 2D US probes during in-plane and out-of-plane needle insertion. The problem of motion-based needle localization can be split into two main components: 1) Detecting moving objects in each frame, and 2) associating the detections corresponding to the needle over time. Consequently, the proposed method consists of three main stages illustrated in Figure 4-1: 1) we detect scene changes caused by needle motion in the US image scene. In each frame of the US sequence, the needle tip is treated as the foreground, while the rest of the image is designated as background data. Needle enhancement is performed from logical subtraction of the dynamic reference US frame from the US frame of interest. This step doesn't require a priori knowledge of needle insertion side or angle, 2) we augment the appearance of the enhanced needle tip, obtained from step 1, using a spatial regularization filter, 3) we localize the needle tip using a deep learning approach adapted from the YOLO architecture

[120]. Next, we describe how these three major processes are achieved.

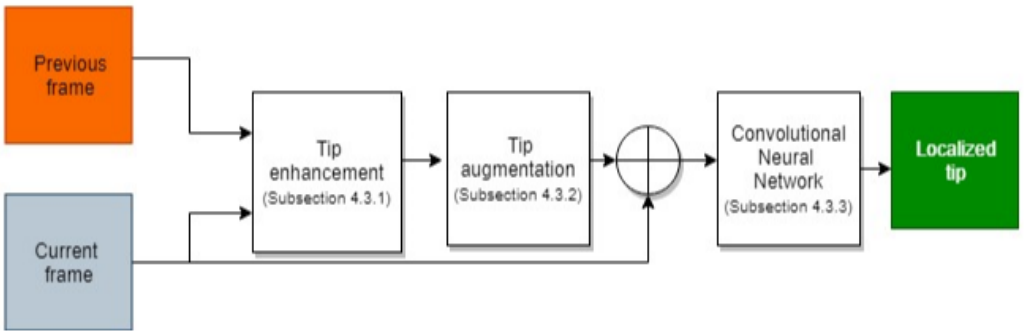


Figure 4-1. Block diagram of the proposed framework for needle tip localization from two successive US frames

4.3.1 Tip enhancement model

Consider a US frame sequence with temporal continuity, represented by the function $p(x, y, t)$, where t denotes the position in the time sequence and $(x, y)$are the spatial coordinates. We propose a dynamic background subtraction model which quickly adapts to changes in the US scene based on logical differencing between adjacent frames. For the first frame, the background is denoted as: $b(x, y, t_0) = p(x, y, t_0)$. For all subsequent frames, the background is modeled as the previous frame in the sequence i.e. $b(x, y, t_n) = p(x, y, t_{n-1})$. We then determine the bitwise complement of the background image. Considering only spatial variation, for $b(x, y) = (x, y)|b(x, y) \neq 0$, the complement is $b^c(x, y) = (x, y) \in \mathbb{Z}^2|(x, y) \notin b(x, y)$. For an 8-bit image, the complement of each pixel (an unsigned integer) is equal to itself subtracted from 255. For any current frame $p(x, y)$, the needle enhanced image is given by:

$$q(x, y) = b^c(x, y) \wedge p(x, y), \qquad (4.1)$$

where ∧ denotes the point-wise AND logical operation. (4.1) yields only the objects in the US data that moved between two successive frames, and thus gives an enhanced current tip location. Although it is plausible that tissue surrounding the needle tip moves concurrently, we consider collocated motion of the tissue and tip to be more significant than any other motion. This assumption usually holds during percutaneous needle procedures such as biopsies and regional anesthesia. Depending on the needle visibility profile, $q(x, y)$ may also contain shaft pixels.

4.3.2 Needle tip augmentation

The output of (4.1), $q(x, y)$, may contain artifacts unrelated to the needle. These may be caused by brightness variations, motion artifacts and speckle. We need to further enhance $q(x, y)$ to minimize the effect of this noise. This step is crucial before the employment of the deep learning framework explained in section 4.3.3. Without it, our model may attempt to over-fit the noise at the expense of needle features. Therefore, we first devise means of denoising $q(x, y)$. First, $q(x, y)$ is passed through a median filter with an 8×8 kernel. We denote the resulting image as $r(x, y)$. While speckle noise is multiplicative, we formulate an additive noise model to aggregate the effect of speckle, motion artifacts and any other stochastic or deterministic noise sources: $r(x, y) = e(x, y) + n(x, y)$, i.e. a sum of two components; the desired image $e(x, y)$ and the aggregate noise, $n(x, y)$. We consider $e(x, y)$ to be a function of bounded variation. Going forward, we will adopt a notation where the images are represented by vectors. The image restoration model becomes:

$$\mathbf{r} = \mathbf{e} + \mathbf{n}, \tag{4.2}$$

where $\mathbf{e} \in \mathbb{R}^{mn\times 1}$ is the desired augmented needle tip image (of size $m \times n$), $\mathbf{r} \in \mathbb{R}^{mn\times 1}$ is the corrupted image obtained from the previous step, while $\mathbf{n} \in \mathbb{R}^{mn\times 1}$ is the noise. In this notation, $\mathbf{r}$, e and n are vectors containing all the pixel values in the respective image matrices in lexicographic order. Conceptually, this problem necessitates recovering low-rank matrices from under-sampled measurements, and it can be solved using Total Variation (TV) based methods [144,145]. Problems of this nature are ill-conditioned and solving them directly is difficult due to noise sensitivity. Since pixels in the segmented image have spurious detail and possibly high TV, we formulate a TV regularization (TVR) problem of the form:

$$\min_{\mathbf{e}} \frac{\lambda}{2}\|\mathbf{r}-\mathbf{e}\|_2^2 + \|\mathbf{e}\|_{TV}, \tag{4.3}$$

where λ is a regularization parameter and $\|\mathbf{e}\|_{TV} = \|\boldsymbol{D_x}\mathbf{e}\|_1 + \left\|\boldsymbol{D_y}\mathbf{e}\right\|_1$ is the anisotropic TV norm, defined by $\boldsymbol{D_x}$ and $\boldsymbol{D_y}$, the spatial first-order forward finite difference operators along the horizontal and vertical directions respectively. (4.3) is a constrained formulation of a non-differentiable optimization problem. This problem can be efficiently solved with the Split Bregman approach [146], in which the main problem is reduced to a sequence of unconstrained optimization problems and variable updates. We first transform (4.3) into a constrained equivalent problem by introducing intermediate variables $\mathbf{v}$ and $\mathbf{w}$, i.e.:

$$\min_{\mathbf{v},\mathbf{w},\mathbf{e}} \frac{\lambda}{2}\|\mathbf{r}-\mathbf{e}\|_2^2 + \|\mathbf{v}\|_1 + \|\mathbf{w}\|_1$$

$$\text{subject to } \mathbf{v} = \boldsymbol{D_x}\mathbf{e} \tag{4.4}$$

$$\mathbf{w} = \boldsymbol{D_y}\mathbf{e}$$

The formulation in (4.4) can be converted into an unconstrained convex optimization problem (4.5) by use of augmented Lagrangian and Split Bregman techniques [146], where the constraints in (4.4) are weakly enforced by introducing quadratic penalties:

$$\min_{\mathbf{v},\mathbf{w},\mathbf{e}} \frac{\lambda}{2}\|\mathbf{r}-\mathbf{e}\|_2^2 + \|\mathbf{v}\|_1 + \|\mathbf{w}\|_1 + \frac{\nu}{2}\|\mathbf{v}-\boldsymbol{D_x}\mathbf{e}-\boldsymbol{b_1}\|_2^2 + \frac{\nu}{2}\left\|\mathbf{w}-\boldsymbol{D_y}\mathbf{e}-\boldsymbol{b_2}\right\|_2^2, \quad 4.5)$$

where ν is an additional regularization parameter, and $\boldsymbol{b_1}$ and $\boldsymbol{b_2}$ are Bregman relaxation variables which are determined through Bregman iteration. Inclusion of the last two augmented Lagrangian terms in (4.5) improves algorithm robustness since we don't have to strictly reinforce the equality constraint. (4.5) can be split into three sub-problems, solved by fixing one variable and minimizing over the other in turn:

$$\min_{\mathbf{v}} \|\mathbf{v}\|_1 + \frac{\nu}{2}\|\mathbf{v}-\boldsymbol{D_x}\mathbf{e}-\boldsymbol{b_1}\|_2^2 \tag{4.6}$$

$$\min_{\mathbf{w}} \|\mathbf{w}\|_1 + \frac{\nu}{2}\left\|\mathbf{w}-\boldsymbol{D_y}\mathbf{e}-\boldsymbol{b_2}\right\|_2^2 \tag{4.7}$$

$$\min_{\mathbf{e}} \frac{\lambda}{2}\|\mathbf{r}-\mathbf{e}\|_2^2 + \frac{\nu}{2}\|\mathbf{v}-\boldsymbol{D_x}\mathbf{e}-\boldsymbol{b_1}\|_2^2 + \frac{\nu}{2}\left\|\mathbf{w}-\boldsymbol{D_y}\mathbf{e}-\boldsymbol{b_2}\right\|_2^2 \tag{4.8}$$

(4.6) and (4.7) decouple over space and have closed-form solutions as vectorial shrinkages (soft thresholding):

$$\mathbf{v} = sign(\boldsymbol{D_x}\mathbf{e}+\boldsymbol{b_1}) \times max\{|\boldsymbol{D_x}\mathbf{e}+\boldsymbol{b_1}| - \frac{1}{\nu}, 0\}$$

$$\mathbf{w} = sign\left(\boldsymbol{D_y}\mathbf{e}+\boldsymbol{b_2}\right) \times \max\left\{\left|\boldsymbol{D_y}\mathbf{e}+\boldsymbol{b_2}\right| - \tfrac{1}{\nu}, 0\right\} \tag{4.9}$$

(4.8) is a simple least square problem (Tikhonov regularization) which can be solved analytically using a gradient descent algorithm. First, we derive the pertinent normal equation:

$$\mathbf{e}\left[\lambda\mathbf{I} - \nu\left\{\boldsymbol{D}_{\boldsymbol{x}}^{T}\boldsymbol{D}_{\boldsymbol{x}} + \boldsymbol{D}_{\boldsymbol{y}}^{T}\boldsymbol{D}_{\boldsymbol{y}}\right\}\right] = \lambda\mathbf{r} + \nu\boldsymbol{D}_{\boldsymbol{x}}^{T}(\mathbf{v} - \boldsymbol{b}_{\mathbf{1}}) + \nu\boldsymbol{D}_{\boldsymbol{y}}^{T}(\mathbf{w} - \boldsymbol{b}_{\mathbf{2}}) \qquad (4.10)$$

(4.10) is solved using LSMR [147], an iterative least squares solver. $\boldsymbol{b}_{\mathbf{1}}$ and $\boldsymbol{b}_{\mathbf{2}}$ are initialized to zero and updated between every consecutive iteration of the sub-problems: $\boldsymbol{b}_{\mathbf{1}}{}^{i+1} = \boldsymbol{b}_{\mathbf{1}}{}^{i} + \boldsymbol{D}_{\boldsymbol{x}}\mathbf{e} - \mathbf{v}$, $\boldsymbol{b}_{\mathbf{2}}{}^{i+1} = \boldsymbol{b}_{\mathbf{2}}{}^{i} + \boldsymbol{D}_{\boldsymbol{y}}\mathbf{e} - \mathbf{w}$. The enhancement process is summarized in Algorithm 1. Figures 4-2 illustrates the result of needle tip augmentation for in-plane and out-of-plane needles.

Algorithm 1: Algorithm for needle tip enhancement

Result: Enhanced needle tip image, $e(x, y)$

input data $\mathbf{r}$

input parameters λ and ν

initialize $\mathbf{e} = \mathbf{r},\ \mathbf{v} = 0, \mathbf{w} = 0,\ \boldsymbol{b}_{\mathbf{1}} = 0; \boldsymbol{b}_{\mathbf{2}} = 0$

compute $\boldsymbol{D}_{\boldsymbol{x}}$ and $\boldsymbol{D}_{\boldsymbol{y}}$

while $\left\| \mathbf{e}_{\boldsymbol{current}} - \mathbf{e}_{\boldsymbol{previous}} \right\|_2 > 10^{-5}$ **do**

    solve the $\mathbf{v}$ sub-problem using (9)

    solve the $\mathbf{w}$ sub-problem using (9)

    solve the $\mathbf{e}$ sub-problem using (10)

    update $\boldsymbol{b}_{\mathbf{1}}$ and $\boldsymbol{b}_{\mathbf{2}}$

end

4.3.3 Needle tip detection

We have achieved a needle tip enhanced image $e(x, y)$ in which the tip exhibits a high intensity. However, we still need to localize the tip. Often, the needle tip will not move in each US frame since needle actuation speed may not match the US frame rate. Insignificant changes in needle motion will lead to insignificant tip intensity in $r(x, y)$. Therefore, we need to identify frames in which no significant motion has occurred. Further, despite the prior enhancement process, there high intensity interfering artifacts not associated with needle motion such as from motion of tissues/organs could persist. Hence, we cannot rely on the tip to always exhibit the highest intensity in $e(x, y)$. Consequently, we formulate a deep learning framework for efficient needle tip detection. Next, we describe elements of the deep learning framework that are unique to our method.

***CNN architecture*:** The proposed deep learning framework is shown in Figure 4-3, and is built based on YOLO [120], a state-of-the-art single shot object detection CNN architecture. The framework outputs 2D bounding box predictions consisting of 5 components: x, y, w, h and η, where (x, y) coordinates represent the center of the box, w and h are the width and height respectively, and η is the confidence that the box contains an object and that the object is the needle tip. The new framework consists of a 256×256 image input layer. To further reduce computational complexity toward real-time performance, we use only 8 convolutional layers. We implement a pixel-level fusion layer in which the current US image $p(x, y)$ and its tip enhanced counterpart $e(x, y)$ are concatenated before inputting to the CNN.

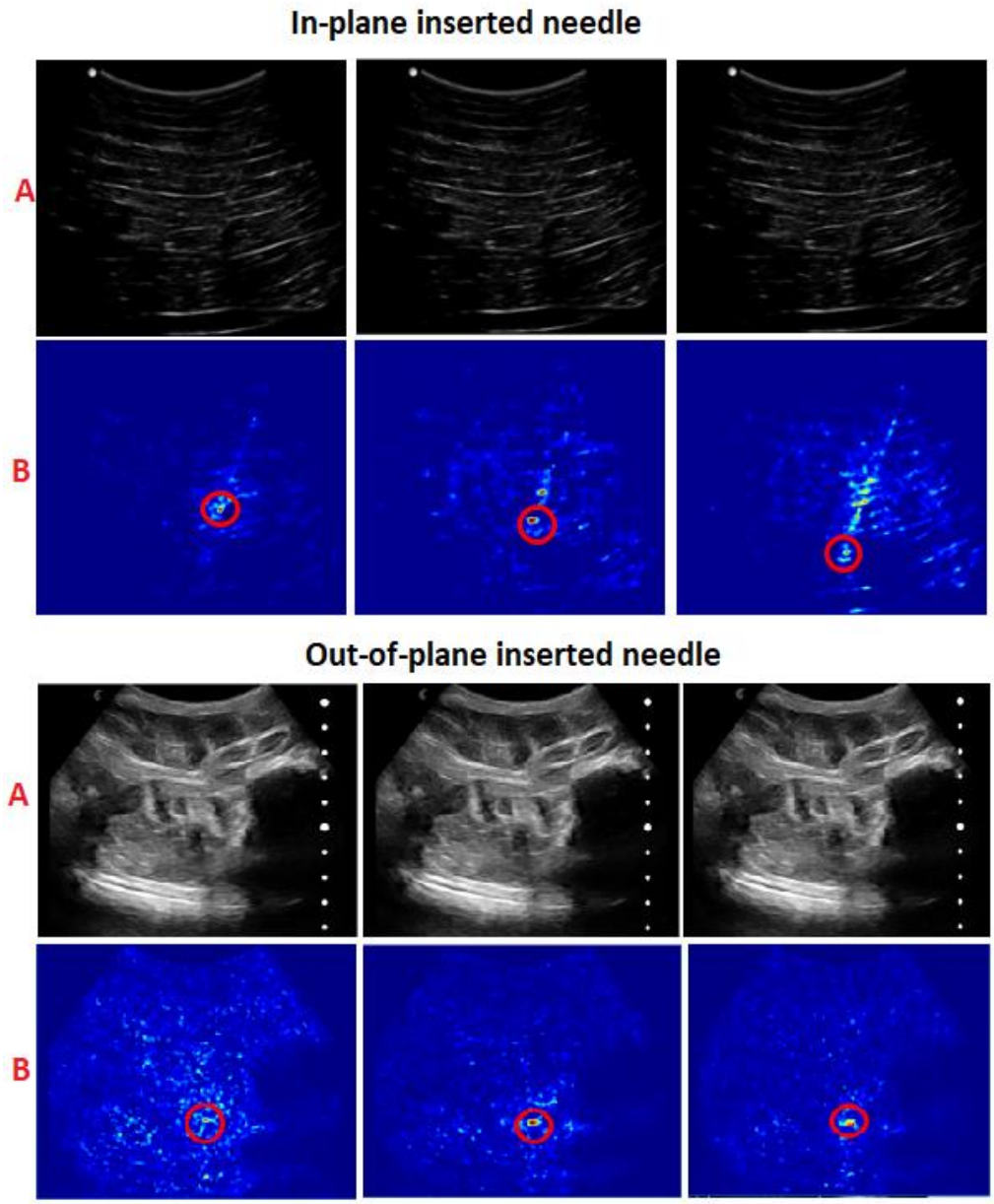


Figure 4-2. Needle augmentation in consecutive frames with in-plane insertion of a 17G needle in a bovine tissue phantom and out-of-plane insertion of a 17G needle in a porcine shoulder phantom. (A) original images before tip enhancement and augmentation. Identifying the needle in these images is difficult. (B) tip augmented image $e(x, y)$ (color coded). Circle surrounds the augmented tip. The proposed method achieves accurate enhancement of the tip despite low tip intensity in the original image or presence of high intensity artifacts.

Since the needle tip is a fine-grained feature, we configure the convolution layers to maintain spatial dimensions of the respective inputs, thus mitigating reduction in resolution. More so, CNN neurons at deeper layers always have large receptive fields that will ensure incorporation of image-level context pertinent to needle tip appearance.

Uniquely, each of the first 7 convolution layers is followed by an exponential linear unit (ELU) [99] with $\alpha = 0.5$. The use of the ELU non-linearity is advantageous because it makes activations close to zero mean and unit variance to always converge towards zero mean and unit variance even under the presence of noise and perturbations. In Section 4.5, we will present comparative analysis of the proposed model's performance with and without ELU. The first 5 convolution layers are followed by a 2×2 max pooling layer with a stride of 2. All the other physical attributes of the original YOLO architecture [120] are unchanged.

At test time, the model is malleable to any input size. Two advantages accrue from treating our challenge as a detection problem. Inherently, needle tip features will be learned end-to-end, thus eliminating the need to explicitly encode them. It is expected that frames where no needle tip has moved will exhibit no detectable features, while the learned model will accurately extract the tip when it is present.

***Training details*:** The model is initialized with weights derived from training on the PASCAL VOC dataset [148]. The ground truth bounding box labels are defined using an electromagnetic tracking system and an expert sonographer: The ground truth tip localization is used as the center of the bounding box (x, y) and the

thickness $w \times h$ is chosen to be at most 20×20 pixels in all images. We use an initial learning rate of 10^{-4}, a batch size of 4 and train for 60 epochs. Our choice of optimizer is Adam.

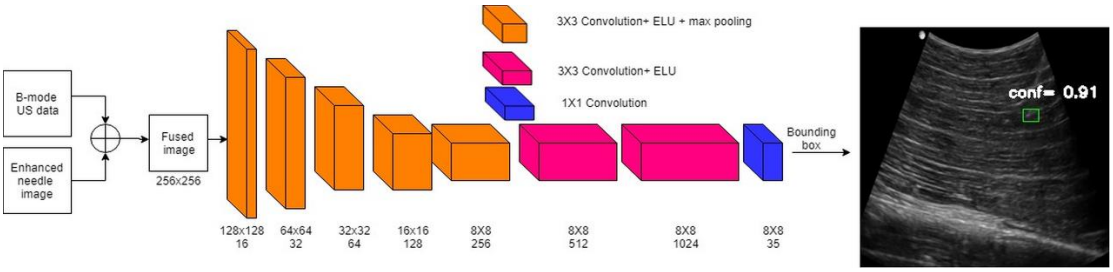


Figure 4-3. Block diagram of the needle tip detection CNN architecture. In the output, the needle tip is enclosed in a bounding box (green) annotated with a confidence score; a measure of classification and localization accuracy.

4.4 Data acquisition and experimental validation

To train and evaluate our model, we collected a dataset of 2D B-mode US images using materials and settings specified in Table 4-1. Two imaging systems: SonixGPS (Analogic Corporation, Peabody, MA, USA) with a hand-held C5-2/60 curvilinear probe, and 2D hand-held wireless US (Clarius C3, Clarius Mobile Health Corporation, Burnaby, BC, Canada) were used. Experiments were performed on a freshly excised bovine tissue, and a porcine shoulder phantom, with insertion of a 17G (1.5 mm diameter, 90 mm length) Tuohy epidural needle (Arrow International, Reading, PA, USA), a 17G SonixGPS vascular access needle (Analogic Corporation, Peabody, MA, USA) and a 22G spinal Quincke-type needle (Becton, Dickinson and Company, Franklin Lakes, NJ, USA). In all our experiments, the probe was hand-held. Noisy motion not resulting from insertion

of the needle was simulated by manually pressing the probe against the imaging medium and rotating it slightly about its long axis. With the SonixGPS needle, we collected ground truth needle tip localization data using an electromagnetic tracking system (Ascension Technology Corporation, Shelburne, VT, USA). In-plane insertion was performed at $40^{\circ} - 70^{\circ}$ and the needle was inserted up a depth of 70 mm. 40 (20 in-plane, 20 out-of-plane) sequences of US images, each containing more than 400 frames were collected.

