Robotic assistance system based on ultrasound images for regional anesthesia

Mohammad Alkhatib

To cite this version:

Mohammad Alkhatib. Robotic assistance system based on ultrasound images for regional anesthesia. Signal and Image processing. Université d'Orléans, 2020. English. NNT : 2020ORLE3186 . tel-03952569

HAL Id: tel-03952569
https://theses.hal.science/tel-03952569

Submitted on 23 Jan 2023

UNIVERSITÉ D'ORLÉANS

ÉCOLE DOCTORALE
MATHÉMATIQUES, INFORMATIQUE, PHYSIQUE THÉORIQUE ET INGÉNIERIE DES SYSTÈMES

Laboratoire PRISME

THÈSE présentée par :

Mohammad ALKHATIB

soutenue le : **17 Decembre 2020**

pour obtenir le grade de : **Docteur de l'université d'Orléans**
Discipline/ Spécialité : Robotique et Image Vision

Robotic assistance system based on ultrasound images analysis for regional anesthesia

---

THÈSE dirigée par :

Pierre VIEYRES	Professeur des universités, Université d'Orléans
Adel HAFIANE	Maître de conférences - HDR, INSA Centre Val-de-Loire

Co-Encadrée par :

Cyril NOVALES	Maître de conférences, Université d'Orléans

RAPPORTEURS :

Alexandre KRUPA	Chargé de Recherche, Inria Rennes-Bretagne Atlantique
Didier VRAY	Professeur des universités, creatis INSA-lyon

JURY:

Alexandre KRUPA	Chargé de Recherche, Inria Rennes-Bretagne Atlantique
Didier VRAY	Professeur des universités, creatis INSA-lyon
Veronique PERDEREAU	Professeur des universités, Sorbonne Université - Présidente du jury
Su RUAN	Professeur des universités, Université de Rouen
Arnaud LELEVÉ	Associate Professeur, INSA-lyon
Cyril NOVALES	Maître de conférences, Université d'Orléans
Adel HAFIANE	Maître de conférences - HDR, INSA Centre Val-de-Loire
Pierre VIEYRES	Professeur des universités, Université d'Orléans

INVITÉ:

Alain DELBOS	Practicien hospitalier, Clinique Médipole Gâronne

Contents

List of Figures

List of Tables

General Introduction

Regional anesthesia (RA) is performed to block the sensation of pain in a specific region of the body by stopping the nerve impulses connection between that region and the central nervous system. Nowadays, RA is a well-known procedure in many operating rooms and is used to avoid performing general anesthesia. RA is performed by the anesthetist in order to reduce pain scores, facilitates earlier hospital discharge, and improves postoperative mobility. Traditionally, this technique was performed with needle blind guidance which increases the risk of block failure, nerve trauma, and local anesthetic toxicity [1]. Hence, Ultrasound-guided regional anesthesia (UGRA) becomes a powerful technique in surgical procedure and pain management [2].

UGRA offers a huge performance impact on the practice of regional anesthesia [3, 4]. However, when performing UGRA, experts face several challenges: finding the nerve location in the poor quality ultrasound (US) image; keeping the nerve visualization all the time; locating different organs in the image such as arteries and bones; finding the best needle insertion point; controlling the needle insertion by taking into account the interaction pivot point between the needle and the skin of the patient; aligning the needle within the 2D US plane; finding the suitable needle trajectory to avoid sensitive anatomical or cardiovascular regions (arteries, bones, etc). Hence, UGRA procedure requires a long learning process and years of experience [5, 6].

Robotic systems are already used in medical settings and have shown their important impacts in assisting the medical experts during their medical act [7]. Robotic-assisted UGRA could provide great assistance by helping the experts with techniques and tools that improve the procedure accuracy and safety such as avoiding nerve trauma or damage of healthy tissues. Moreover, it could increase the anesthetist's experience by more teaching and learning to avoid unintentional injuries [8]. The main objective of this thesis work is to provide anesthetists with robotic and image processing tools to assist and facilitate their UGRA routine for patient health benefits.

There are two critical issues in the robotic-assisted UGRA system: the visual quality of the US image which suffers from the presence of noises; and the robotic control and accuracy in the needle insertion. To overcome these difficulties and to perform an accurate robotic-assisted UGRA system, this thesis aims to develop a robotic-assisted UGRA system that focuses on these two issues and proposes nerve detection and tracking methods and robotic needle insertion control.

Chapter 1 presents the UGRA procedure, difficulties, challenges, and limitations as performed *by anesthetists. As well as, analyzing the nerve structure and characteristics. Finally, the proposed robotic-assisted UGRA system and the objectives of this thesis are explained.*

In Chapter 2, the US image is analyzed by highlighting its noisy texture properties and the importance of finding robust and discriminative features in the image. In this chapter, we propose a new robust feature Robust Adaptive Median Binary Pattern (RAMBP). RAMBP handles images with highly noisy textures and increases the discriminative properties by capturing microstructure and macrostructure texture information.