Performance of the proposed method was evaluated by comparing the automatically detected tip location (center of the detected bounding box) to the ground truth determined from the electromagnetic tracking system for data collected with the SonixGPS needle. For data collected with needles without tracking capability, the ground truth was determined by an expert sonographer. Tip localization accuracy was determined from the Euclidean distance between the corresponding measurements.

We implemented our methods on an NVIDIA GeForce GTX 1060 6GB GPU, 3.6 GHz Intel(R) Core^{TM} i7 16GB CPU Windows PC. The needle tip enhancement and augmentation methods were implemented in MATLAB 2018a. For the sub-problems in (4.9) and (4.10), we empirically determined $\nu = 2$ and $\lambda = 5$ as optimum values. Throughout the validation experiments, these values were not changed. The tip detection framework was implemented in Keras. 5000 images from 20 video sequences were used for training, while 1000 images from 10 other sequences were used for validation. Lastly, 500 images from 10 sequences not used in training or validation were used for testing.

Table 4-1. Materials and experimental settings for 2D US data collection. IP=in-plane insertion, OP=out-of-plane insertion.

Bovine tissue			
Imaging system	**Needle type, dimensions and insertion profile**	**# of videos**	**Pixel size**
SonixGPS	17G SonixGPS (1.5 mm,70 mm), IP 17G Tuohy (1.5 mm,70 mm), IP 22G BD(0.7 mm, 90 mm), IP 22G BD(0.7 mm, 90 mm), OP	5 5 5 5	0.17 mm 0.17 mm 0.17 mm 0.17 mm
Porcine tissue on spine phantom			
Imaging system	**Needle type, dimensions and insertion profile**	**# of videos**	**Pixel size**
Clarius C3	17G SonixGPS (1.5 mm,70 mm), IP 17G Tuohy (1.5 mm,70 mm), IP 22G BD(0.7 mm, 90 mm), IP 22G BD(0.7 mm, 90 mm), OP	5 5 5 5	0.24 mm 0.24 mm 0.24 mm 0.24 mm

4.5 Results

Qualitative Results: Figure 4-4 shows needle detection results for 4 consecutive frames for both in-plane and out-of-plane insertion. Note that the tip is accurately localized despite presence of other high intensity interfering artifacts in the B-mode US data. In this case, these features arise from partial enhancement of the shaft. The detection CNN learns to automatically identify the tip at the distal end of the point cloud in the enhanced image $e(x, y)$. For out-of-plane insertion, the temporal window for needle tip visibility is always limited, but our method can be useful for tracking small perturbations of the needle tip close to the target. Meanwhile, our method is agnostic to the type and size of needle used. For out-of-plane insertion of the 22G needle, we do not have the benefit of the electromagnetic tracking system and the tip is only discernible by jiggling. Nevertheless, our method achieves accurate localization.

Model comparison: Ablation studies, where the structural configuration of a deep learning framework is altered to assess the impact on model performance, are used to justify design choices. In line with this standard approach, we compare efficiency of our needle tip detection framework to that from alternative implementation approaches. We evaluate accuracy of detection using the mean average precision (mAP) metric on the validation dataset. mAP is calculated as the average value of the precision across a set of 11 equally spaced recall levels [149], yielding one value that depicts the shape of the precision-recall curve. Table 4-2 shows the mAP for different configurations of the detection CNN.

Table 4-2. Detection accuracy from the proposed method vs alternative approaches.

Method	mAP
Detection CNN + $p(x, y)$	0.202
Detection CNN + $e(x, y)$	0.867
Detection CNN (with leakyReLU) + $p(x, y) + e(x, y)$	0.914
Detection CNN (with ELU) + $\boldsymbol{p(x, y) + e(x, y)}$	**0.946**

First, we examine performance of the proposed CNN with the raw US image $p(x, y)$ as an input. As expected, the detection efficiency is very low $(20.2\ \%)$. This is because without our tip enhancement algorithm, tip features are barely discernible and are overshadowed by other high intensity artifacts in the cluttered US image. We also consider only the enhanced image $e(x, y)$ as the input. A high mAP of $86.7\ \%$ is achieved, showing that our enhancement algorithm is efficient. Furthermore, we show that fusion of $e(x, y)$ and $p(x, y)$ achieves the highest mAP of $94.6\ \%$.

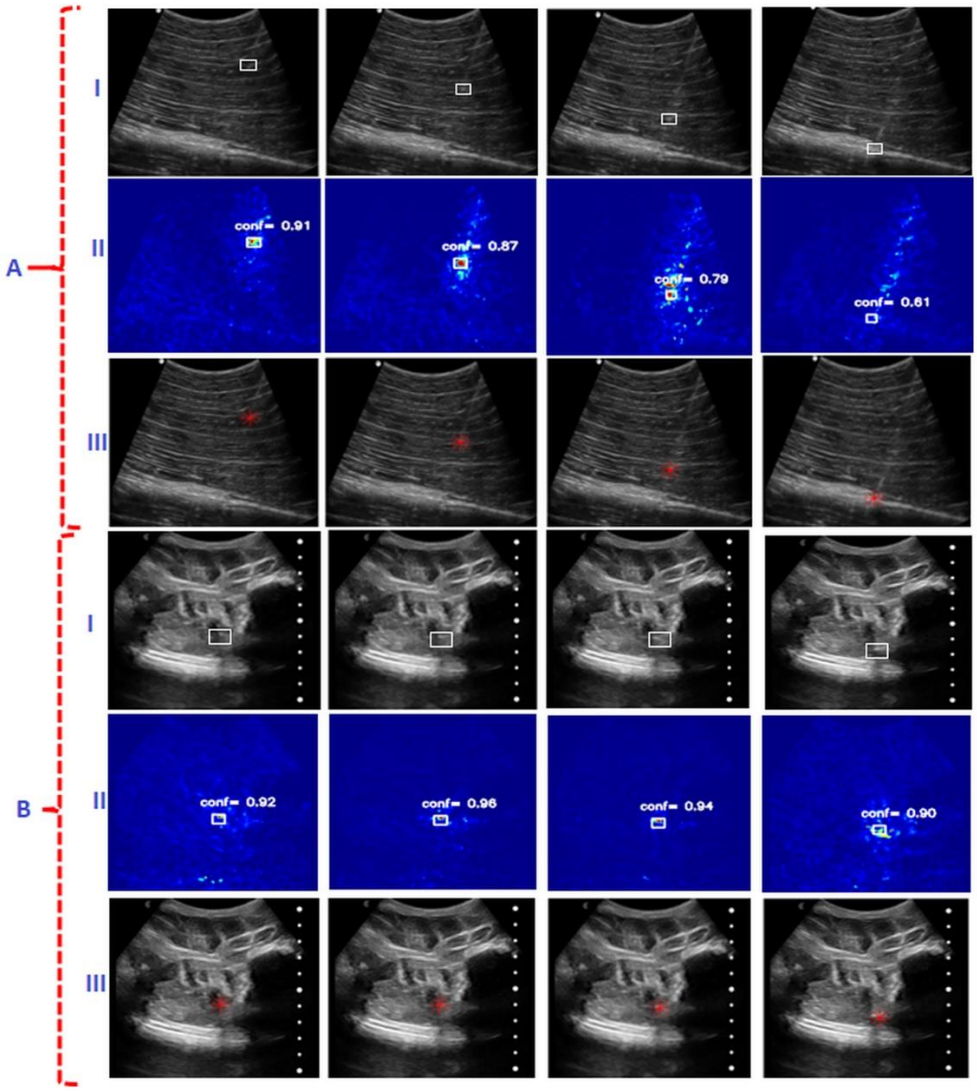


Figure 4-4. Needle detection and localization in 4 consecutive frames with (A) in-plane insertion of the 17G SonixGPS needle into bovine tissue and (B) out-of-plane insertion of the 22G needle into porcine shoulder phantom. (I) Original image. The white box is the annotated ground truth label, determined with an electromagnetic tracking system for (A) and an expert sonographer for (B). (II) Detection result with bounding box (white) overlayed on enhanced image $e(x, y)$. The inset number alongside the box annotation is the detection confidence. (III) Localized tip, the center of the detected bounding box (red star) overlayed on original image. Our method achieves high detection and localization accuracy.

With the fusion input, and with other hyperparameters maintained constant, we compare performance of the proposed method against a similar model with leakyReLU activation layers [120]. The proposed method outperforms this configuration. Worthy to mention, we chose a batch size of 4 in all our experiments because of the memory constraints of the GPU. It is expected that a bigger batch size would have resulted in an even higher mAP from the proposed model.

Runtime performance*:* On the NVIDIA GeForce GTX 1060 GPU, our framework runs at 0.094 ± 0.01 s per frame (0.014 s for enhancement, 0.06 for augmentation and 0.02 s for detection). This is about 10 fps, and to the best of our knowledge, the fastest needle tip localization framework reported so far. Certainly, the processing speed can be increased with more computing resources. In frames where the needle tip is somehow visible, the augmentation step is unnecessary, and the runtime speed increases to 29 fps.

Mitigating false detections: Since YOLO is a multi-object detection framework, it is possible that several bounding boxes with different confidence scores can be detected on the same input image. We sought to minimize these false positives by selecting the bounding box with the highest confidence score and using a hard-threshold of 0.35 for the score, a value which was empirically determined and kept constant throughout validation. In so doing, bounding boxes completely skewed from the needle trajectory were eliminated. It is expected that the robustness of tip detection would further be improved if a bigger training dataset was used.

Tip localization accuracy: Overall, the tip localization error was 0.83 ± 0.02 mm. Direct and fair comparison to other works is difficult. Most methods in prior art depend on initial visibility of the needle in the US image, usually as a line-like structure. Our method does not require initial needle visibility, and therefore, the data set chosen for its evaluation would not suit these other methods. The closest to our work is [71] which reports a similar localization accuracy (0.82 mm). However, their computation time of 1.18 s per frame (~1 fps) is significantly slower than our real-time approach (10 fps).

We compared the proposed method against the method in Chapter 3 [150] by evaluating the two on the same set of 200 randomly selected US images with only in-plane needle insertion. The results are shown in Table 4-3. Note that the proposed method outperforms the previous method in in both tip localization accuracy and computational efficiency. A one-tailed paired t-test shows that the difference between the localization errors from the proposed method and the previous method is statistically significant $(p < 0.005)$. The localization accuracy obtained from the previous method is worse than previously reported because we used a more challenging dynamic dataset with very low shaft intensity.

Table 4-3. Comparing tip localization accuracy from the proposed method vs the method in Chapter 3 for only in-plane insertions.

Method	localization error (mm)	Overall processing time (s)
Proposed method (with detection CNN)	0.89 ± 0.05	0.092
Method in Chapter 3	1.08 ± 0.44	0.56

We also sought to ascertain how the proposed method would have performed if we had relied on only the needle tip enhancement and augmentation phase (without the needle detection step). In this case, a possible route is determining needle localization from the maximum intensity in $e(x, y)$. The results are shown in Table 4-4 and demonstrate that the localization accuracy is worse without the detection framework. This is expected because without the benefit of implicitly learning heuristic features associated with the tip via deep learning, there is a higher likelihood of localizing artifacts with similar intensity to the tip.

Table 4-4. Comparing tip localization errors for the proposed method with and without the detection framework. Results are presented as 95% confidence interval of the mean.

Localization approach	Ground truth	# of images	error (mm)
Proposed method (with detection CNN)	Tracking system	250	0.76 ± 0.03
Proposed method (with detection CNN)	Expert	250	0.89 ± 0.04
Proposed method (with detection CNN) (overall)		**500**	$\mathbf{0.83 \pm 0.02}$
Highest intensity in $e(x, y)$ (without detection CNN)	Tracking system	250	0.94 ± 0.04
Highest intensity in $e(x, y)$ (without detection CNN)	Expert	250	1.23 ± 0.05
Highest intensity in $e(x, y)$ (without detection CNN) (overall)		**500**	$\mathbf{1.09 \pm 0.04}$

4.6 Discussion

We have demonstrated a novel approach for needle tip localization in 2D US, suitable for challenging imaging scenarios in which the needle is not continuously visible. The main strength of our work is in the robust and accurate tip localization at a close to real-time processing rate of 10 fps. This is better than state-of-the-art. The proposed method does not require the needle to appear as a high intensity,

continuous linear structure in the US image. Therefore, both in-plane and out-of-plane needle localization are achieved. We used the thinner 22G needle in our experiments to demonstrate the robustness of our method. Typically, such thin needles are prone to bending and the shaft has limited visibility, but this did not affect the accuracy of tip localization. Therefore, it is possible that our method can localize bending needles. However, we will further investigate this in our future work.

The detection component in our method mitigates motion artifacts that may arise from physiological activity and minor probe reorientation. Generally, any method reliant on motion detection is prone to drastic motion between consecutive frames, for example due to abrupt changes in probe alignment or rapid physiological motion, such as during transseptal punctures in a beating heart. Therefore, the proposed method would not be efficient in these scenarios. Despite these limitations, our method is a step forward toward a real-time image processing-based method for needle localization in 2D US.

Chapter 5 -Single shot localization of invisible needle in 2D US

5.1 Overview

In Chapter 4, we proposed a learning-based framework for localization of the needle tip in scenarios where the tip has a low intensity and the shaft is imperceptible. Our approach utilized a novel algorithm which extracts tip information from digital subtraction of consecutive frames. The tip feature is then augmented and localized using YOLO [120], a state-of-the-art detection architecture. YOLO produces bounding boxes surrounding the detected feature. Therefore, we estimate the needle tip location as the center of these bounding boxes. The overall processing speed achieved, 10 fps, is good but can be improved.

In this Chapter, we investigate the following question: what if instead of using an architecture which outputs bounding boxes, we utilized a network that directly outputs the location of the needle tip from the digital subtracted imaged? We expect that this route, if feasible, would be faster than the approach that requires needle tip argumentation and tip detection.

To test this hypothesis, we developed a novel framework consisting of a cascade of two twin CNNs: a needle tip classifier and a needle tip location regressor. The classification CNN determines whether a tip exists in a tip enhanced image obtained from digital subtraction of two consecutive frames in a

US sequence. The regressor then directly outputs the spatial coordinates of the tip. Our approach is trained and evaluated on an ex vivo dataset collected with 17G and 22G needles inserted in-plane and out-of-plane in bovine and porcine phantoms. We use 9000 images extracted from 30 video sequences for training and validation, and 200 images from 10 sequences for testing. The framework achieves a tip localization error of 0.44 ± 0.06 mm, an overall localization time of 0.006 s (166 fps) and an overall processing time of 0.015s (67 fps). Because of the fast execution time and accurate tip localization, we believe that it is potentially a breakthrough for real-time needle tip localization.

5.2 Background

In this section, we give theoretical context to regression neural networks and their application in computer vision tasks. Unlike classification tasks (mentioned in Chapter 4) where the objective is to output a categorical label, regression networks output a continuous number. This makes them suitable for predicting the location of points of interest in an image.

5.2.1 From linear regression to deep neural networks

Consider statistical problem with n-dimensional input x, and a set of target responses y. Linear regression modeling seeks to define a mapping of the inputs to the targets using a linear function:

$$y = \theta^T x + \theta_o + \xi, \quad \xi \sim \mathcal{N}(0, \sigma^2) \tag{5.1}$$

Here, θ defines the regression coefficients (weights) and θ_o is the bias. It is assumed that the outputs are corrupted by Gaussian noise of unknown variance

σ^2. A neater way of generalizing (5.1) to cater for problems where the distribution of the targets doesn't follow a Gaussian distribution is by combining the bias and the regression coefficients:

$$y = \theta^T x$$

$$\mathbb{E}[y] = \mu = f^{-1}(y) \tag{5.2}$$

5.2 represents a Generalized Linear Model (GLM). The GLM includes a link function $f(.)$ which relates the mean value of the model's outcome (response) to the weighted combination of inputs.

The similarity of 5.2 to 3.1 is rather obvious. A deep neural network implements regression at each layer, with $f(.)$ defining the activation function while θ are the weights. In a FF neural network, these layers are repeated in a recursive GLM paradigm. Therefore, deep neural networks can be used to efficiently solve regression problems. Further, when we are dealing with regression problems related to image processing tasks, CNNs come in handy.

The solution of a regression problem with a deep neural network follows the same principles as we saw before. We have a set of images (inputs) and a set of target responses (labels). We define a loss function $\mathcal{L}$, basically a mechanism of determining how far the current prediction is from the ground truth. If the prediction deviates a lot from the expected result, the loss function gives a very large output and vice versa. During training, we feed our network with training examples. Using the loss function, we determine the loss over all training data for values of the

weights. Then using gradient descent, the weights are iteratively optimized until the overall loss is minimized.

5.2.2 Classification and regression loss

Loss functions come in many flavors. For classification tasks (where we want to know if an object belongs to a certain category out of a finite set of categorical values), the Log Loss (Cross Entropy Loss) is commonly used. The Log Loss is defined as:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N} y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)) \quad (5.3)$$

Here, y_i is the label and $\hat{y}_i = p(y_i)$ is the predicted probability. The overall network would then consist of convolution layers (interspersed with activation and pooling layers), fully connected layers, and lastly, Softmax activation and the Log Loss to implement the classification layer.

A softmax function normalizes the output of each node in the last fully connected layer to $\mathcal{Y}: [0,1]$, while also making the sum of all the units equal to 1. Here, $\mathcal{Y}$ is the output vector of length N, and the Softmax output becomes:

$$\sigma(\mathcal{Y})_j = \frac{e^{\mathcal{Y}_j}}{\sum_{i=1}^{N} e^{\mathcal{Y}_i}} \quad (5.4)$$

The outputs of Softmax activation are then assigned as probabilities to each categorical label, and the loss is computed. This facilitates multi-class classification. For binary classification, Sigmoid activation can be used.

For regression tasks, the desired output is a continuous number. The most commonly used loss function for regression tasks is the Mean Squared Error (MSE) or L_2 loss:

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}(y_i - f(x_i))^2 \quad (5.5)$$

Here, we are taking the mean of the square of the differences between the true values y_i and predicted values $f(x_i)$. Other loss functions are the Mean Absolute Error (L_1), the Smooth Mean Absolute Error (Huber), the Log-Cosh and the Quantile loss. The overall network is like the classification network, without the classification layer: Instead, we have a fully connected regression layer with linear or sigmoid activations. Application of such CNNs with a linear regression top layer is sometimes called vanilla deep regression.

In Chapter 3 and Chapter 4, we proposed methods for needle localization that utilize detection learned models. The detection problem incorporates both classification (we want to know if an object exists) and localization (if it exists, show me where it is). No wonder, the loss function used for our detection network (3.5) is a combination of a classification and a regression loss.

5.2.3 State of the art regression models

Regression neural networks employing CNNs have become the de facto standard for solving a variety of problems in computer vision, such as camera pose estimation [151,152], 6D object pose estimation [153], human pose estimation [154,155,156] and facial landmark detection [157]. In medical imaging, regression

networks have been applied in instrument pose estimation [158] and image registration [159,160,161,162].

Unfortunately, most of the regression approaches employ methods that are custom to the dataset, and various authors rarely justify their methods of choice. It is therefore not easy to adapt one approach to similar problems. Concerning needle tip localization, the closest prior art is that for keypoint localization [157], where the objective is to find points of interest in an image.

In the sections that follow, we propose a novel approach for needle tip localization in 2D US using a cascade of a classification and regression neural network. The classification network is used to predict whether a needle tip exists in an image. We treat the needle tip as a landmark which we define using keypoints. Using a regression approach instead of the detection approach described in Chapter 4 ensures that we need a single pass through the cascade FF network to obtain a localization result. Our main contributions are: (a) a novel framework consisting of a cascade of a classification and a regression CNN for learning needle tip descriptors in 2D US, (b) a single shot approach for extracting the needle tip using the regression network. The proposed method outperforms state-of-the-art methods in computational speed. Next, we describe our approach in detail.

5.3 Methods

Our objective is a framework that automatically localizes the needle tip in a time series of US frames and is robust to intensity variations between images

captured under different imaging scenarios and needle insertion profiles. Like the approach of Chapter 4, the proposed methods targets both out-of-plane and in-plane inserted needles in which the tip exhibits low contrast to the rest of the US image. Consider a 2D US image $US(x, y)$ where the needle tip is at a spatial location $T(x, y)$. The needle localization task involves estimating the posterior distribution $p(T(x, y)|US(x, y))$. The proposed approach consists of three stages illustrated in Figure 5-1: a) enhancement of the needle tip from 2 consecutive US frames, b) determining whether the enhanced image contains substantial needle tip information, using a classifier c) estimating $p(T(x, y)|US(x, y))$ and hence the needle tip location using a keypoint regression CNN. The classification step is necessary because the frame rate rarely matches the insertion speed of the needle: the needle tip may not move smoothly through space. Therefore, there are bound to be frames in the sequence where the needle tip has not changed spatial location. Next, we describe these processes in detail.

5.3.1 Needle tip enhancement

The needle enhancement process follows the paradigm introduced in the Section 4.3.1. From a US frame sequence $F_t, F_{t-1}, \ldots\ldots F_1, F_t, F_0$ where t denotes the respective temporal position and F_0 is the first frame, we compute $z(x, y)$, the logical difference of consecutive frames: For a current frame $n(x, y)$ and an immediate previous frame $m(x, y)$,

$$z(x, y) = m^c(x, y) \wedge n(x, y) \tag{5.6}$$

Here, $m^c(x,y)$ is the bitwise complement of $m(x,y)$, while ∧ is the bitwise AND logical operation. We need to reiterate here that these logical operations are conducted on images whose pixel intensity is an 8-bit unsigned integer. For an intensity $\mathcal{T}(x_i,y_j)$, the complement is 255- $\mathcal{T}(x_i,y_j)$.

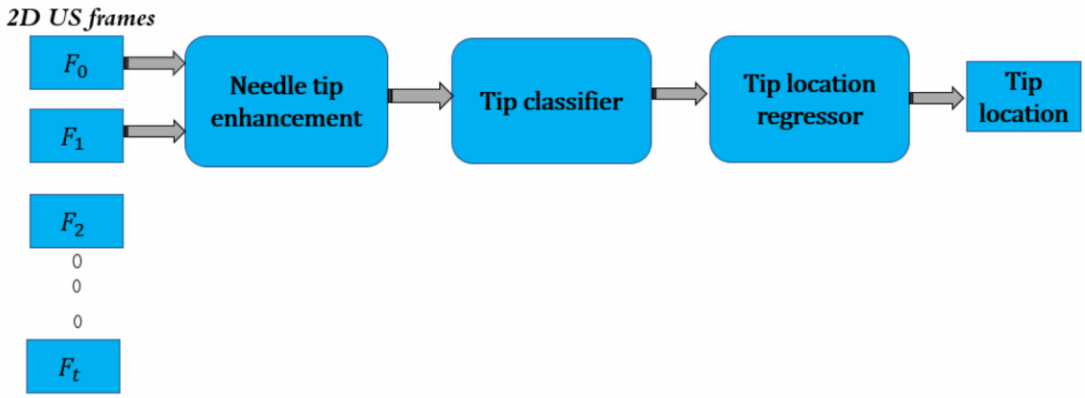


Figure 5-1. Block diagram of the proposed method. F_0 to F_t are sequential US frames.

The bitwise AND operation will also produce intensities $\mathcal{T}$: [0,255]. Therefore, the output image is not a binary image, but consists of a range of intensities representative of the difference between the two frames. Any subtle motion that occurred between the two frames will thus be captured. It is expected that the needle tip will yield the most salient feature in $z(x,y)$. To remove noisy artifacts and enhance the needle tip, $z(x,y)$ is filtered with a 12×12 median filter. Different than the method in Chapter 4, with the aim of saving computational cost, we do not perform any further needle augmentation. The use of a larger kernel median filter (the method in Chapter used an 8×8 kernel) compensates for this step. Figure 5-2 shows the result of the enhancement process.

5.3.2 Network Architecture

The architecture of our classification + regression network is shown in Figure 5-3. The network consists of 6 blocks of convolution, ReLU, batch normalization and max pooling layers, 2 blocks of convolution, ReLU and batch normalization layers, and 3 fully connected layers. The hyperparameters of the learned layers are shown in Table 5-1. The input consists of a 256×256 image.

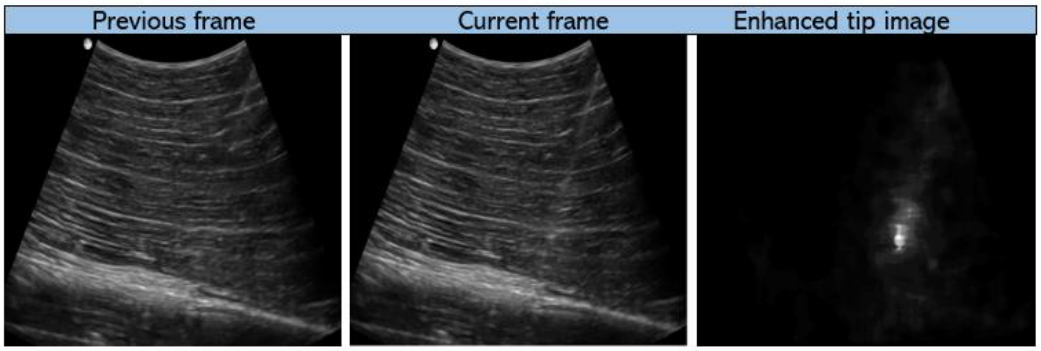


Figure 5-2. The tip enhancement process. From the current frame alone, it is hard to discern the needle tip location. However, with knowledge of the previous frame, our tip enhancement model efficiently extracts the tip.

This size is a compromise for maintaining adequate image resolution while not increasing computational complexity.

In effect, we have a cascade of two twin CNNs which share all the learned layers but differ only in the output. The classifier terminates with a 2-output fully connected layer while the regressor terminates with a 4-output fully connected layer (the needle tip is defined by 2 keypoints as we will describe later). During training and testing, the classification task and the tip location regression task are run in series.

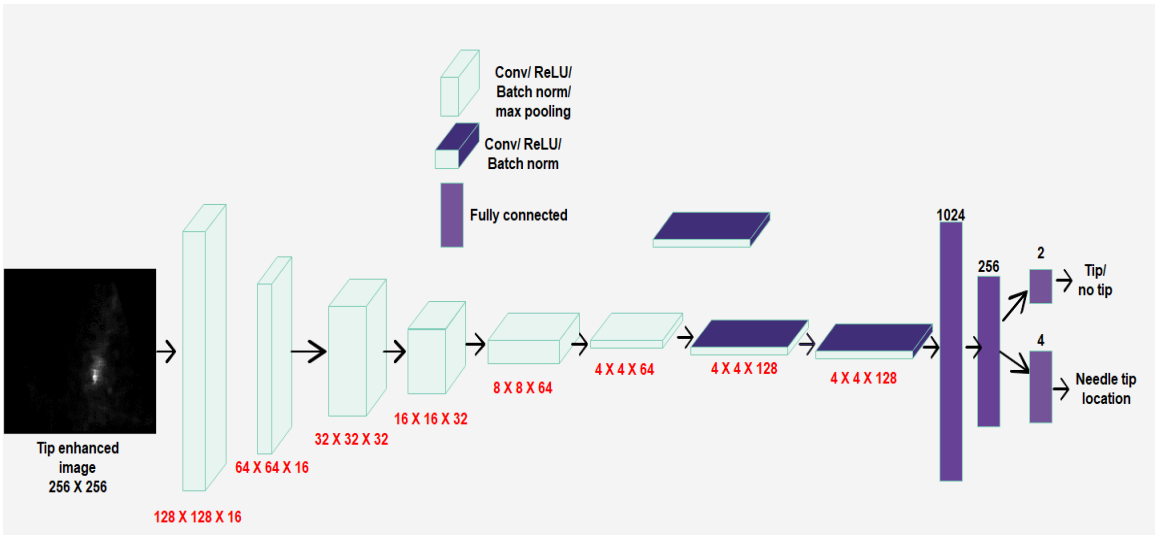


Figure 5-3. Network architecture of the proposed classification + regression network. The network consists of 6 convolution/ReLU/ batch normalization/max pooling blocks, 2 convolution/ReLU/ batch normalization blocks, 3 fully connected layers and 2 dropout layers. The tensor dimensions of the activation feature maps after each block are indicated.

Table 5-1. Hyperparameters of the learned layers. Conv=Convolution layer, FC=Fully Connected Layer.

N	Layer	Kernel	Stride	Padding	Output ($W \times H \times D$)
0	Input				256×256
1	Conv_1	3	1	1	$256 \times 256 \times 16$
5	Conv_2	3	1	1	$128 \times 128 \times 16$
9	Conv_3	3	1	1	$64 \times 64 \times 32$
13	Conv_4	3	1	1	$32 \times 32 \times 32$
17	Conv_5	3	1	1	$16 \times 16 \times 64$
21	Conv_6	3	1	1	$8 \times 8 \times 64$
25	Conv_7	3	1	1	$4 \times 4 \times 128$
28	Conv_8	3	1	1	$4 \times 4 \times 128$
31	FC_1				$1 \times 1 \times 1024$
32	FC_2				$1 \times 1 \times 256$
33	FC_3				$1 \times 1 \times 2$ (classifier) $1 \times 1 \times 4$ (tip regressor)

5.3.3 Dataset

The 2D B-mode US images we use to evaluate our framework were collected using two imaging systems: SonixGPS (Analogic Corporation, Peabody, MA, USA) with a hand-held C5-2/60 curvilinear probe, and 2D hand-held wireless US (Clarius C3, Clarius Mobile Health Corporation, Burnaby, BC, Canada). We inserted needles of types: a 17G SonixGPS vascular access needle (Analogic Corporation, Peabody, MA, USA) and a 22G spinal Quincke-type needle (Becton, Dickinson and Company, Franklin Lakes, NJ, USA) in freshly excised bovine and porcine tissue phantoms. The needles were inserted both in-plane and out-of-plane. For in-plane insertion, the needles were inserted at various angles ($30^{\circ} - 70^{\circ}$) up to a depth of 70 mm. The SonixGPS needle facilitates electromagnetic tracking, so we collected data for groundtruth needle tip location. In total, we collected 40 volumes (20 in-plane, 20 out-of-plane), with each video sequence having more than 600 frames. For the classification task, we used a total of 5000 positive examples (enhanced images with the tip) and 4000 negative examples (enhanced images without tip information). For the regression task, we used 5000 images (only positive examples) for training and validation, and for testing, we used 200 images from sequencies not used for training and validation.

5.3.4 Classifier design

In our framework, the classifier and detector networks are trained in series. For the classifier, positive and negative examples for the needle tip are labelled by an expert. Figure 5-4 and Figure 5-5 illustrate positive and negative examples. Comparing the negative and positive images, notice that the needle tip exists as a

high intensity feature, a few pixels thick, in sharp contrast to a low intensity background. Such a feature is lacking from the negative training examples. It is possible that other high intensity artifacts may exist, arising from say, change in probe position, hand tremor or in the clinical setting, physiological activity such as breathing and pulsation. We expect that our classifier can differentiate such artifacts from the needle.

The last feature map for the classification task, from the last FC layer, is a vector of dimension $1 \times 1 \times 2$ i.e., it is meant to differentiate images of two classes: those with tip and those without it. We apply Softmax activation (5.4) to this feature map and use the Log Loss (5.3) to calculate deviations between the network output and the groundtruth.

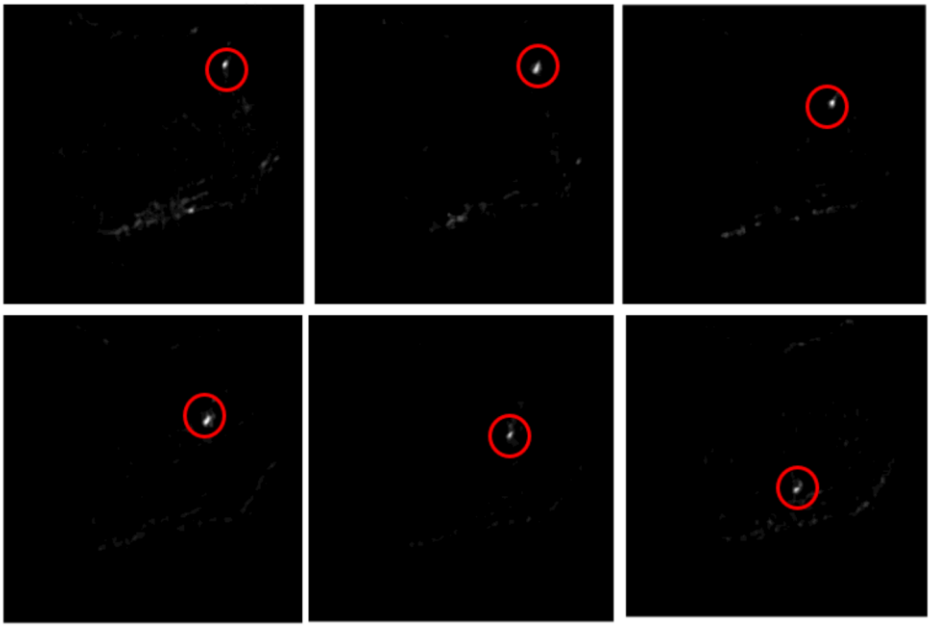


Figure 5-4. Positive training examples. The enhanced tip is surrounded by the red circle.

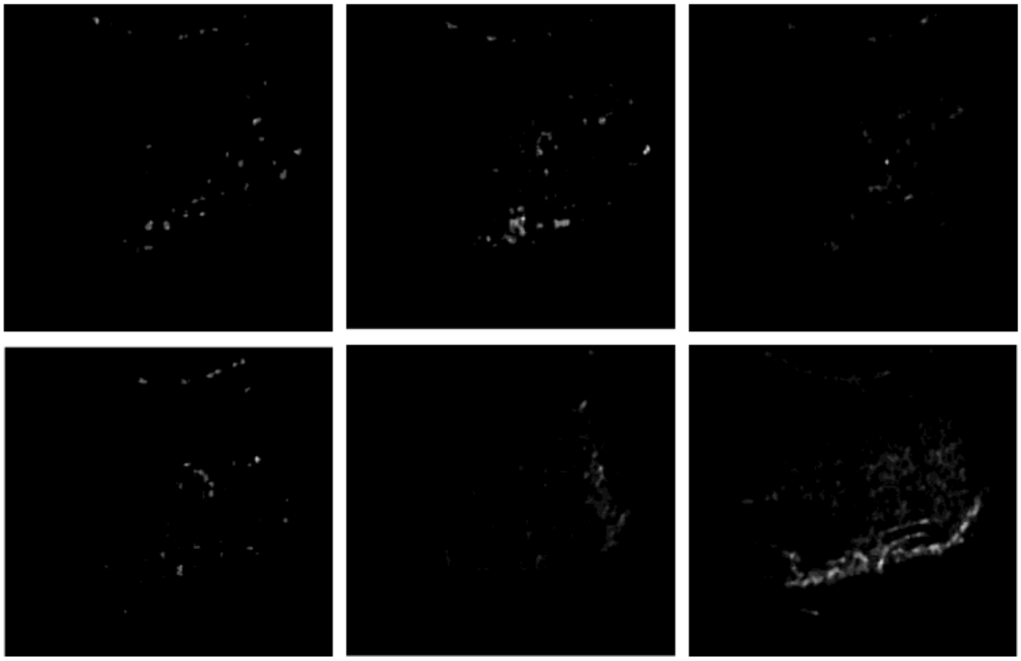


Figure 5-5. Enhanced images where there is no needle tip (negative examples).

During training, we use stochastic gradient descent with momentum (SGDM) optimizer, an initial learning rate of 10^{-2}, and train for 20 epochs. Figure 5-6 shows the evolution of mini batch loss and accuracy, as well as the validation loss and accuracy throughout the training process. We achieve a high classification rate of 98.9% on validation data, meaning that 98.9% of the validation images match the true labels for tip/no tip images.

5.3.5 Tip location regression design

For the regression network, we input the target object, the enhanced needle tip, and the corresponding location labels into the CNN. Keypoints for the needle tip are labelled as shown in Figure 5-8. Note that we use 2 keypoints on the needle tip, i.e., t_p: $\{(x_1, y_1), (x_2, y_2)\}$. Unlike the bounding box approaches like Faster R-CNN [118] and YOLO [120] and used in Chapters 3 and 4 respectively, where

region proposals are utilized to generate candidate regions for detection, here we do not need to define a region bounding the needle tip. Moreover, there is one object of interest in the image, thus we do not need a search region. The use of 2 keypoints is a design choice to geometrically constrain the spatial likelihood of the tip since the enhanced tip feature is not geometrically definable for labelling purposes. The keypoint labels are placed geometrically opposite along a line through the center of the enhanced tip feature and the pixel at the distal end.

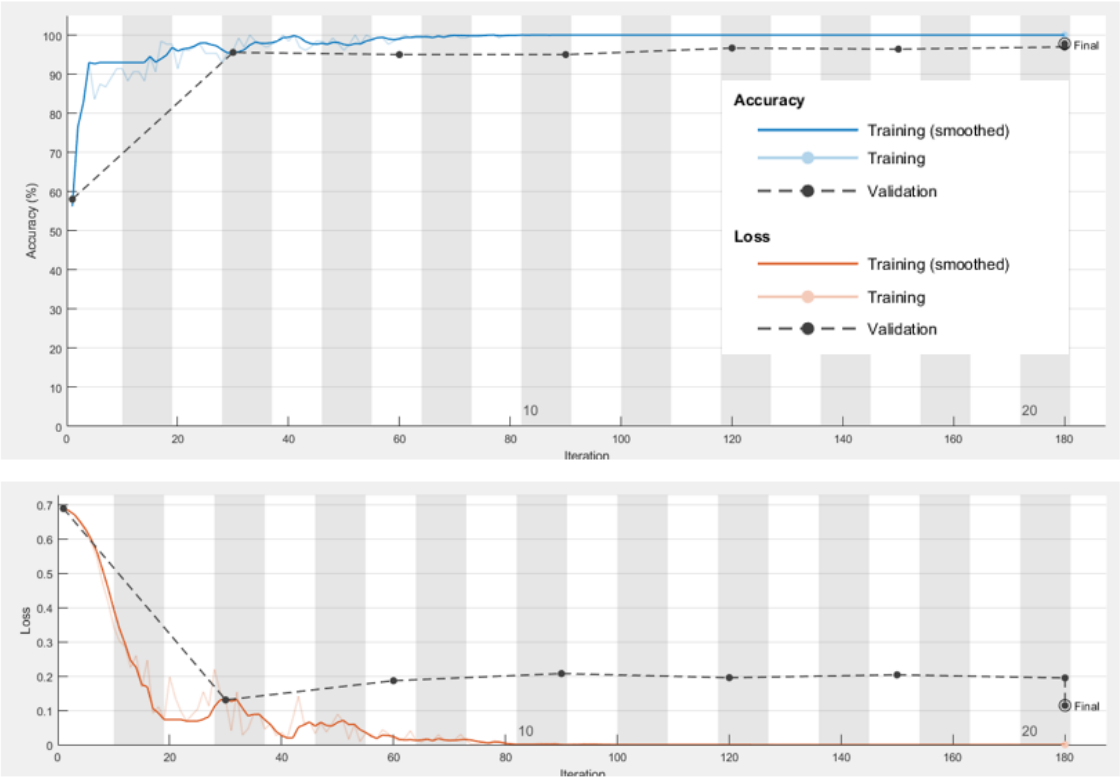


Figure 5-6. Training progress for tip classification. Top graph shows evolution of the mini batch (blue) and validation (black) accuracy, while the bottom graph shows evolution of the mini batch (red) and validation (black) loss.

Figure 5-7. Illustrating keypoint labels for the needle tip. The two labels (red) are on opposite ends of the needle tip.

The enhanced images are augmented by rotating them through $90^0, 180^0$ and 270^0. The labels are computationally manipulated to match the rotated images. From the original 5000 enhanced images, this yields 20,000 training examples. Further, the labels are normalized to be in the range $Q: [-1,1]$. Since the initial labels are in the range $[1,256]$, scaling of the labels reduces the

magnitude of the MSE, and the magnitude of the gradients. Normalization quickens the training process and ensures stable convergence. At test time, the outputs are rescaled to match the original data.

The last feature map for the regression task, from the last FC layer is a vector of dimension $1 \times 1 \times 4$, containing the four ordinates of the two labels: $\{(x_1, y_1), (x_2, y_2)\}$. We use the half MSE (5.5) as our Loss function:

$$\mathcal{L} = \frac{1}{2N} \sum_{i=1}^{N} (y_i - f(x_i))^2 \tag{5.7}$$

In this case, the number of responses $N = 4$, y_i denotes the target output (label) while $f(x_i)$ is the network's prediction.

During training, we use RMSprop optimizer, an initial learning rate of 10^{-3}, a mini batch size of 32 and train for 30 epochs. Figure 5-8 shows the evolution of the mini batch loss and RMSE, as well as the validation loss and RMSE throughout the training process. We achieve a validation RMSE of 0.006.

At test time, the tip location $T(x_t, y_t)$ is directly output from the network outputs as the average of the x and *y* outputs:

$$x_t = \frac{x_1 + x_2}{2},\ y_t = \frac{y_1 + y_2}{2} \tag{5.8}$$

Since the tip location is automatically output from the network without the need for a region-proposal step to tell the network where to look, our framework provides a single-shot approach for tip localization.

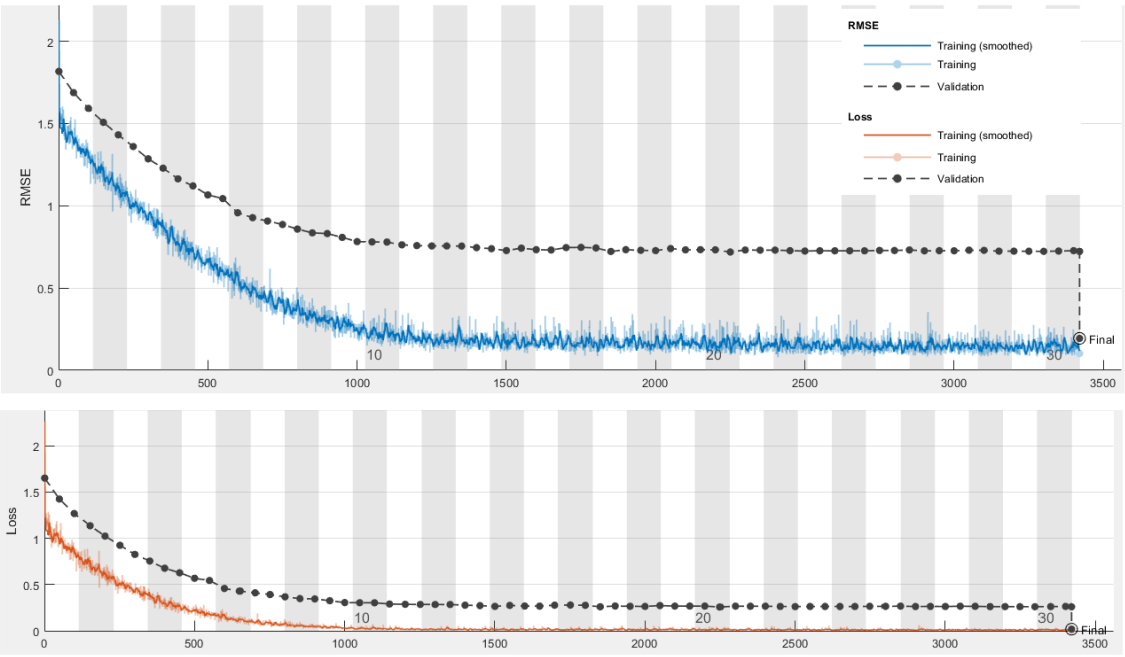


Figure 5-8. Training progress for tip location regression. Top graph shows evolution of the mini batch (blue) and validation (black) RMSE, while the bottom graph shows evolution of the mini batch (red) and validation (black) loss.

5.4 Experimental results

We now present evaluation results for our approach. This includes qualitative results for tip localization, quantitative results for the proposed method using two evaluation metrics: tip localization error, and the computation time. These results are compared with the results of the method in Chapter 4 applied to the same test dataset. Further, we perform ablation studies to justify our design choices for the regression approach and compare our approach with state-of-the-art regression methods.

5.4.1 Qualitative results

Figure 5-9 shows results obtained on a sequence of images collected with in-plane insertion of the 22G needle in a bovine tissue. Imaging is conducted with the SonixGPS system equipped with the C5-2/60 curvilinear probe. Because the 22G has a small diameter, its shaft is imperceptible, and the tip is imperceptible without our enhancement and localization technique. In all cases, the needle tip is accurately localized.

Also note that the frames in Fig 5-9 contain conspicuous needle-like artifacts and high intensity features that have no relation to the needle insertion. Using a method that utilizes static video frames, such as the one in Chapter 3, these features could mistakenly be identified with the needle. Because we use dynamic image information and these features will be present in consecutive frames, our digital subtraction algorithm automatically negates the effect of these features.

Figure 5-10 shows results obtained on a sequence of images collected with in-plane insertion of the 17G needle in porcine tissue. Imaging is conducted with the SonixGPS system equipped with the C5-2/60 curvilinear probe. Again, the needle shaft and tip are inconspicuous or have low contrast with the background, but our framework accurately localizes the needle tip.

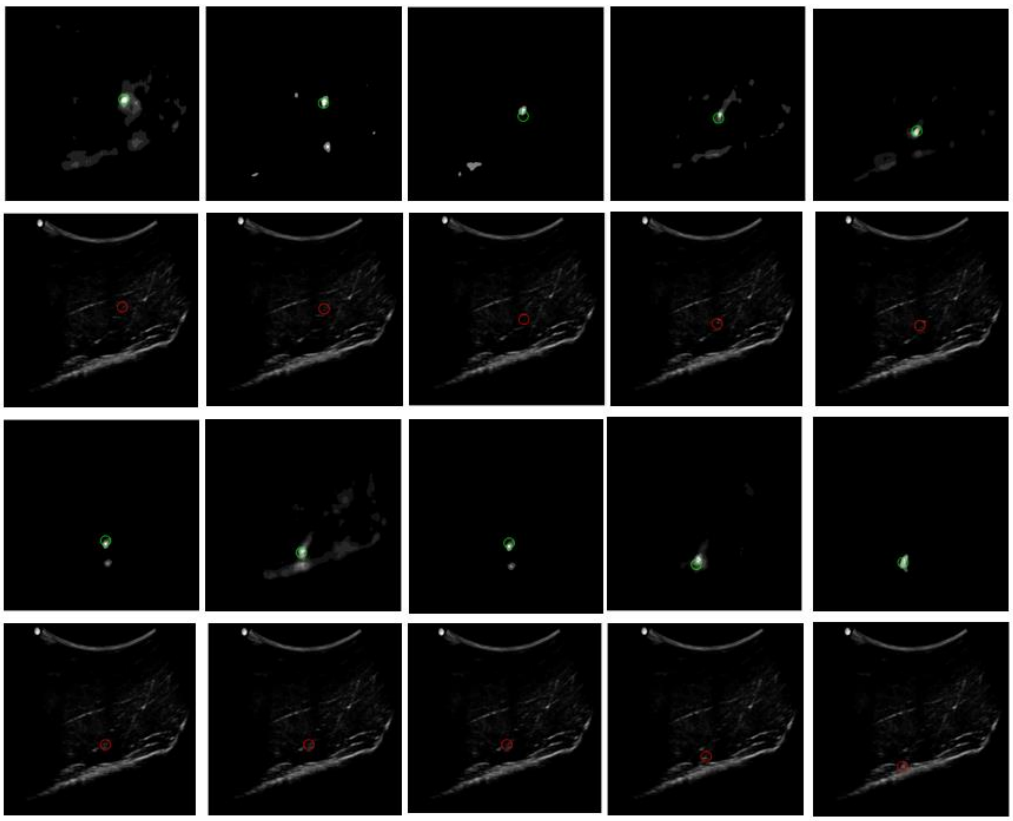


Figure 5-9. Needle localization results for the 22G needle inserted in bovine tissue. Rows 1 and 3 show consecutive enhanced frames with the automatically localized tip position (green). The frames that are classified to be without the needle tip are skipped. Rows 2 and 4 show the enhanced tip position (red) overlayed on the original image. Without our localization framework, the needle tip is imperceptible in almost all the frames.

Figure 5-11 shows results from a on a sequence of images collected with in-plane insertion of the 17G needle in a phantom consisting of porcine tissue overlaid on a spine model. Here, imaging is conducted with the Clarius C3 handheld wireless US system. Images collected with this imaging system were not used at all in training the networks. We note many high intensity interfering

artifacts and the needle tip is not always conspicuous. Nevertheless, our framework accurately localizes the tip.

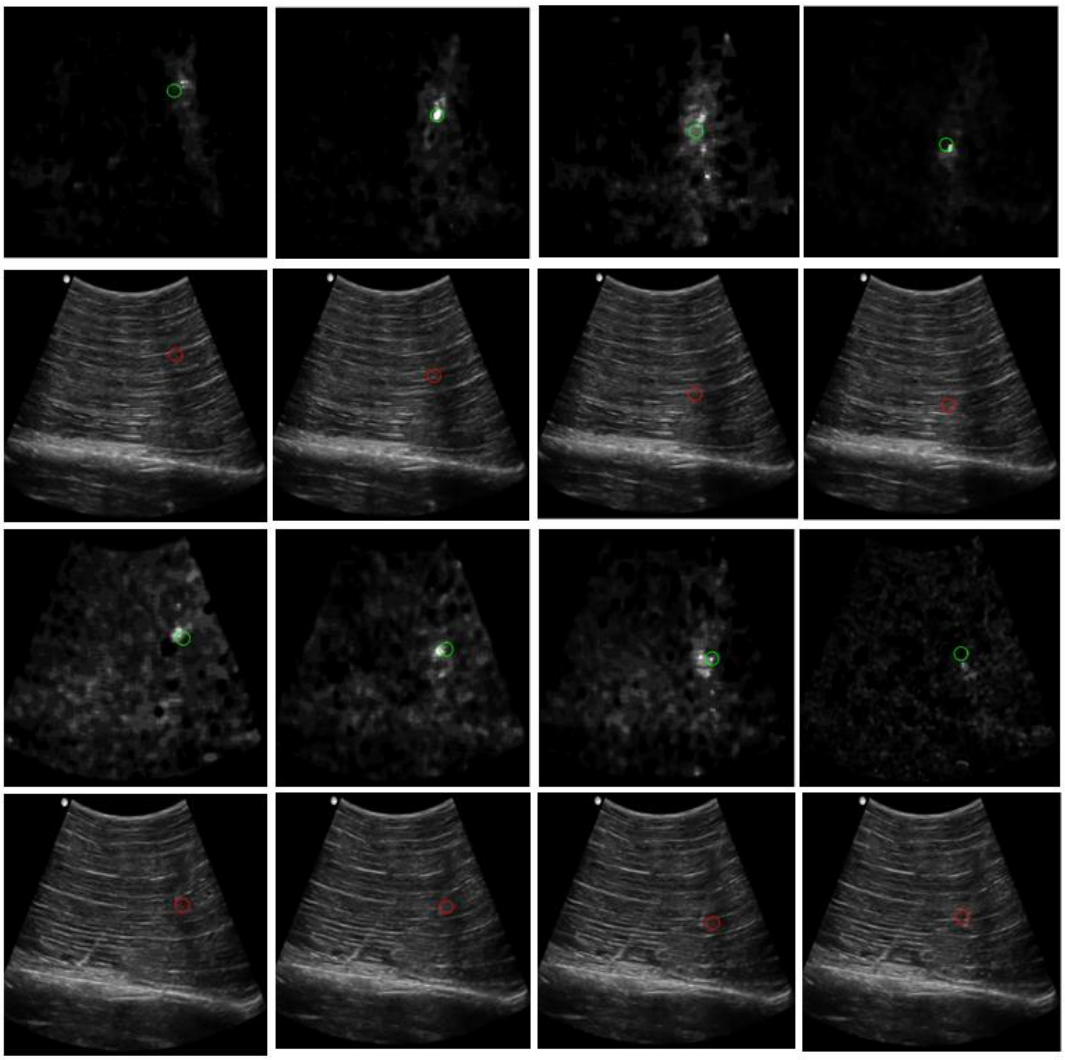


Figure 5-10. Needle localization results for the 17G needle inserted in porcine tissue. Rows 1 and 3 show consecutive enhanced frames with the automatically localized tip position (green). The frames that are classified to be without the needle tip are skipped (This doesn't mean that in the original sequence, there is no tip, but rather, the tip hasn't changed spatial position from the previous frame). Rows 2 and 4 show the enhanced tip position (red) overlayed on the original image. Our method accurately localizes the tip.

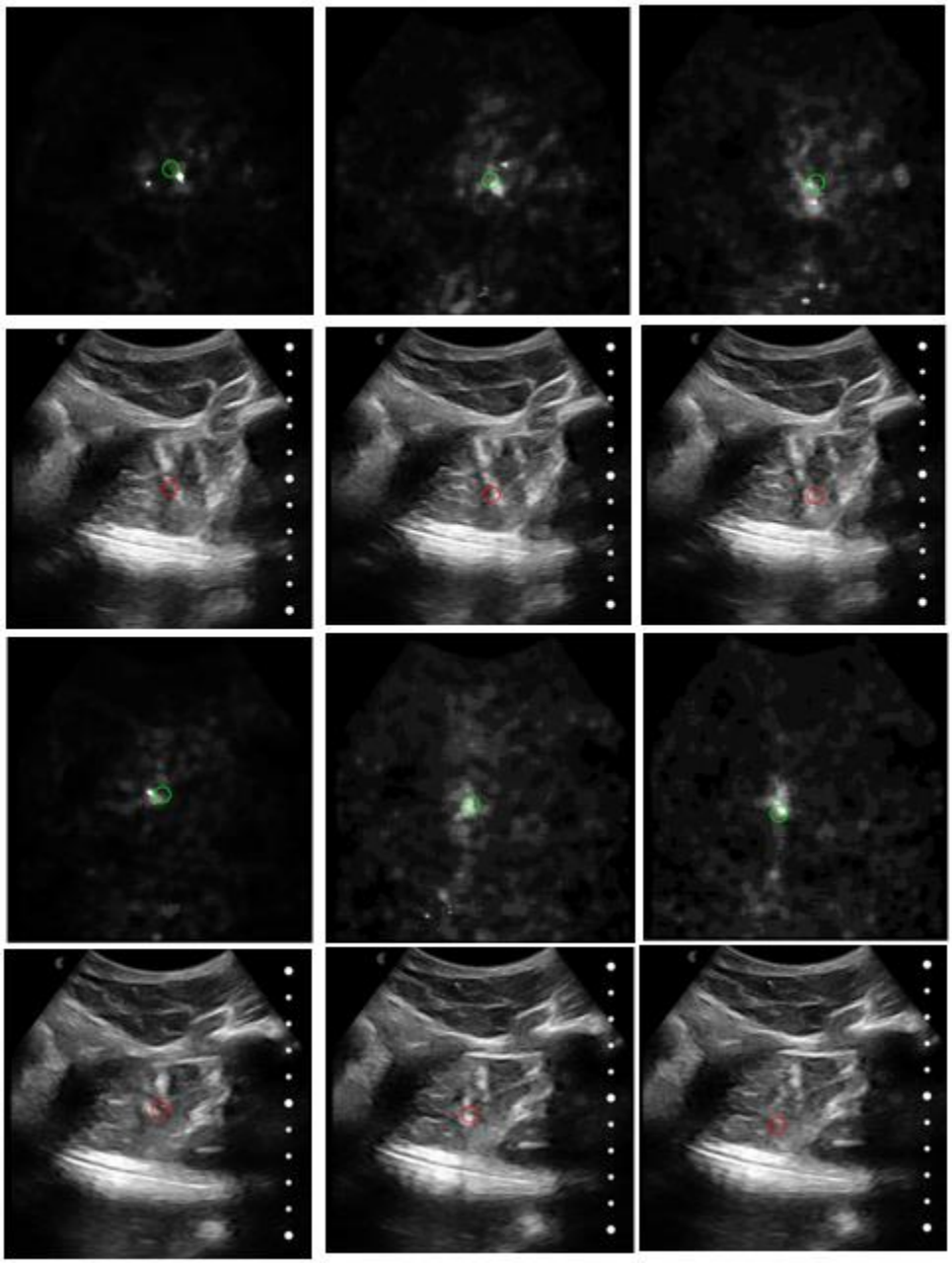


Figure 5-11. Needle localization results for the 17G needle inserted in porcine tissue overlaid on a spine model. Imaging is done with the Clarius C3 wireless US system. Rows 1 and 3 show consecutive enhanced frames with the automatically localized tip position (green). Rows 2 and 4 show the enhanced tip position (red) overlayed on the original image.

5.4.2 Quantitative results

Tip localization error: Localization error is determined from the ED between the automatically determined tip location from our approach and the ground truth from an electromagnetic tracking system for data collected with the 17G SonixGPS needle, and manual identification for the 22G spinal Quincke-type needle since the latter doesn't have tracking capability. During data collection, this needle was repeatedly giggled if the tip was invisible collection so that the tip location could be discerned from the dynamic frame sequence.

Table 5-2 shows the results obtained with our method for both in-plane (IP) and out-of-plane (OP) insertions. We achieve an overall tip localization error of 0.44 ± 0.07 mm. We compare performance of the proposed method with the framework of Chapter 4, on the same test dataset. The results are shown in Table 5-3 and Figure 5-12.

Table 5-2. Tip localization results from the proposed method.

	IP insertion (140)	OP insertion (60)	Total (200)
Mean	0.41	0.56	0.44
SD	0.41	0.42	0.41
95% CI	0.07	0.11	0.06
RMSE	0.58	0.69	0.60
Maximum error	1.64	1.36	1.64

Table 5-3. Tip localization results from the method in Chapter 4.

	IP insertion (140)	OP insertion (60)	Total (200)
Mean	0.58	0.71	0.61
SD	0.37	0.33	0.37
95% CI	0.06	0.08	0.05
RMSE	0.69	1.35	0.71
Maximum error	1.67	0.78	1.67

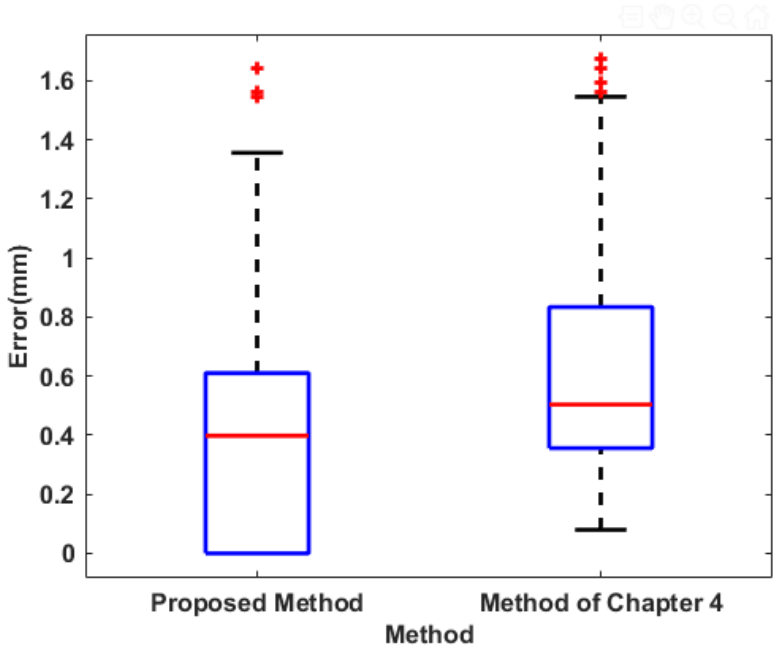


Figure 5-12. Localization error from the proposed method vs the method of Chapter 4.

The proposed method gives a 28% improvement in tip localization error compared to the method in Chapter 4.

Computation time: All our experiments were ran on an NVIDIA GeForce GTX 1060 6GB GPU, 3.6 GHz Intel(R) Core^{TM} i7 16GB CPU Windows PC. The proposed method was fully implemented in MATLAB 2018a using the Deep Learning Toolbox. The learning framework method of Chapter 4 was implemented on the same hardware with Keras 2.2.4 (on the Tensorflow 1.1.2 backend). This computational environment is also used to compare the different architectures proposed in the next section where we perform ablation experiments.

The training of the proposed learning framework is performed completely offline. During test time, we feed the framework with offline video sequencies of

US data. Average processing time for the enhancement process (Section 5.3.1) is 0.002 s. Classification of the enhanced image takes an average of 0.007 s while tip localization lasts for 0.006 s. The overall processing time is thus 0.015 s (67 fps)—a real time speed. This is a drastic improvement on the method of Chapter 4 which achieved an overall processing speed of ~10 fps. To the best of our knowledge, our is the fastest needle tip localization method ever achieved.

5.4.3 Ablation studies

In this section, we present alternative implementations of the proposed CNN framework to justify our design choices.

Network depth: We investigated different network structures, consisting of different numbers of convolution and pooling layers. Recall that the proposed architecture consists of 11 learned layers: 8 convolution layers and 3 FC layers. Each of the first 6 convolution layers is followed by ReLU, batch normalization and max pooling layers. The last 2 convolution layers follow the same pattern but do not have pooling layers.

Since our input image size is 256×256, and the network keeps down sampling the respective input at each layer, the maximum network depth if we included max pooling after all the convolution layers is 8. The tensor after the last pooling layer would be $1 \times 1 \times 128$. We implemented this architecture and evaluated it on the same dataset. We also considered an architecture like the proposed method but with 2 less convolution layers, i.e., 6 convolution/ReLU/batch normalization/max pooling blocks. We evaluate performance of all the architectures using the RMSE on the whole dataset with 5-fold cross validation. All

the other parameters and hyperparameters as in the proposed method were maintained.

A comparison of the performance of these networks vs the proposed architecture is shown in Table 5-4. Note that the proposed architecture gives the best result. It is also confirmed that increasing the number of layers improves network performance.

Table 5-4. Investigating the effect of network depth.

Architecture	**RMSE on validation dataset (normalized)**
Proposed method	0.006
Maximum depth (8 conv layers, all with max pooling)	0.014
6 convolution layers	0.05

Effect of pretraining: In Chapter 3, we saw that pretraining the Faster R-CNN network on natural images and finetuning the weights with the US images (transfer learning) improved network performance. This is because pretraining achieves weights relevant to high level features such as edges, and finetuning refines these weights. In Chapter 3, this approach worked efficiently since edge-like features were present in the images (part of the needle shaft was visible). The results presented for the proposed method in this Chapter were obtained without any weight initialization. When we initialized the weights of the classification and detection models with ImageNet [101] weights, there was no marked improvement in model performance. This is expected since the needle tip in the enhanced images doesn't exhibit features like strong edges that benefit from pretraining.

Impact of network optimizer: Our proposed method employs RMSprop optimizer. We tried alternative training options with Adam [131] and SGDM [128] optimizers. The comparative performance is presented in Table 5-5. RMSprop gives the best performance while SGDM gives the worst performance.

Table 5-5. Comparing different optimizers.

Optimizer	**RMSE on validation dataset (normalized)**
Proposed method (RMSprop)	0.006
Adam	0.05
SGDM	0.14

Impact of batch size: It is well known that a using a larger batch size improves network performance. In the proposed framework, we used a batch size of 32. This is the largest size that could be accommodated by our computing resources. Table 5-6 shows the effect of smaller batch sizes on model performance.

Table 5-6. Effect of batch size on model performance.

Batch size	**RMSE on validation dataset (normalized)**
32 (proposed method)	0.006
16	0.024
8	0.047
4	0.103

Impact of dropout: As earlier explained in Section 3.2.3, dropout layers [100] are sometimes used in learning frameworks to reduce overfitting, especially when training data is limited. We investigated the effect of using dropout layers at several

depths of the proposed architecture, but there was no significant improvement in performance.

5.4.4 Comparison with other regression methods

Here, we compare the proposed regression network with state-of-the-art regression methods [153, 163]. In [153], *Xiang et al.* propose a CNN dubbed PoseCNN for object pose estimation in 6D. Their network for feature extraction consists of 13 convolution layers and 4 max pooling layers. We implemented this architecture without the follow-up layers for semantic labelling. In [163], *Agarwal et al.* implement a network for facial keypoint detection consisting of 4 convolution layers, each with a different kernel size, 4 max pooling layers and 3 FC layers. All convolution/activation/max pooling/dropout sequences and all FC layers are interspersed with dropout layers. We also implemented a similar architecture for our needle tip localization task.

Table 5-7 shows comparative analysis between the proposed method and these alternative approaches. Our method achieves the best RMSE. Although the method in [163] is marginally faster, it has poor localization accuracy, evidenced by the high RMSE.

Table 5-7. Comparison with state-of-the-art regression frameworks.

Method	**RMSE on validation dataset (normalized)**	**Computation time (s)**
Proposed method	0.006	0.006
PoseCNN [153]	0.029	0.008
NaimishNet [163]	0.303	0.006

5.5 Discussion

In this Chapter, we have introduced a novel method for needle tip localization from 2D US sequencies. The approach combines three subprocesses: tip enhancement from digital subtraction of consecutive frames, tip classification to determine whether an enhanced image contains tip information, and regression to automatically output the tip spatial coordinates. We demonstrate experimentally that our novel learning framework facilitates accurate needle tip detection (28% improvement on the method in Chapter 4), and real-time computation speed of 67 fps (570% improvement on the method in Chapter 4). To the best of our knowledge, this is the fastest needle enhancement and localization approach ever reported. Moreover, given more data and computing resources, our method can further be improved.

Our method works best if the enhanced needle tip appears as a salient high intensity feature in the tip enhanced image. If multiple such features are present in the enhanced image, or if part of the shaft is enhanced, our method doesn't perform as well. In fact, augmenting data for training the regression network by rotation becomes an impediment in this respect: the top part of an elongated feature can be mistaken as the tip. These scenarios in which the method fails are illustrated in Figure 5-14. These occurrences were few in the evaluation dataset.

Ultimately, the proposed method would be implemented in a pseudo real-time computational imaging platform to give the radiologist additional insight into location of the needle. The estimation of tip localization doesn't eliminate the need

for the radiologist to use actual US image information. Therefore, even if the tip localization is off by a few pixels, our method is still helpful since it would narrow the region in which the radiologist should look.

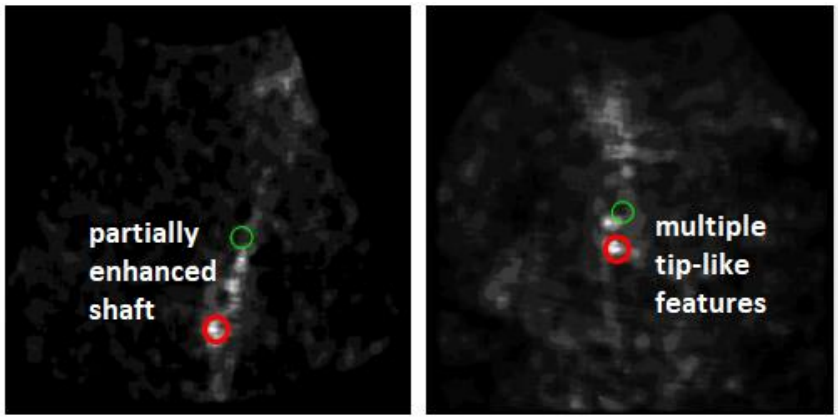


Figure 5-13. Instances where the proposed method fails. The correct tip location is surrounded by the red circle. The output of the proposed method is surrounded by the green circle.

Note that since our method relies on subtraction of consecutive frames, it is prone to motion artifacts arising from say, transducer motion or physiological activity such as breathing or pulsation in the clinical setting. During data collection, transducer motion was minimized, although minimal motion artifacts were simulated by pressing the transducer against the imaging medium. We hypothesize that even in scenarios where there is drastic motion, our classification step, subject to a large dataset will still accurately detect presence of the needle tip and the regression step would then accurately localize it. This will form part of our future work.

Chapter 6 -Learning based needle localization in 3D US

6.1 Overview

In the previous Chapters, we have developed methods for detection, enhancement and localization of needles in 2D US. In this Chapter, we want to demonstrate that these methods can be extended to 3D US. Because 3D US produces volume data, aligning the needle with the transducer is less of a challenge because at least one of the slices will capture needle tip information. Nevertheless, the needle dimensions are very small compared to the whole volume. Therefore, visualizing the needle in the context of the whole volume is like trying to find the proverbial ‘needle in a haystack’!

In the context of interventional procedures, we do not require the whole volume information. Rather, we are interested in the subsection of the volume containing needle data and the orthogonal plane containing the needle tip. In this chapter, we propose a novel learning-based algorithm for localizing the orthogonal planes containing needle data, the best orthogonal plane for needle visualization, and localization of the tip in 3D.

The proposed algorithm is trained on 100 volumes and tested on 20 volumes collected in ex vivo experiments using two different 3D transducers with a 17G needle inserted parallel to the long axis of the transducers. The framework

achieves a best plane localization time of 1.55 ± 0.46 s, a best plane localization error of 0.18 ± 0.04 and a tip localization error of 0.54 ± 0.15 mm.

6.2 Background

The US volume consists of a sequence of 2D slices arranged in a 3D Cartesian grid. Figure 6-1 illustrates the geometrical arrangement of the slices. Each volume element (voxel) surrounds a sample point, that is part of a set $\Omega(x, y, z, \chi)$ with χ as the value of the data at a spatial 3D point (x, y, z). Because of the dimensional mismatch between the needle voxels and the rest of the volume, identifying the needle in the whole volume is difficult.

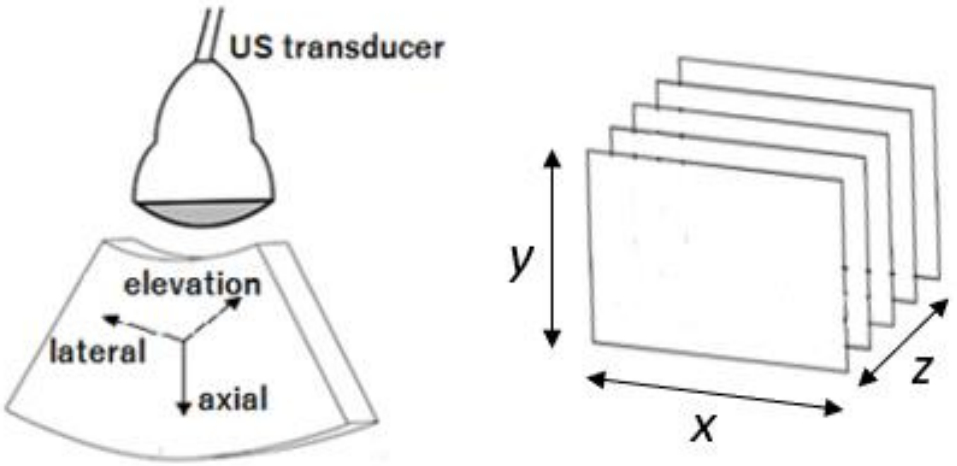


Figure 6-1. Geometrical illustration of a 3D volume in a Cartesian plane. Here, x, y and z are the lateral, axial and elevation directions respectively. x-y slices, parallel to the long axis of the transducer are of interest for in-plane needle insertion.

Previously, algorithms for needle enhancement and localization in 3D US were reported. A cross section of these methods includes: the 3D Hough transform (HT) [55], iterative model-fitting methods based on RANSAC [164] and learning

based methods [73]. These methods generally suffer from computational complexity due to the large amount of volume data that must be processed. Although the RANSAC based ROI-RK method proposed in [55] reduces calculation time, it is not robust to high intensity artifacts and steep insertion angles. In [64], oscillation of a needle stylus was modeled into a projection-based localization framework, providing a more robust solution. However, oscillating the stylet during US guided needle insertion is difficult in a single operator scenario, especially for shallow angles.

Our proposed framework is designed to overcome these challenges by 1) providing a means for needle localization in low contrast and occluded volume scenes, 2) achieving needle localization at a high computational speed, thus making the approach suitable for a real-time application.

6.3 Methods

We propose a two-stage framework illustrated in Figure 6-2. The algorithm is designed to achieve two objectives: 1) slice selection to extract slices (2D frames acquired from a motorized 3D transducer) that contain most needle information. 2) needle enhancement and tip localization. The following sub-sections describe this process in detail.

6.3.1 Needle detection

We use a Faster R-CNN learning framework described in Section 3.3.1 and illustrated in Figure 3-4. Since the 3D volume consists of 2D parallel images, we break down the 3D detection problem into a series of 2D sub-problems.

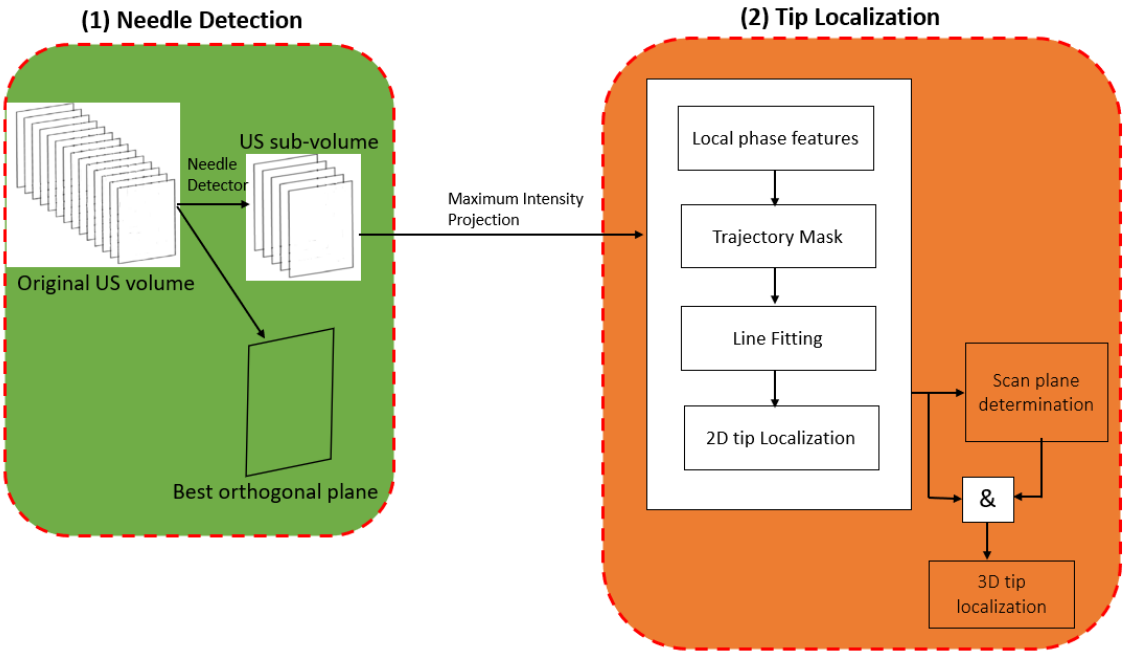


Figure 6-2. Block diagram of the proposed approach. (1) A detector is used to classify slices that contain needle data. From this stage, we can obtain the orthogonal slice with the best needle visibility, and a sub-volume with only slices that contain needle data. (2) Needle tip localization is performed on the sub-volume after enhancement of needle data.

Preprocessing: Before inputting to the 2D model, each $x-y$ slice in the volume is preprocessed to remove speckle and needle look-alike artifacts. This is accomplished in two stages. First, the 2D image's contrast is stretched by saturating the bottom 1% and the top 1% of all pixel values in the image. Horizontal needle like artifacts are then removed using a morphological operation with a Top-hat filter:

$$TF\big(I(x,y)\big) = I(x,y) - D_L[E_L(I(x,y))], \tag{6.1}$$

Here, $I(x, y)$ is the contrast-stretched image, L is a linear structuring element while D_L and E_L denote dilation and erosion operations respectively. The result of the processing step is illustrated in Figure 6-3.

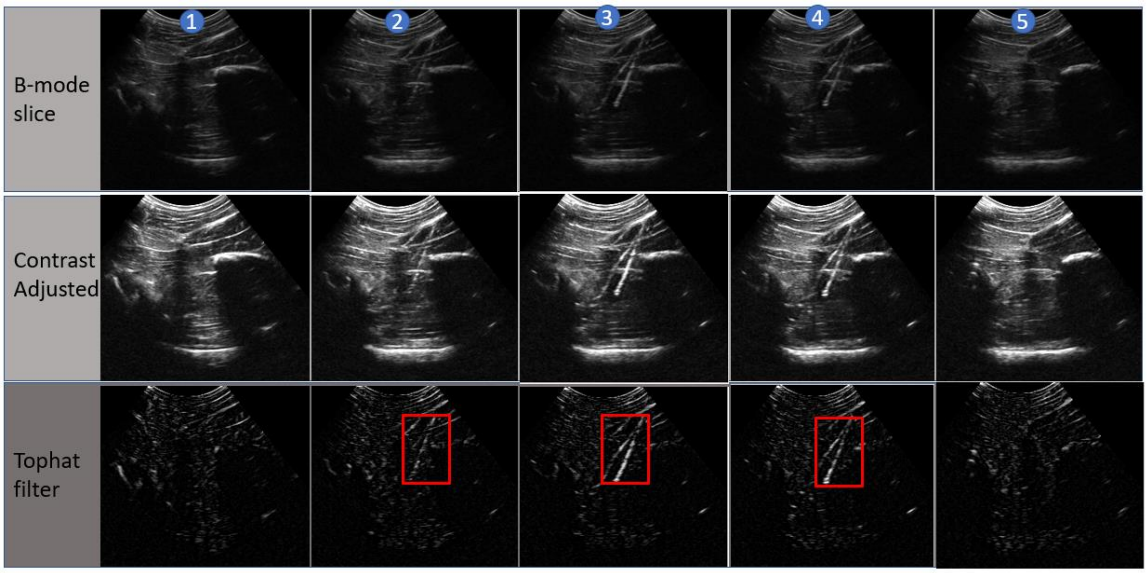


Figure 6-3. Preprocessing of consecutive slices extracted from a 3D US volume. (1) and (5) do not contain needle data. In the final preprocessed image, the needle is annotated with the red rectangle.

Learning guided slice extraction: In a typical volume acquired with a motorized transducer, less than 10% of the coronal ($x - y$) slices contain needle data. This can be explained if we consider a typical needle of diameter 1.5 mm vs the US beam of thickness ~ 1 mm and a volume resolution of 0.3 mm. These slices are sandwiched between slices without needle data (Figure 6-3), and the task of the detection model is to identify them.

The learned model is applied to each of the $x - y$ slices in the original volume, US_{volume}. It is expected that slices containing the needle will give a high detection score, accompanied by the corresponding bounding box information. We

use a detection score threshold of 0.6 to eliminate false detections. This value is empirically determined. The slices with needle information can now be separated from the rest of the volume. These slices now constitute a sub volume, US^*_{volume}. If the needle is well aligned parallel to the long axis of the transducer and the insertion is not very deep, one of the coronal slices in US^*_{volume} will give complete visualization of the needle. It is expected that this slice would give the highest detection confidence.

Volume reduction saves computing load in the needle enhancement and localization steps that follow. It also removes slices that have artifacts which would degrade needle enhancement. Figure 6-4 illustrates an example of needle detection (rectangular annotation) from volume data. Figure 6-5 shows a plot of the corresponding needle detection confidence vs slice number. In this example, the slice with the highest detection confidence provides full visibility of the needle.

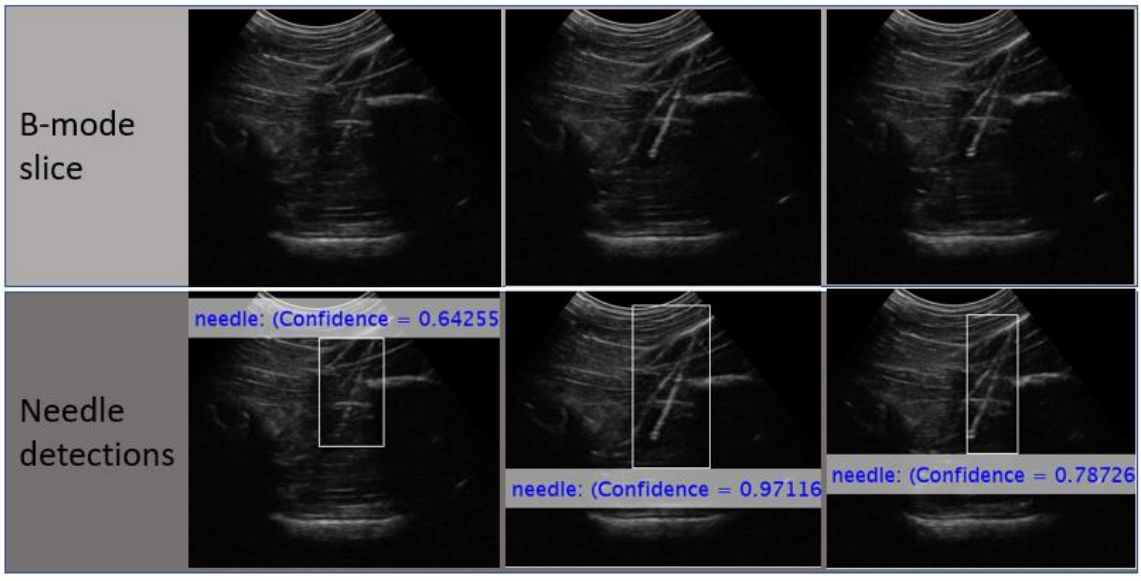


Figure 6-4. Detection in 3 consecutive slices containing needle data, extracted from a 3D US volume.

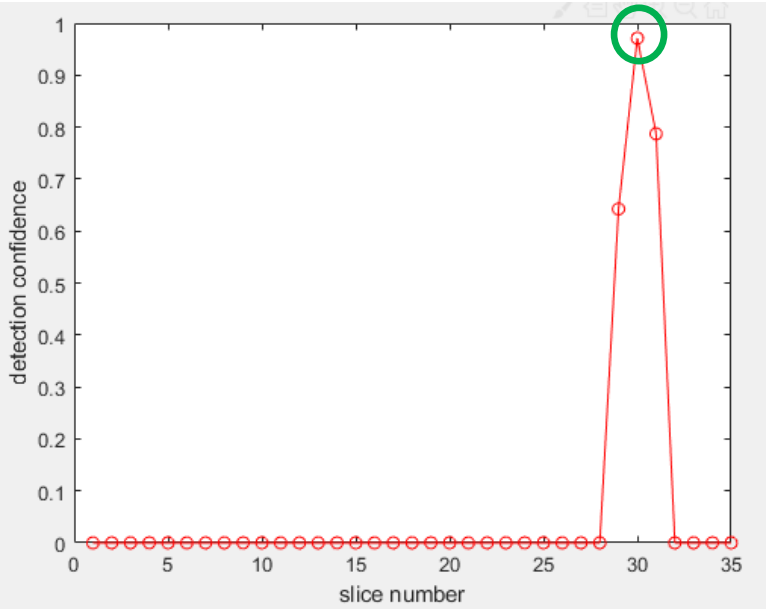


Figure 6-5. Plot of needle detection confidence vs slice number in the 3D US volume. The highest confidence (green) corresponds to the best orthogonal slice for needle visualization.

6.3.2 3D needle tip localization

We will consider the acquisition system illustrated in Figure 6-6, where the needle is inserted in the $y-z$ plane, and the $x-y$ (coronal) plane is parallel to the needle insertion direction. Our interest is determining $\Omega(x', y'z')$, the 3D tip location.

Recall that the needle detection step enables us to extract a sub volume, ${US^*}_{volume}$ containing only slices that have needle data. Although the detection results may yield the best orthogonal slice, thereby making further localization unnecessary, there could be instances when the needle shaft is broken such that the tip is disconnected, or needle features have low contrast in all the slices. This scenario may result when the needle insertion is deep, or the needle is not properly

aligned with any of the imaging planes from which the 3D volume is constructed. In this case, needle enhancement and localization from US^*_{volume} is necessary.

A typical detection result in this scenario is shown in Figure 6-7. From US_{volume} containing 41 slices, only one slice gives a detection confidence above the threshold. Note from Figure 6-7 that even within this slice, the needle shaft is discontinuous, and the shaft is not easily discernible. The 3D tip localization process that follows comprises two steps: 1) 2D tip localization, 2) scan plane determination.

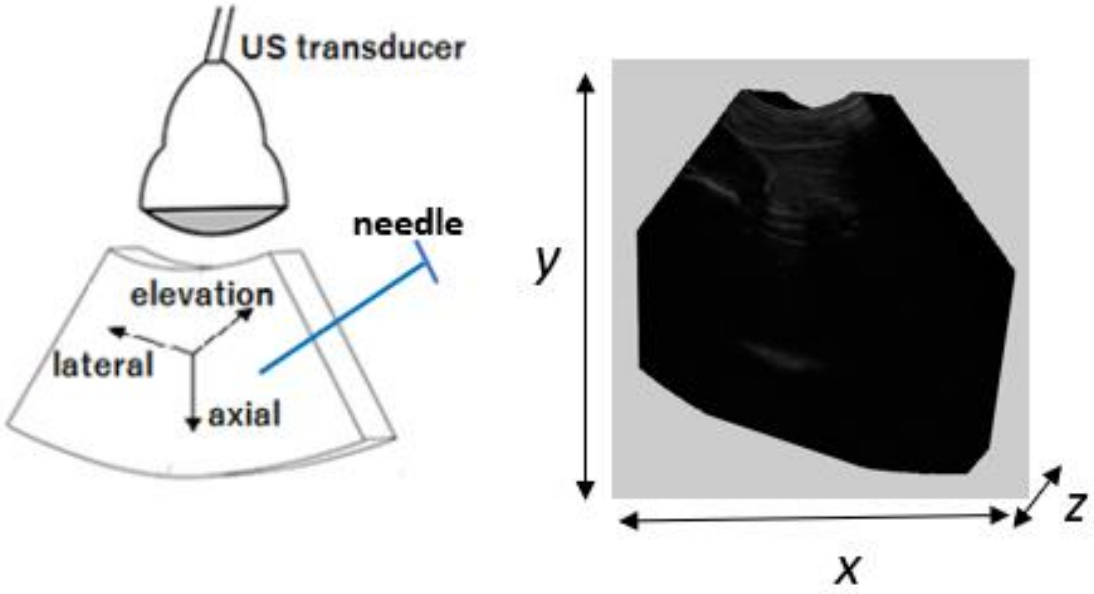


Figure 6-6. 3D US acquisition geometry.

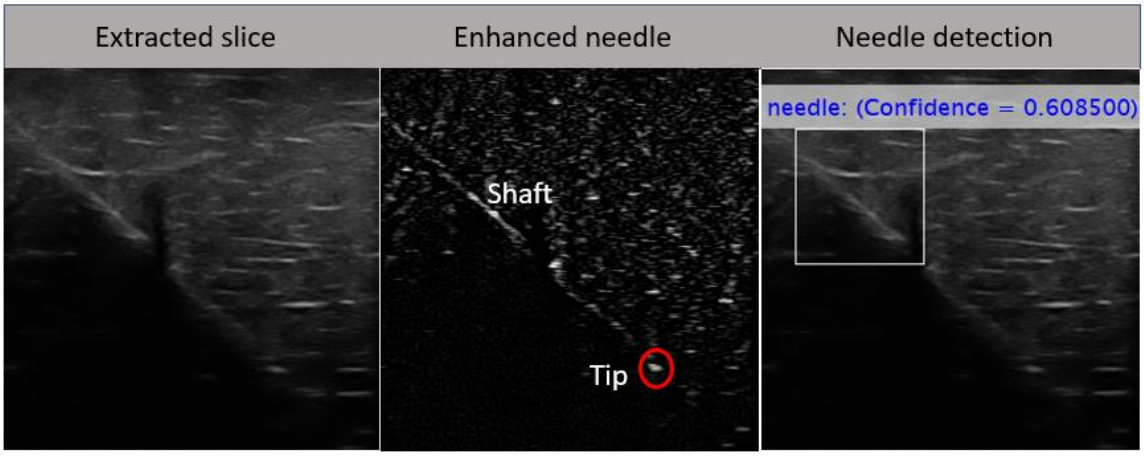


Figure 6-7. From detection to tip localization. Here, only one slice in the volume gave a detection confidence above the threshold of 0.6. The middle image is the preprocessed slice used as input to the detection model.

2D Tip localization

2D tip localization follows the approach described in Section 2.3.4. Figure 6-8 shows the 2D tip localization pipeline. Hereafter, we briefly describe the processes involved.

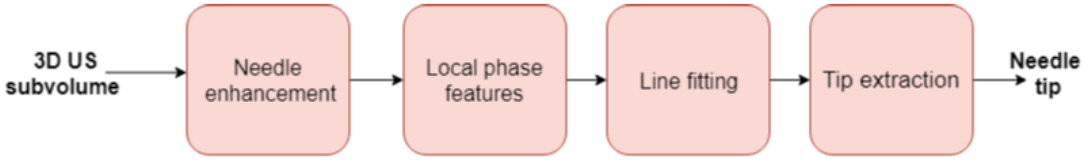


Figure 6-8. Processing steps for 2D tip localization in 3D US data.

Needle enhancement: The needle enhancement process utilizes the preprocessed images previously used as input images to the learning model, described in Section 5.3.1, i.e., contrast stretching to saturate the top 1% and

bottom 1% of the pixels in each slice, followed by a Top-hat filter with a linear structuring element. We are interested in only the slices constituting US^*_{volume}.

We derive the enhanced needle 2D image $P_{x,y}$ from a Maximum Intensity Projection (MIP) of US^*_{volume}, by extracting maximum intensity values along optical paths in the z direction. This approach works efficiently since all needle voxels have collocated x' and y' ordinates in US^*_{volume}. If only one slice exists in US^*_{volume}, computation of the MIP image is unnecessary.

Local phase features: We determine the phase symmetry $PS(x,y)$ of $P_{x,y}$ in a region automatically defined by the needle bounding in US^*_{volume}. $PS(x,y)$ is calculated using a bank of orientation tuned Log-Gabor filters (Section 2.3.4). $PS(x,y)$ is used to estimate needle trajectory using the radon transform. The estimated trajectory is expanded to define a mask of the region where the needle must lie in $P_{x,y}$.

Line fitting: Needle data in the trajectory region defined in the previous step filtered using the MLESAC algorithm, which prunes outliers and keeps only data points with linear coherence.

Tip extraction: The point cloud from the previous step is filtered with a Log-Gabor filter bank without orientation selectivity. This is followed by feature extraction on the resultant point cloud using a combination of spatially distributed image statistics which enhance the needle tip (Section 2.3.4). This yields the projection enhanced image denoted as $PE(x,y)$. (x',y') is determined from the first maximum intensity pixel at the distal end of the needle trajectory in $PE(x,y)$. The

overall process for tip localization in a projection of US^*_{volume} is illustrated in Figure 6-9.

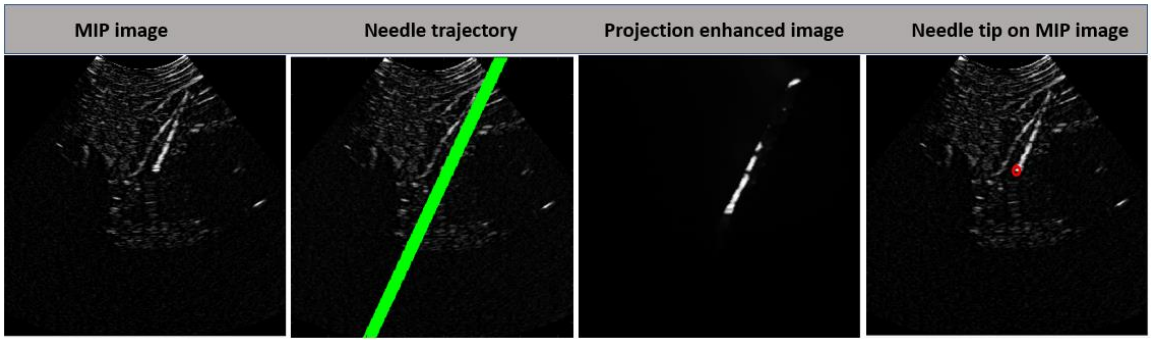


Figure 6-9. Tip localization on projection of the original volume US^*_{volume}.

Scan plane determination

In our context, *scan plane* refers to the slice containing the needle tip, which is the most advanced portion of the needle in the elevation (z) direction of the volume. The scan plane is determined by calculating $\sum_{i=-\gamma}^{+\gamma}\sum_{j=-\gamma}^{+\gamma} I(x'+i, y'+k)$, the sum of pixel intensities in a bounded square patch of length 2γ centered at (x', y') in each slice within US^*_{volume}. The scan plane is estimated as the slice with the maximum intensity sum. The result gives us z'.

6.4 Learning from handcrafted features

We sought to compare the deep learning approach for identifying (classifying) slices that contain needle data, with a traditional machine learning approach. As earlier mentioned in Section 3.2, such learning models rely on handcrafted features; manually extracted features from raw data or features

extracted by other models. Since the needle exists as a hyperechoic oriented linear feature, it is easy to define for the learning task.

Previously, locally normalized histograms of oriented gradients (HOG) descriptors were shown to be efficient at capturing gradient information [165]. They are also invariant to translations or rotations, demonstrating performance like Scale Invariant Feature Transformation (SIFT) descriptors. As such, locally normalized HOG descriptors make robust feature sets for needle detection. In our design, we extract intensity-invariant local phase descriptors and use them to derive HOG descriptors.

6.4.1 Local Phase Descriptors for Needles

We apply orientation tuned log-Gabor filter banks to each slice in US_{volume} to extract a needle phase descriptor, hereafter denoted as $NPD(x, y)$. The filter parameters are selected automatically using the framework proposed in [68]. Here, it is assumed that the insertion side of the needle is known *a priori,* and the calculation is limited to a fixed ROI on the insertion side containing a visible part of the needle shaft. The output of the filter operation gives a phase-based descriptor called phase symmetry, $PS(x, y)$ which is used as an input to the MLESAC algorithm. We use MLESAC to prune false positive pixels and connect inliers to yield $NPD(x, y)$. Figure 6-10 shows examples of slices with and without $NPD(x, y)$. Investigating Figure 6-10 (first and last columns), we note that slices without needle data do not contain $NPD(x, y)$ while slices with needle data (middle 7 columns) possess $NPD(x, y)$, existing as bright intensity straight features, expected from a rigid needle.

6.4.2 Detector Architecture

The application of the HOG algorithm in our approach is illustrated in Figure 6-11. Details of the HOG approach are well presented in [165]. We use $L2-Hys$ $(Lowe-style\ clipped\ L_2\ norm)$ contrast normalization on overlapping 3×3 cell blocks of 4×4 pixel cells: From the unnormalized descriptor vector $\mathbf{v}$, $L2-Hys$ is determined by clipping the $L_2\ norm,\ \mathbf{v} \rightarrow \mathbf{v}/\sqrt{||\mathbf{v}||_2^2 + \epsilon^2}$ where ϵ is a small constant. This normalization is done to achieve invariance to geometric transformations. HOG computation is performed using a 64×128 sliding detection window, and the resulting descriptor is fed to a linear support vector machine (SVM) baseline classifier.

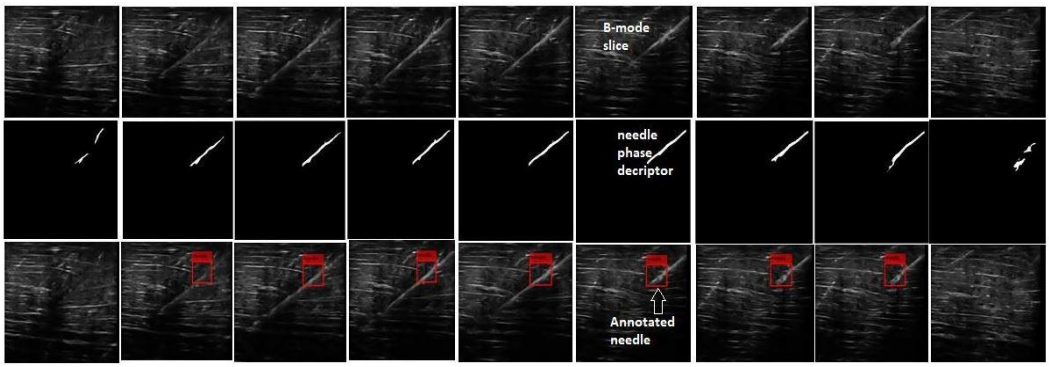


Figure 6-10. The needle detection process. Top-row: B-mode US slices constituent of US_{volume}. The original volume comprised of 41 slices. Here, we show only 7 slices containing needle data, sandwiched between two slices (first and last columns) without needle data. Middle row: Respective $NPD(x, y)$ images. The slices with needle data possess a salient straight feature with minimum bending. The slices without needle data lack such features. Bottom row: Slice classification results after running the detector. The classification accuracy here was 100 %.

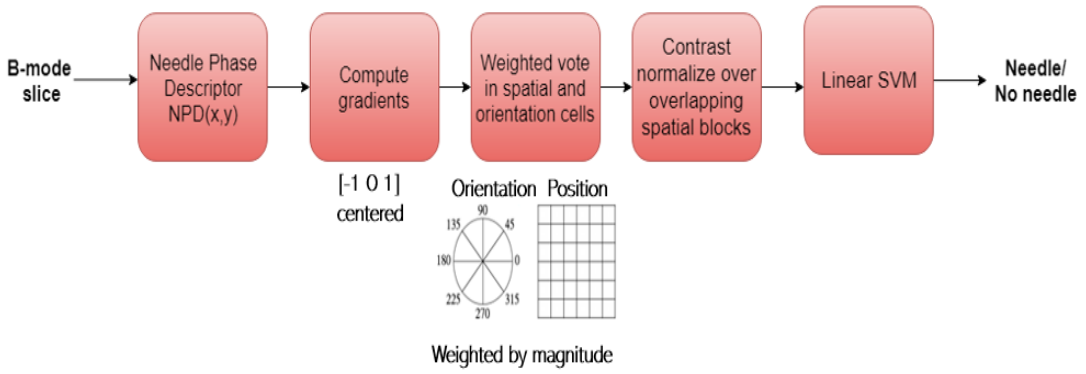


Figure 6-11. Using HOG algorithm for slice classification.

The detector is applied to each slice in US_{volume} after preprocessing to elicit needle phase descriptors like those used in training the detector., The resulting sub-volume, US^*_{volume}, consists of only slices that contain needle data. Figure 6-10 (bottom row) illustrates an example of needle detection from volume data. Detected needles are shown with rectangular annotation. Once we have US^*_{volume}, the 3D tip localization approach follows the same approach as that used for the deep learning method (Section 5.3.2).

6.5 Data Acquisition and Experimental Validation

3D US volumes were acquired using the SonixTouch system (Analogic Corporation, Peabody, MA, USA) equipped with the 4DL14-5/38 linear and the 4DC7-30/40 convex volumetric transducers. A 17-gauge (1.5 mm diameter, 90 mm length) Tuohy epidural needle (Arrow International, Reading, PA, USA) was inserted into freshly excised bovine tissue and porcine shoulder phantoms. We conducted experiments using different transducers and imaging media to

investigate robustness of our method to different imaging conditions. The dataset properties are shown in Table 6-1.

The transducer motor was automatically controlled during insertion to achieve a FOV of 10° for sweeps of 0.244° degrees per frame and 41 frames per volume. Each volume has an average dimension of $432 \times 480 \times 41$ $(x \times y \times z)$. Multiple experiments were performed at various needle depths (40-80 mm) and orientations ($30^{\circ} - 70^{\circ}$) with the needle inserted in a native axial/elevation $(y - z)$ direction of the volume. A total of 200 volumes, 100 for each probe type were collected. The US system settings were fixed for all imaging sessions. The volumes were divided into 3 sets without overlap: 170 for training and validation, and 30 for testing.

Table 6-1. Materials and experimental settings for 3D US data collection.

Bovine tissue			
Transducer	**Needle dimensions and insertion profile**	**# of videos**	**Voxel size (mm)**
4DL14-5/38	17G (1.5 mm,70 mm), $30^{\circ} - 50^{\circ}$ 17G (1.5 mm,70 mm), $50^{\circ} - 70^{\circ}$	50 50	0.3 mm 0.3 mm
Porcine shoulder			
Imaging system	**Needle type, dimensions and insertion profile**	**# of videos**	**Pixel size**
4DC7-30/40	17G (1.5 mm,70 mm), $30^{\circ} - 50^{\circ}$ 17G (1.5 mm,70 mm), $50^{\circ} - 70^{\circ}$	50 50	0.36 mm 0.36 mm

The proposed framework was implemented in MATLAB 2018a on a 3.6 GHz 4.2 GHz Intel(R) CoreTM i7 CPU, 16GB RAM Windows PC. The Log-Gabor filter parameters were determined automatically using the method proposed in [68]. For the training dataset, groundtruth data: 400 2D US images were manually extracted manually from the volumes. Performance of the needle detector was evaluated by

calculating Precision (P) and Recall Rate (RR) where: $P = True\ Positive/(True\ Positive + False\ Positive)$ and $RR = True\ Positive/(True\ Positive + False\ Negative)$. To determine localization accuracy, the ground truth tip location was segmented manually by an expert user in volumes where the tip was visible. Tip localization error was determined by calculating the ED between the automatically localized tip and the manually segmented tip.

6.6 Results

We present qualitative and quantitative results for the proposed deep learning approach for needle detection and subsequent tip localization. We also present a comparative performance analysis with the alternative machine learning approach utilizing handcrafted features for slice classification.

6.6.1 Qualitative results

Figure 6-12 shows qualitative results for needle enhancement and localization. The needle tip is overlayed on the scan plane; the orthogonal plane containing the tip. The first row shows results for US data collected with a curvilinear volumetric transducer, while the other two rows show results for US data from a linear volumetric transducer.

Our method gives accurate tip localization for moderate to steep insertion angles, including cases where the shaft is discontinuous. Shaft discontinuity is compensated for by MIP which aggregates needle data from neighboring slices.

6.6.2 Qualitative results

The proposed method yields an average precision of 94%, recall rate of 96% and detector execution per slice of 0.046 s for a 300×300 input (0.016 s for preprocessing and 0.03 s for detection). For a 41-slice volume, this translates to 1.89 s. The post-processing step for tip localization and scan plane determination takes 1.05 s. Therefore, the overall execution time is 2.93 s per volume. Further, we achieve an average tip localization error of 0.46 ± 0.44 mm.

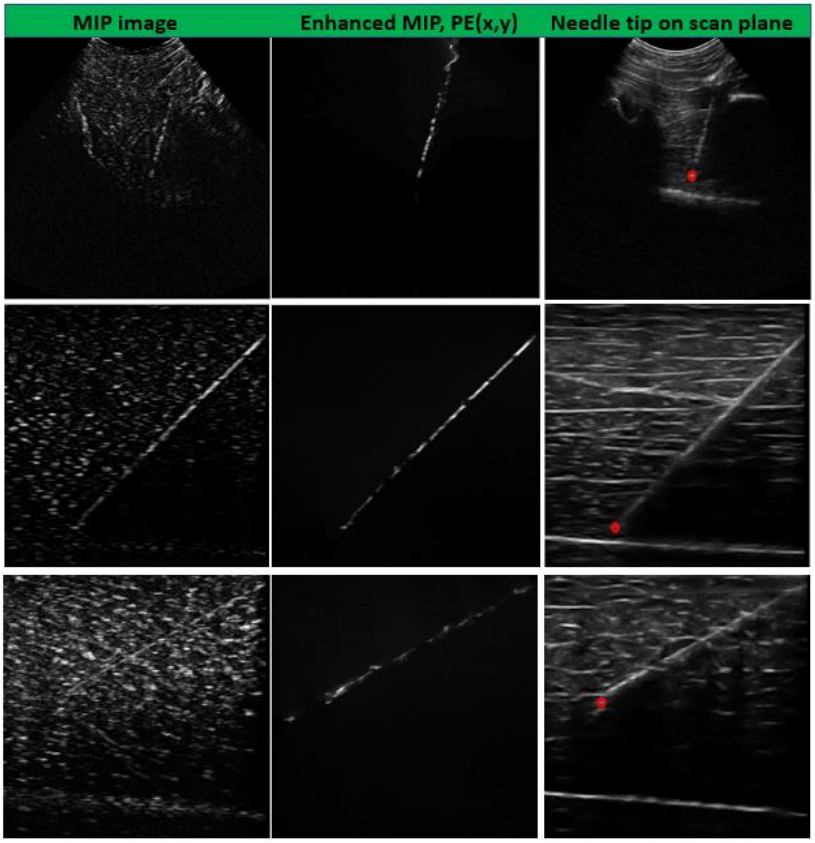


Figure 6-12. Needle tip localization (red) results in 3D US data.

Comparative statistics for the alternative method utilizing traditional machine learning, where HOG features are extracted from handcrafted needle features, are shown in Table 6-2 and Figure 6-13. We observe that the proposed method outperforms the alternative approach in computation speed. This is owed to the featured extraction step necessary for the latter. We do not see a big difference in the accuracy of identifying the slices with needle data. Further, there is no significant difference between the tip localization errors. This is expected since both methods accurately extract useful slices, and we use the same method for tip localization.

Table 6-2. Performance metrics for slice classification and needle localization

	Proposed Method (Deep learning)	**HOG featured based learning**
Preprocessing time (whole volume)	0.66 s	15 s
Slice classification	1.23 s	2.5 s
Tip localization	1.05 s	1.05 s
Total processing time	**2.93 s**	**18.55 s**
Precision	94%	88%
Recall	96%	98%
Tip localization error (95% CI)	0.46 ± 0.14 mm	0.44 ± 0.13 mm

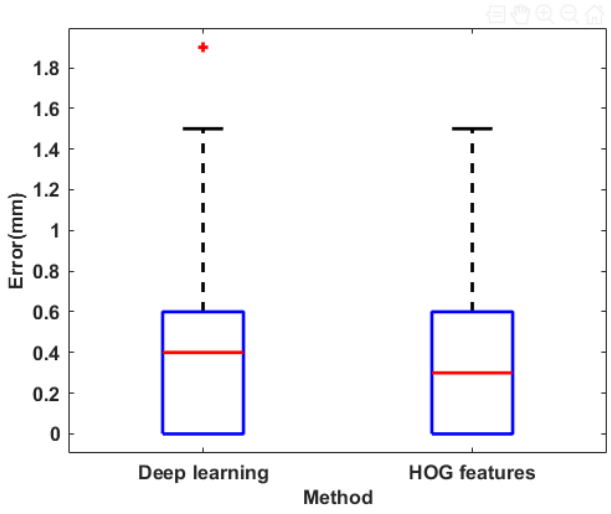


Figure 6-13. Comparing tip localization errors.

6.7 Discussion

We have proposed a novel learning-based method for automatic detection and localization of needles in US volumes. Our method localizes multiple orthogonal planes containing needle information. These planes are then used to determine the orthogonal plane containing the tip, and explicit tip localization. To the best of our knowledge, this is the first method to perform both plane and tip localization from 3D US data. The proposed method executes for ~3 s per volume. Although this is not a real-time rate, it is better than that reported in previous approaches [64,73]. Moreover, this rate can be improved with better computing hardware. Note also that the detection framework doesn't have to be continually run. Once the optimal orthogonal plane has been identified, this view can be maintained until the needle tip is out of view, ultimately saving computing cost.

We have demonstrated the advantage of deep learning for the challenge of needle detection in 3D data, as opposed to the use of a handcrafted feature-based classifier. Although both methods give similar accuracy, deep learning is superior in computational speed since it learns needle features end-to-end without prior need for feature extraction. The tip localization accuracy is competitive with that obtained from the 2D approaches (Chapter 2, 3 and 4). On account of including needle data from multiple slice, accurate tip localization is enhanced even when the needle is misaligned with the scan plane. The sufficiently high recall rate demonstrates that the detected volume always contains enough needle data to support the localization process.

Chapter 7 -Conclusion and Future Work

7.1 Summary and significance of research

In this dissertation, we have investigated techniques for improving 2D/3D US for more efficient and safer needle guidance during minimally invasive percutaneous procedures such as biopsies, regional anesthesia and peripheral vascular interventions. We have proposed several computational image-based methods for needle detection, enhancement and localization in 2D/3D US frames.

In **Chapter 1**, we discussed the clinical challenges in 2D US guided minimally invasive procedures, stemming from loss of needle visibility at certain depths and insertion angles. This results from difficulty in aligning the needle with the imaging plane, non-axial specular reflection, depth dependent signal attenuation and interfering artifacts. Since accurate needle localization is critical for procedure efficacy, loss of needle visibility could cause debilitating injury, mortality, increased hospital stays and increased medical costs. To motivate the methods we would employ to solve this challenge, we discussed the basics of US and the US image formation pipeline, and how these relate to needle visibility. We presented the current technological advances for improving needle localization in US: mechanical needle guides, needle tracking systems, robotic systems, 3D US, specialized needles and probes and software-based methods. We mentioned that these methods all have limitations: all hardware-based approaches add substantial cost to the basic US imaging system and disrupt the normal clinical workflow. Current software-based approaches on the other hand are not robust to

challenging imaging scenarios and do not achieve real-time speed. Our work was therefore premised on the need to develop more robust and accurate software-based techniques for needle localization in 2D/3D US. Our methods are presented to address different levels of needle visualization complexity and are complementary.

In **Chapter 2**, we introduced an approach for localizing needles in 2D US by reversing signal attenuation effects. This method targets in-plane inserted needles with a partially visible shaft but with the tip lacking a characteristic high intensity. First, we model US signal attenuation using penalized random walks in the image space. We use the attenuation model to derive a signal transmission map by optimizing an objective function cognizant of US imaging constraints. The transmission map is then utilized in a signal restoration model to enhance the needle shaft and tip. We also introduced a novel method for minimizing the effect of high intensity artifacts using a Tophat filter. From the enhanced needle image, the tip is localized using a combination of extracting local phase features using an orientation tuned band pass 2D Log-Gabor filter bank, trajectory estimation with the radon transform, line fitting of the resulting features with MLESAC and statistical optimization of the resulting colinear point cloud. This method achieved accurate needle tip localization and shaft enhancement and is the first of its kind to achieve needle localization by modeling US signal transmission in US. However, it is semi-automatic (it requires a priori knowledge of the needle insertion side, and a fixed ROI around a portion of the needle) and slow (~0.5 fps).

To make the method of Chapter 2 automatic and faster, we introduced the concept of deep learning using CNNs in **Chapter 3**. First, we presented a theoretical background to artificial neural networks, application of CNNs in image processing tasks, CNN building layers and the pertinent design considerations. We then presented a first of its kind CNN based approach for automatic localization of needles in 2D US. The framework, based on the Faster R-CNN architecture, automatically detects a portion of the needle shaft, and that information is used to automatically estimate the needle insertion side and approximate insertion angle. Needle tip localization then follows a similar process to that of Chapter 2, where local phase features are extracted and used to fit needle data along a line, from which the needle tip position is estimated. We achieved good tip localization accuracy. Like the method of Chapter 2, this method is also suitable for only in-plane inserted needles with some shaft information. By making the overall process automated, we solve one of the challenges in the method of Chapter 2. However, although the needle detection process runs at 25 fps on a GPU, the overall detection and localization step runs at ~2 fps. This is still far from real-time.

To further improve the speed of needle localization while addressing both in-plane and out-of-plane insertions, we presented a framework for enhancing and localizing the needle tip from dynamic 2D US (video) data in **Chapter 4**. The needle tip is enhanced from consecutive frames using a novel digital subtraction model that treats the tip as a moving target and its motion is detected from the foreground against a background constituted by the rest of the image. The enhanced tip is then augmented by solving a spatial total variation regularization problem using

the Split Bregman method. Lastly, we filter irrelevant motion events with a CNN learning framework based on the YOLO architecture. The learned model outputs bounding boxes around the enhanced tip, from which the needle tip location is estimated. The whole framework achieved good tip localization accuracy at a faster computation speed (10 fps). The fact that we do not need any shaft information in this approach makes it suitable for in-plane localization at steep insertion angles, as well as out-plane insertion.

Chapter 5 builds on the result of Chapter 4 and aimed at localizing both in-plane and out-of-plane needles in 2D US at a faster processing rate. Using a similar approach to the one of Chapter 4, we enhance the needle tip from consecutive images using digital subtraction. We then use a novel CNN framework to classify the enhanced image, and if the image contains the tip, to automatically localize it by regressing onto it. The approach gives accurate and real-time results: we achieve a computation rate of 67 fps. This is the fastest needle localization approach ever reported.

In **Chapter 6**, we extended the developed methods to automate needle detection and localization in 3D US. Using a Faster R-CNN detection model, we search a 3D volume and automatically classify the orthogonal planes containing needle information. The tip is then enhanced from a maximum intensity projection of the resulting sub volume. Tip localization follows using information derived from local phase features. Tip information is used to localize the best orthogonal plane for visualizing the needle tip. Although we did not achieve real-time performance,

our approach provides the fastest localization method for needles in 3D US reported so far.

7.2 Contributions

The main contributions of this dissertation are summarized as follows:

1) We introduced a novel semi-automatic and accurate method for enhancing steeply in-plane inserted needles based on modeling transmission of the 2D US signal. The derived transmission map is integrated in an image restoration model for enhancement of the needle shaft and tip while considering US specific signal propagation constraints. Needle localization is achieved by extracting intensity-invariant needle features from the enhanced image using Log-Gabor filters.
2) We presented an automatic and accurate learning-based framework for detection of steeply in-plane inserted low contrast needles in 2D US. The detection results are utilized for automatic determination of the needle trajectory and subsequently, tip localization from trajectory constrained features.
3) We developed a novel tip localization technique for both in-plane and out-of-plane inserted needles based on logical differencing of consecutive frames to detect subtle spatiotemporal variations in intensity arising from tip motion. From the difference image, the tip is augmented by solving a spatial total variation regularization problem using the Split Bregman method. This is followed by filtering irrelevant motion events with a

learning-based end-to-end data-driven method that models the appearance of the needle tip in US images.

4) We introduced a new design of a dual classification/regression learning-based framework for single-shot tip localization in 2D US image sequencies. By treating the needle tip as key point, and using a CNN based regression approach, we demonstrate real-time and accurate localization.
5) We developed a novel learning-based approach for needle enhancement and localization in 3D US. The original volume is shrunk to only slices containing needle information using a learned needle detection model. The model achieves high detection specificity and sensitivity. Enhancement of the needle in 3D space and localization of the best orthogonal plane is achieved using features derived from MIP.

7.3 Future work

We have presented techniques for robust, accurate and real-time localization of needles in 2D and 3D US. The goal is to incorporate these methods in a computational imaging platform integrated with US imaging systems. Such an imaging platform should:

a) Operate at real-time speed, i.e., ≥ 30 fps.
b) Have no platform dependence, i.e., it should work across different US imaging systems from different equipment manufacturers, with different image quality.

c) Support both 2D and 3D US transducers.

d) Accurately localize needles of different sizes.

e) Be robust to different needle insertion profiles with respect to depth and insertion angles typical of the target clinical procedures.

Toward this end, while the proposed methods achieve promising results, additional studies are required to further improve them to be ready for a clinical application. These are highlighted below:

Software optimization: Computational complexity is a key factor in any software-based platform for image guided interventions. Our methods were developed in MATLAB, a high-level proprietary programming environment. Although we achieved real-time rate for needle localization in 2D US, faster approaches for 3D localization should be studied. Certainly, the algorithms can be optimized and implemented in a more efficient programming language. Further, advances in computing power supported by GPUs promise to exponentially improve the speed of computing, especially with implementation of parallel computing. With the pertinent improvements in execution time, our methods would then be packaged into an application which can reside on the US machine, or communicate remotely with the US machine, for instance via transmission control protocol/Internet protocol (TCP/IP). The 2D methods are expected to perform at an execution speed of $30 - 50$ Hz while the 3D methods should achieve a rate of $10 - 20$ Hz. In the practical sense, current 3D transducers have not yet achieved this refresh rate, and therefore 3D US is not yet adopted for interventional support in clinics. However, we expect that improvements in 3D US technology will soon

make it possible for real-time applications in which our methods can be integrated. In fact, phased array 3D transducers such as the Philips X5-1 already incorporate electronic beam steering and orthogonal plane visualization. This makes such a system a prime candidate for integration of our smart search needle algorithm.

Better groundtruth: Localization errors should be investigated with more robust groundtruth approaches, such as an optical tracking system. The electromagnetic system we used in our studies had an inherent localization error, because the sensor in the needle doesn't go all the way to the tip and tracking was available on only one needle supported by the tracking system. This made this method of obtaining the groundtruth unreliable.

Animal studies: All our methods are developed and tested on offline ex vivo data collected on animal tissue phantoms. Therefore, additional clinical studies on animal models should be conducted to further evaluate online performance of the proposed methods and inform further improvements thereof. More so, animal studies would give an opportunity to study the effect of physiological events such as pulsation and breathing, which are expected to create artifacts in the US images that interfere with needle localization.

Cadaver studies: The final evaluation stage of the online system should involve testing performance of the needle localization methods on fresh cadavers. The cadaver would provide requisite anatomical complexity like what is expected in the clinical scenario. Therefore, the localization accuracy obtained would mimic what is expected during minimally invasive procedures. Additional localization error sources would be investigated and thereafter, mechanisms for mitigating them.

Portability: In our experiments, we investigated performance of our methods on data collected with a hand-held wireless US transducer, compatible with mobile phones, tablets and computers. The current trend in US is to make it more portable to suit Point of Care (PoC) applications. Some portable US manufacturers provide research interfaces through which researchers can integrate custom algorithms for clinical applications. Such platforms suit integration of our algorithms and would make our innovation suitable for adoption in low-income settings.

7.4 Other Applications

In this dissertation, we focused on hand-held needles and transducers. However, our methods can easily be integrated in robot-guided systems that employ US imaging. For example, our methods automatically determine needle trajectory, and location, information that could be used to give image assisted feedback to the robotic system for more efficient and accurate targeting.

Further, we focused on localization of needles, the work we have presented avails many promising directions. For example, the same techniques can be used for localization of other percutaneously inserted devices such as guidewires, catheters, and atherectomy devices.

Other US imaging modalities such as Transesophageal echo (TEE) US and Transrectal US (TRUS) can also benefit from our methods. For instance, TEE US is used to guide transseptal punctures in which a transseptal needle is used. Localization of the transseptal needle is difficult in a beating heart. We anticipate that our methods could contribute to addressing this challenge while also testing

the robustness in anatomy and procedures where there is drastic motion. Regarding TRUS, the commonest procedures are TRUS guided biopsies, where the biopsy needle is fired via a gun. This is a good candidate for our needle localization approaches because we are only interested in localizing a needle that is fired into a location but is not in continuous motion.

Lastly, other imaging modalities such as fluoroscopy and endoscopy, in which video information is available, and are also used for guidance of interventional devices could benefit from our algorithms especially the ones reported in Chapters 4 and 5 for surgical tool localization from sequential US frames. For instance, our method could facilitate accurate placement of pedicle screws in spinal interventions.

BIBLIOGRAPHY

1. Khati N, Gorodenker J, Hill M (2011). Ultrasound-guided Biopsies of the Abdomen. Ultrasound Q, 27(4): 255-268.

2. Korbe S, Udoji E, Ness T, Udoji M (2015). Ultrasound-guided interventional procedures for chronic pain management. Pain Management, 5(6):466-482.

3. Dietrich CF, Horn R, Morf S, Chiorean L, Dong Y, Cui XW, Atkinson N, Jenssen C (2016) US-guided peripheral vascular interventions, comments on the EFSUMB guidelines. Med Ultrason, 18(2):231-239.

4. Kinsella S, Young N (2009) Ultrasound-Guided Central Line Placement as Compared with Standard Landmark Technique: Some Unpleasant Arithmetic for the Economics of Medical Innovation. Value in Health, 12(1):98-100.

5. Nawapun K, Phithakwatchara N, Jaingam S, Viboonchart S, Mongkolchat N, Wataganara T (2018) Advanced ultrasound for prenatal interventions. Ultrason, 37:200-210

6. Copelan A, Scola D, Roy A, Nghiem HV (2016) The Myriad Advantages of Ultrasonography in Image-Guided Interventions. Ultrasound Quarterly, 32(3):247-57.

7. Sheafor DH, Paulson EK, Simmons CM, DeLong DM, Nelson RC (1998) Abdominal percutaneous interventional procedures: comparison of CT and US guidance. Radiology, 207:705–710.

8. Brass P, Hellmich M, Kolodziej L, Schick G, Smith AF (2015) Ultrasound guidance versus anatomical landmarks for subclavian or femoral vein catheterization. Cochrane Database of Systematic Reviews, 1: CD011447.

9. Liu SS Ngeow J, John RS (2010) Evidence basis for ultrasound-guided block characteristics: onset, quality, and duration. , 35(2 Suppl):S26-35

10. Barrington MJ, Kluger R. Ultrasound guidance reduces the risk of local anesthetic systemic toxicity following peripheral nerve blockade. Reg Anesth Pain Med, 38(4): 289–99.

11. Bianchi S, Zamorani MP (2007) US-guided interventional procedures. In: Bianchi S, Martinoli C (eds) Ultrasound of the musculoskeletal system, 1st edn. Springer, Berlin, 891–917

12. Chin KJ, Perlas A,Chan VWS, Brull R (2008) Needle visualization in ultrasound-guided regional anesthesia: challenges and solutions. Reg Anesth Pain Med 33(6):532–544

13. Rathmell, JP, Benzon HT, Dreyfuss P, Huntoon M, Wallace M et al (2015) Safeguards to Prevent Neurologic Complications after Epidural Steroid Injections: Consensus Opinions from a Multidisciplinary Working Group and National Organizations. Anesthesiology 122: 974-984.

14. Pathak R, Karmacharya P, Aryal MR, Alweis R (2014) Iatrogenesis imperfecta: stroke caused by accidental carotid artery catheterization. J Vasc Access, 15(6): 537–40.

15. Burckhardt CB (1978) Speckle in ultrasound B-mode scans. IEEE Trans. On Sonics and Ultrasonics, 25(1):1–6.

16. Prasad N, Kumar S, Manjunath R, Bhadauria D, Kaul A, Sharma R, Gupta A, Lal H, Jain M, Agrawal,V (2015). Real-time ultrasound-guided percutaneous renal biopsy with needle guide by nephrologists decreases post-biopsy complications. Clin Kidney J, 8(2): 151-156.

17. Elsharkawy H, Babazade R, Kolli S, Kalagara H, Soliman ML (2016) The Infiniti plus ultrasound needle guidance system improves needle visualization during the placement of spinal anesthesia. Korean J Anesthesiol, 69(4): 417-419.

18. Tsui, BC (2007) Facilitating needle alignment in-plane to an ultrasound beam using a portable laser unit. Reg Anesth Pain Med, 32:84-88.

19. Stolka PJ, Foroughi P, Rendina M, Weiss CR, Hager GD, Boctor EM (2014) Needle Guidance Using Handheld Stereo Vision and Projection for Ultrasound-Based Interventions. MICCAI LNCS, 8674: 684-691.

20. Krücker J, Xu S, Glossop N, Viswanathan A, Borgert J, Schulz H, Wood BJ (2007) Electromagnetic Tracking for Thermal Ablation and Biopsy Guidance: Clinical Evaluation of Spatial Accuracy. J Vasc Interv Radiol, 18(9): 1141-50.

21. Hui Z, Banovac F, Lin R, Glossop N, Wood B, Lindisch D, Levy E, Cleary K (2006) Electromagnetic Tracking for Abdominal Interventions in Computer Aided Surgery. Computer Aided Surgery 11(3): 127-36.

22. Fevre MC, Vincent C, Picard J, Vighetti A, Chapuis C, Detavernier M, Allenet B, Payen JF, Bosson JL, Albaladejo P (2018) Reduced variability and execution time to reach a target with a needle GPS system: Comparison between radiologists, residents and nurse anaesthetists. Anaesth Crit Care Pain Med, 37(1):55-60.

23. Desjardins AE, Hendriks BH, Van Der Voort M, Nachabé R, Bierhoff W, Braun G, Babic D, Rathmell JP, Holmin S, Söderman M, Holmström B (2011) Epidural needle with embedded optical fibers for spectroscopic differentiation of tissue: ex vivo feasibility study. Biomed Opt Express, 2(6):1452-1461.

24. Desjardins AE, Van Der Voort M, Roggeveen S, Lucassen G, Hendriks BH, Brynolf M, Holmström B (2011) Needle stylet with integrated optical fibers for spectroscopic contrast during peripheral nerve blocks. J Biomed Opt., 16(7):077004.

25. Shen Z, Zhou Y, Miao J, Vu KF (2015). Enhanced visualization of fine needles under sonographic guidance using a MEMS actuator. Sensors, 15: 3107–15.

26. Mung J, Vignon F, Jain A (2011). A non-disruptive technology for robust 3D tool tracking for ultrasound-guided interventions. MICCAI LNCS, 6891: 153–60.

27. Lu H, Li J,Lu Q, Bharat S, Erkamp R, Chen B, Drysdale J, Vignon F, Jain A (2014) A new sensor technology for 2D ultrasound-guided needle tracking. MICCAI LNCS: 389-396.

28. Xia W, West S, Finlay M, Mari J, Ourselin S, David A, Desjardins A (2017) Looking beyond the imaging plane: 3D needle tracking with a linear array ultrasound probe. Scientific Reports, 7(1):3674.

29. Hebard S, Graham H. (2011) Echogenic technology can improve needle visibility during ultrasound-guided regional anesthesia. Reg. Anesth. Pain Med, 36(2), 185–189.

30. Deam RK, Kluger R, Barrington MJ, McCutcheon CA (2007). Investigation of a new echogenic needle for use with ultrasound peripheral nerve blocks. Anaesth Intensive Care, 35:582–586.

31. Menhadji A, Nguyen V, Cho J, Chu R, Osann K, Bucur P, Patel P, Lusch A, McDougall E, Landman J (2013). In vitro comparison of a novel facilitated ultrasound targeting technology vs standard technique for percutaneous renal biopsy. Urology, 82:734–737.

32. Sviggum HP, Ahn K, Dilger JA, Smith H, Sviggum H, Ahn K, Dilger J, Smith H (2013) Needle echogenicity in sonographically guided regional anesthesia: Blinded comparison of 4 enhanced needles and validation of visual criteria for evaluation. J Ultrasound Med, 32: 143–148.

33. Hopkins RE, Bradley M (2001) In-vitro visualization of biopsy needles with ultrasound: A comparative study of standard and echogenic needles using an ultrasound phantom. Clin Radiol, 56:499-502.

34. Nichols K, Wright LB., Spencer T, Culp WC (2003). Changes in ultrasonographic echogenicity and visibility of needles with changes in angles of insonation. J Vasc Interv Radiol, 14:1553-1557.

35. Mignon P, Poignet P, Troccaz J (2018) Automatic Robotic Steering of Flexible Needles from 3D Ultrasound Images in Phantoms and Ex Vivo Biological Tissue. Ann Biomed Eng, 46(9):1385-1396

36. Adebar TK, Fletcher AE, Okamura AM (2014) 3D ultrasound-guided robotic needle steering in biological tissue. IEEE Trans Biomed Eng, 61(12):2899–2910

37. Priester AM, Natarajan S, Culjat MO (2013) Robotic ultrasound systems in medicine. IEEE EEE Trans Ultrason Ferroelectr Freq Control, 60: 507-523

38. Neubach Z, Shoham M (2010). Ultrasound-guided robot for flexible needle steering. IEEE Trans. Biomed. Eng, 57:799–805.

39. Boctor EM, Choti MA, Burdette EC, Webster RJ (2008) Three-dimensional ultrasound-guided robotic needle placement: an experimental evaluation. Int J Med Robot Comput Assist Surg 4(2):180–91

40. Xu S, Krueckern J, Jiang H, Settlemier S, Glossop N, Venkatesan A, Wood B. (2008) 3D ultrasound guidance system for needle placement procedures. In Progress in Biomedical Optics and Imaging - Proceedings of SPIE (Vol. 6918).

41. Chang H, Chen Z, Huang Q, Shi J, Li X (2015) Graph-based learning for segmentation of 3D ultrasound images, Neurocomputing, 151(2):632–644.

42. Clendenen SR, Robards CB, Clendenen NJ, Freidenstein JE, Greengrass RA (2010) Real-Time 3-Dimensional Ultrasound-Assisted Infraclavicular Brachial Plexus Catheter Placement: Implications of a New Technology. Anesthesiology Research and Practice, 1-4.

43. Huang Q, Zeng Z (2017) A Review on Real-Time 3D Ultrasound Imaging Technology. BioMed Research International, 2017, 6027029

44. Fenster A, Downey DB (1996) 3-D ultrasound imaging: a review, IEEE Eng Med Biol Mag, 15(6): 41–51

45. Prager RW, Ijaz UZ, Gee AH, Treece GM (2010) Three-dimensional ultrasound imaging. Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine, 224(2): 193–223.

46. Smith SW, Pavy HG, von Ramm OT (1991) High-speed ultrasound volumetric imaging system. I. Transducer design and beam steering, IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control, 38(2):100–108

47. Démoré CEM, Joyce AW, Wall K, Lockwood GR (2009) Real-time volume imaging using a crossed electrode array, IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control, 56(6): 1252–1261

48. Gebhard RE, Eubanks TN, Meeks R (2015). Three-dimensional ultrasound imaging. Current Opinion in Anaesthesiology, 28(5): 583–7.

49. Cohnen M, Saleh A, Lüthen R, Bode J, Mödder U (2003) Improvement of sonographic needle visibility in cirrhotic livers during transjugular intrahepatic portosystemic stent-shunt procedures with use of real-time compound imaging. J Vasc Interv Radiol, 14:103-106.

50. Zhuang B, Dickie K, Pelissier L (2013) In vivo needle visualization in ultrasound images using tensor-based filtering. IEEE Ultrason Symp, 667–670.

51. Baker JA, Soo MS, Mengoni P (1999) Sonographically guided percutaneous interventions of the breast using a steerable ultrasound beam. AJR Am J Roentgenol, 172:157-159.

52. Cheung S, Rohling R (2004) Enhancement of needle visibility in ultrasound-guided percutaneous procedures. Ultrasound Med Biol, 30(5):617–624.

53. Harmat A, Rohling RN, Salcudean S (2006) Needle tip localization using stylet vibration Ultr in Med and Biol, 32(9), 1339-1348.

54. Greer JD, Adebar TK, Hwang GL, Okamura AM (2014). Real-time 3D curved needle segmentation using combined B-mode and power doppler ultrasound. In Medical Imaging Computing and Computer Assisted Intervention (MICCAI), 17(Pt 2):381-8

55. Qui W, Yuchi M, Ding M, Fenster A, Tessier D (2014) Needle segmentation using 3D Hough transform in 3D TRUS guided prostate transperineal therapy. Med. Phys. 40 (4): 042902.

56. Elif A, Jaydev P (2014) Optical Flow-Based Tracking of Needles and Needle-Tip Localization Using Circular Hough Transform in Ultrasound Images. Annals of Biomedical Engineering, 43(8): 1828–1840.

57. Okazawa S, Ebrahimi R, Chuang J, Rohling R, Salcudean S (2006) Methods for segmenting curved needles in ultrasound images. Med Image Anal, 10(3):330–42.

58. Ding M, Fenster A (2003). A real-time biopsy needle segmentation technique using Hough transform. Med Phys, 30(8): 2222-33.

59. Cool D, Gardi L, Romagnoli C, Saikaly M, Izawa J, Fenster A (2010). Temporal-based needle segmentation algorithm for transrectal ultrasound prostate biopsy procedures. Med Phys, 37(4):1660-73.

60. Wu Q, Yuchi M, Ding M (2014) Phase Grouping-Based Needle Segmentation in 3-D Trans-rectal Ultrasound-Guided Prostate Trans-perineal Therapy.Ultrasound in Med. and Biol., 40 (4), 804-816.

61. Novotny PM, Stoll JA, Vasilyev NV, Del Nido PJ, Dupont PE, Zickler TE, Howe RD (2007) GPU based real-time instrument tracking with three-dimensional ultrasound. Med Image Anal 2007;11(5):458–64.

62. Barva M, Uhercik M, Mari JM, Kybic J, Duhamel JR, Liebgott H, Hlavac V, Cachard C (2008) Parallel integral projection transform for straight electrode localization in 3-D ultrasound images. IEEE Trans Ultrason Ferroelectr Freq Control, 55(7):1559–1569

63. Ding M, Cardinal H, Fenster A (2003). Automatic needle segmentation in three-dimensional ultrasound images using two orthogonal two-dimensional image projections. Med Phys 30(2):222-34.

64. Beigi P, Rohling R, Salcudean T, Lessoway VA, Ng GC (2015) Needle Trajectory and Tip Localization in Real-Time 3-D Ultrasound Using a Moving Stylus. Ultrasound Med Biol, 41(7):2057-70

65. Gaufillet F, Liegbott H, Uhercik M, Cervenansky F, Kybic J, Cachard C (2010) 3D Ultrasound real-time monitoring of surgical tools. Ultrasonics Symposium (IUS), IEEE.

66. Uhercik M, Kybic J, Liebgott H, Cachard C (2010) Model fitting using RANSAC for surgical tool localization in 3-D ultrasound images. IEEE Trans Biomed Eng, 57(8):1907–16.

67. Zhao Y, Cachard C, Liebgott H (2013) Automatic Needle Detection and Tracking in 3D Ultrasound Using an ROI-Based RANSAC and Kalman Method. Ultrasonic Imaging, 35(4):283-306.

68. Hacihaliloglu I, Ng G, Rohling RN, Salcudean S, Abolmaesumi P (2015) Projection-Based Phase Features for Localization of a Needle Tip in 2D Curvilinear Ultrasound. MICCAI LNCS, 9349: 347-354.

69. Beigi P, Rohling R, Salcudean SE, Ng GC (2016) Spectral analysis of the tremor motion for needle detection in curvilinear ultrasound via spatiotemporal linear sampling Int J CARS 11(6):1183-1192.

70. Hatt C. R., Ng G., Parthasarathy V.: Enhanced needle localization in ultrasound using beam steering and learning-based segmentation. Comp. Med. Imag. and Grap., 14, 45-54, (2015).

71. Beigi P, Rohling R, Salcudean SE, Ng GC (2017) CASPER: computer-aided segmentation of imperceptible motion-a learning-based tracking of an invisible needle in ultrasound. Int J CARS, 12(11):1857-1866.

72. Pourtaherian A, Ghazvinian Zanjani F, Zinger S, Mihajlovic N, Ng G, Korsten H, With P (2017) Improving Needle Detection in 3D Ultrasound Using Orthogonal-Plane Convolutional Networks. MICCAI LNCS 10434: 610-618.

73. Pourtaherian A, Ghazvinian Zanjani F, Zinger S, Mihajlovic N, Ng G, Korsten H, With P (2018) Robust and semantic needle detection in 3D ultrasound using orthogonal-plane convolutional neural networks. Int J CARS 13(9):1321-1333.

74. Shen D, Wu G, Suk H (2017) Deep learning in medical image analysis. Annu Rev Biomed Eng. 19: 221-248.

75. Leroy A, Mozer P, Payan Y, Troccaz J (2004) Rigid registration of freehand 3D ultrasound and CT-scan kidney images. MICCAI LNCS, 3216: 837–844.

76. Karamalis A, Wein W, Klein T, Navab N (2012) Ultrasound Confidence Maps using Random Walks. Medical Image Analysis, 16(6):1101-1112.

77. Grady L (2006) Random walks for image segmentation. IEEE Trans. Pattern Anal. Mach. Intell, 28(11): 1768–1783.

78. Sheet D, Karamalis A, Eslami A, Noel P, Virmani R, Nakano M, Chatterjee J, Ray AK, Laine AF, Carlier SG, Navab N, Katouzian A (2014) Hunting for necrosis in the shadows of intravascular ultrasound. Comput.Med. Imag.Grap, 38(2): 104–112.

79. Radlak K, Radlak N, Smolka B (2015) Automatic detection of bones based on the confidence map for Rheumatoid Arthritis analysis. Computational Vision and Medical Image Processing V Proceedings of the 5th Eccomas Thematic Conference on Computational Vision and Medical Image Processing (VipIMAGE 2015, Tenerife, Spain, October 19-21, 2015), 215-220.

80. Quader, N., Hodgson, A., & Abugharbieh, R. (2014). Confidence Weighted Local Phase Features for Robust Bone Surface Segmentation in Ultrasound. Clinical Image-Based Procedures. Translational Research in Medical Imaging Lecture Notes in Computer Science, 76-83.

81. Hacihaliloglu I, Abugharbieh R, Hodgson AJ, Rohling R (2008) Bone segmentation and fracture detection in ultrasound using 3D local phase features. MICCAI LNCS, 5241: 287–295.

82. Fu H, Ng MK, Nikolova M, Barlow JL (2006). Efficient minimization methods of mixed L2-L1 and L1-L1 norms for image restoration. Siam J. Sci. Comput, 27(6), 1881–1902

83. Howard AL. Review: Tikhonov A, Arsenin V (1977) Solution of Ill-Posed Problems. Bull. Amer. Math. Soc. (N.S.), 1(3): 521-524

84. Chan T, Esedoglu S (2004) Aspects of Total Variation Regularized L1 Function Approximation, Technical report, University of California at Los Angeles.

85. K¨arkk¨ainen T, Kunisch K, Majava K (2005) Denoising of smooth images using L1-fitting. Computing, 74(4): 353–376.

86. Kuo S, Mammone R (1992) Image restoration by convex projections using adaptive constraints and the L1 norm. IEEE Trans. Signal Process. 40(1):1159–1168.

87. Miyashita T (1992) Super-resolved image restoration of holographic images by L1 norm minimization with clutter rejection. Acoustical Imaging, 19:77–82.

88. Nikolova M (2004) A variational approach to remove outliers and impulse noise. J. Math. Imaging Vision, 20 (1-2): 99–120.

89. Kirsch, R (1971). Computer determination of the constituent structure of biological images. Computers and Biomedical Research. 4: 315-328.

90. Meng G, Wang Y, Duan J, Xiang S, Pan C (2013): Efficient Image Dehazing with Boundary Constraint and Contextual Regularization. In: IEEE International Conference on Computer Vision, 617-624.

91. Torr PHS, Zisserman A (2000). MLESAC: A New Robust Estimator with Application to Estimating Image Geometry. J Comput Vis Image Und, 78(1):138-156.

92. Lundervold AS, Lundervold A (2018) An overview of deep learning in medical imaging focusing on MRI. arXiv:1811.10052v2

93. De Bruijne M (2016) Machine learning approaches in medical image analysis: from detection to diagnosis. Med Image Anal, 33:94–7.

94. Wernick MN, Yang Y, Brankov JG, Yourganov G, Strother SC (2010) Machine learning in medical imaging. IEEE Signal Process Mag, 27:25–38.

95. LeCun Y, Bengio Y, Hinton G (2015). Deep learning. Nature 521, 436:444

96. Raissi M, Karniadakis GE (2018) Hidden physics models: Machine learning of nonlinear partial differential equations, Journal of Computational Physics 357 (2018) 125–141.

97. Glorot X, Bordes A, Bengio Y (2011) Deep sparse rectifier neural networks. In Proc. 14th International Conference on Artificial Intelligence and Statistics, 315-323.

98. Xu B, Wang N, Chen T, Li M (2015) Emphirical evaluation of rectified activations in convolutional network. arXiv:1505.00853

99. Clevert D, Unterthiner T, Hochreiter S (2016) Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs). arXiv:1511.07289

100. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: a simple way to prevent neural networks from overfitting. J. Machine Learning Res, 15:1929-1958.

101. Deng J, Dong W, Socher R, Li LJ, Li K, Fei-Fei L (2009) ImageNet: A Large-Scale Hierarchical Image Database. CVPR09

102. Yosinski J, Clune J, Bengio Y, and Lipson H. How transferable are features in deep neural networks? In Advances in Neural Information Processing Systems 27 (NIPS ’14), NIPS Foundation, 2014

103. Ioffe S, Szegedy C (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. arXiv:1502.03167

104. Samala RK, Chan HP, Hadjiiski LM, Helvie MA, Cha KH, Richter CD (2017) Multi-task transfer learning deep convolutional neural network: application to computer-aided diagnosis of breast cancer on mammograms., Physics in medicine and biology, 62:8894-890

105. Mohsen H, El-Dahshan EA, El-Horbaty EM, Salem AM (2018) Classification using deep learning neural networks for brain tumors. Future Computing and Informatics, 3(1): 68-71.

106. Hwang YN, Lee JH, Kim GY, Jiang YY, Kim SM (2015) Classification of focal liver lesions on ultrasound images by extracting hybrid textural features and using an artificial neural network. Bio-Medical Materials and Engineering 26: S1599- S1611.

107. Twinanda AP, Shehata S, Mutter D, Marescaux J, de Mathelin M, Padoy N (2017), Endonet: A deep architecture for recognition tasks on laparoscopic videos, IEEE transactions on medical imaging, 36(1):86-97.

108. Krizhevsky A, Sutskever I, Hinto GE (2012) ImageNet Classification with Deep Convolutional Neural Networks. NIPS 2012, 1097-1105

109. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556

110. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2014) Going deeper with convolutions. arXiv:1409.4842

111. He K, Zhang X, Ren S, Sun J (2015). Deep residual learning for image recognition. arXiv:1512.03385v1

112. Long J, Shelhamer E, Darrell T (2015) Fully convolutional networks for semantic segmentation. IEEE CVPR, 3431-3440.

113. Roth HR, Shen C, Oda H, Oda M, Hayashi Y, Misawa K, Mori K (2018) Deep learning and its application to medical image segmentation. arXiv:1803.08691v1

114. Ronneberger O, Fischer P, Brox T (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. MICCAI LNCS, 9351: 234-241.

115. Milletari F, Navab N, Ahmadi S (2016) V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation. arXiv:1606.04797v1.

116. Girshick R, Donahue J, Darrell T, Malik J (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. arXiv:1311.2524v5

117. Girshick R (2015) Fast R-CNN. IEEE Int. Conf. Comput. Vis, 1440-48.

118. Ren S, He K, Girshick R, Sun J (2017) Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. IEEE Trans. Pattern Anal. Mach. Intell, 39(6):1137-49.

119. Redmon J, Divvala S, Girshick R, Farhadi A (2015) You Only Look Once: Unified, Real-Time Object Detection. arXiv:1506.02640

120. Redmon J, Farhadi A (2016) YOLO9000: Better, Faster, Stronger. arXiv:1612.08242

121. Kendall A, Grimes M, Cipolla R (2015) PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization. arXiv:1505.07427v4

122. Haskins G, Kruecker J, Kruger U, Xu S, Pinto PA, Wood BJ, Yan P (2018) Learning deep similarity metric for 3D MR-TRUS registration. arXiv:1806.04548v1.

123. Cao X, Yang J, Zhang J, Wang Q, Yap PT, Shen D (2018) Deformable image registration using a cueaware deep regression network, IEEE transactions on bio-medical engineering 65:1900-1911.

124. Zheng J, Miao S, Wang ZJ, Liao R (2018) Pairwise domain adaptation module for CNN-based 2-D/3-D registration. J Med Imaging (Bellingham). 5(2):021204.

125. Rumelhart DE, Hinton GE, Williams RJ (1986) Learning representations by back-propagating errors. Nature 323: 533-536.

126. Cauchy A (1847) M´ethode g´en´erale pour la r´esolution des systemes d´equations simultan´ees, Comp. Rend. Sci. Paris, 25:536-538.

127. Ride S (2017) An overview of gradient descent optimization algorithms. arXiv:1609.04747v2

128. Qian N(1999) On the momentum term in gradient descent learning algorithms. Neural networks: the official journal of the International Neural Network Society, 12(1): 145-151.

129. Duchi J, Hazan E, Singer Y(2011) Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. Journal of Machine Learning Research, 12: 2121-2159.

130. Zeiler MD (2012). ADADELTA: An Adaptive Learning Rate Method. arXiv:1212.5701.

131. Kingma DP, Ba JL (2015) Adam: a Method for Stochastic Optimization. arXiv:1412.6980

132. Tensorflow: https://www.tensorflow.org/

133. Keras: https://keras.io/

134. PyTorch: https://pytorch.org/

135. Caffe: http://caffe.berkeleyvision.org/

136. Theano: http://deeplearning.net/software/theano/

137. Zeiler MD, Fergus R (2013) Visualizing and understanding convolutional neural networks. arXiv:1311.2901v3 .

138. Krizhevsky A. Learning Multiple Layers of Features from Tiny Images. CIFAR10 dataset, 2009.

139. Mwikirize C, Nosher JL, Hacihaliloglu I (2016) Enhancement of Needle Tip and Shaft from 2D Ultrasound Using Signal Transmission Maps. In: Ourselin S, Joskowicz L, Sabuncu M, Unal G, Wells W (eds) MICCAI LNCS, 9900: 362-369.

140. Mwikirize C, Nosher JL, Hacihaliloglu I (2018) Signal attenuation maps for needle enhancement and localization in 2D ultrasound. Int J CARS, 13(3): 363-374.

141. Ayvaci A, Yan P, Xu S, Soatto S, Kruecker J (2011). Biopsy needle detection in transrectal ultrasound. Comput Med Imaging Graph. 35(7-8):653-9.

142. Kolmogorov V, Zabih R (2004) What energy functions can be minimized via graph cuts?, IEEE Trans. Pattern Analysis and Machine Intelligence 26 (2) (2004) 147–159

143. Ayvali E, Desai J (2014) Optical Flow-Based Tracking of Needles and Needle-Tip Localization Using Circular Hough Transform in Ultrasound Images.

144. Afonso M, Bioucas-Dias J, Figueiredo M (2010) Fast image recovery using variable splitting and constrained optimization. IEEE Trans Image Process 19(9): 2345-2356.

145. Chan S, Khoshabeh R, Gibson K, Gill P, Nguyen T (2011) An Augmented Lagrangian Method for Total Variation Video Restoration. IEEE Trans Image Process 20(11): 3097-3111.

146. Goldstein T, Osher S (2009) The Split Bregman Method for L1-Regularized Problems. SIAM Journal on Imaging Sciences 2(2): 323-343.

147. Fong D, Saunders M (2011) LSMR: An Iterative Algorithm for Sparse Least-Squares Problems. SIAM Journal on Scientific Computing, 33(5): 2950-2971.

148. Everingham M, Gool LV, Williams CKI, Winn J, Zisserman A (2010) The PASCAL Visual Object Classes (VOC) Challenge. Int J Comput Vis 88: 303-338.

149. Everingham M, Gool LV, Williams CKI, Winn J, Zisserman A (2010) The PASCAL Visual Object Classes (VOC) Challenge. Int J Comput Vis 88: 303-338.

150. Mwikirize C, Nosher JL, Hacihaliloglu I (2018). Convolution Neural Networks for Real-Time Needle Detection and Localization in 2D Ultrasound. Int J CARS, 13 (5), 647-657.

151. Kendall A, Grimes M, Cipolla R (2015). PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization. arXiv:1505.07427v4

152. Kendall A, Cipolla R (2017). Geometric loss functions for camera pose regression with deep learning. arXiv:1704.00390v2.

153. Xiang Y, Schmidt T, Narayanan V, Fox D (2017) PoseCNN: A Convolutional Neural Network for 6D Object Pose Estimation in Cluttered Scenes. arXiv:1711.00199v3

154. Liu X, Liang W, Wang Y, Li S, Pei M (2016) 3D head pose estimation with convolutional neural network trained on synthetic images. ICIP:1289-1293.

155. Toshev A, Szegedy C (2014) DeepPose: Human Pose Estimation via Deep Neural Networks. arXiv:1312.4659v3

156. Guler RA, Neverova N, Kokkinos I (2018) DensePose: Dense Human Pose Estimation In The Wild. arXiv:1802.00434v1

157. Sun Y, Wang X, Tang X (2013) Deep convolutional network cascade for facial point detection. CVPR.

158. Kügler D, Stefanov A, Mukhopadhyay A. i3PosNet: Instrument Pose Estimation from X-Ray. arXiv:1802.09575v1

159. Cao X, Yang J, Zhang J, Nie D, Kim MJ, Wang Q, Shen D (2017) Deformable Image Registration based on Similarity-Steered CNN Regression. MICCAI, 10433: 300-308

160. Miao S, Wang Z, Liao R (2016) A CNN Regression Approach for Real-Time 2D/3D Registration. IEEE Trans Med Imaging. 35(5):1352-1363.

161. Haskins G, Kruecker J, Kruger U, Xu S, Pinto PA, Wood BJ, Yan P (2018) Learning deep similarity metric for 3D MR-TRUS registration. arXiv:1806.04548v1.

162. Cao X, Yang J, Zhang J, Wang Q, Yap PT, Shen D (2018) Deformable image registration using a cue-aware deep regression network. IEEE Trans Bio-med Eng, 65:1900-1911

163. Agarwal N, Krohn-Grimberghe A, Vyas R (2017). Facial Key Points Detection using Deep Convolutional Neural Network – NaimishNet. arXiv: arXiv:1710.00977v1.

164. Zhao Y, Bernard A, Cachard C, Liebgott H (2014) Biopsy needle localization and tracking using ROI-RK method. Abstr Appl Anal: 973147

165. Dalal N, Triggs B (2005) Histograms of Oriented Gradients for Human Detection. IEEE CVPR.