Nerve detection and tracking in the US images is proposed in Chapter 3. RAMBP descriptor is used with well-known detectors and tracker. However, we believe Convolutional Neural Networks (CNNs) models are effective too for the ultrasound image textures. Thus, to learn more optimal features, we propose to merge RAMBP with CNN models to achieve better performance for nerve detection and tracking. In Chapter 3, the texture descriptor, CNNs models, and the merged model detectors and trackers are evaluated and analyzed in terms of accuracy, consistency, time cost, and handling different nerve situations.

In Chapter 4, the robotized needle insertion control approach is introduced. Here, the needle insertion kinematics are explained. Followed by proposing two techniques based on these kinematics, the robotized needle insertion using haptic force feedback, and the CoBotized needle insertion using a CoBotic system. Finally, robotic experiments and results are presented.

As a conclusion, a final chapter is given which discusses and summarizes the main contributions of this thesis and proposes some perspective work.

In Chapter 5, a complete summary of the thesis is presented in French.

Chapter 1

Context

Outline:
In this chapter, an introduction of this study is presented where we discuss the ultrasound-guided regional anesthesia procedure, difficulties, challenges, and limitations as performed by experts. As well as, analyzing existing robotic systems for medical surgeries. Finally, the proposed robotic-assisted UGRA system and the objectives of this thesis are explained.

Contents

This thesis was achieved in the Laboratoire Pluridisciplinaire de Recherche en Ingénierie des Systémes, Mécanique, Energétique (PRISME), a laboratory of the University of Orléans and the INSA-CVL. The thesis was financed by the DANIEAL2 project 'Détection et Analyse des Nerfs dans les Images Echographiques pour l'Anesthésie Locorégionale'. The University of Orléans, the University of Tours, the Médipôle Garonne hospital (Toulouse) and Adechotech company have developed a collaboration around the DANIEAL2 project with financial assistance from the Center Val-de-Loire region[1].

DANIEAL2 project aims to provide anesthetists with a Robotic-Assisted Medical system. This system provides a collaborative expert-robot-environment platform to significantly improve the routine of Ultrasound-Guided Regional Anesthesia (UGRA). DANIEAL2 addresses the study and development of an intrinsically safe robotic medical device, where the expert is still in the control loop. The support platform will be designed to: facilitate the provision of high quality and personalized medical care, improve the training process, reduce the risk of trauma, and quick patient recovery.

The field of Computer-Assisted Medical Surgery combines the domains of medicine, computer science, and robotics. The techniques in this area focus on the objectives of increasing the doctor's performance by enhancing their abilities. This assistance aims to reduce patient trauma and hospitalization time. UGRA operation has become a standard procedure in many hospitals [9]. Each year in France, hundreds of thousands of UGRA operations are performed. The motivation for this thesis is to develop an efficient and robust robotic-system for assisting the anesthetists in performing UGRA.

In this chapter, we justify both the motivation behind this thesis by introducing the UGRA procedure in Section 1.1. The limitation and challenges of the UGRA procedure are discussed in Section 1.2. Then, existing robotic systems for medical surgeries are explored in Section 1.3. Followed by explaining the objectives of this work and discussing its contributions to the field of anesthesia in Section 1.4.

1.1 Ultrasound-Guided Regional Anesthesia (UGRA)

The term "anesthesia" comes from Latin and means "without sensation". Anesthesia does not mean direct treatment, it allows the surgeons to treat, diagnose, or cure an ailment which would otherwise be painful or hinder patient's health and recovery time.

In the next section, we provide a thorough description of Regional anesthesia (RA) and Ultrasound-Guided Regional Anesthesia (UGRA).

1.1.1 Regional anesthesia

It wasn't until the late 1800s when surgeons and dentists alike began to use cocaine as a topical analgesic to perform surgical procedures painlessly on conscious patients. Prior to this development, painful procedures would require the use of general anesthetic agents, most often ether, developed in 1846 by a dentist by the name of W.T.G. Morton who first

---

[1] Grant number grant 2016-00108375

publicly demonstrated its effectiveness by performing a tooth extraction of a patient under its influence [10]. However, general anesthesia carries inherent risks to the patient and can prove to be impractical, particularly in an outpatient setting. Notable advancements over the following decades would include the addition of vasoconstrictor to local anesthetic solutions, the concept of nerve blockade, the development of the modern syringe, and the introduction of several synthetic analogs of cocaine, which would prove to be safer and more efficacious as local anesthetics [11].

Regional anesthesia (RA) is most often used when the procedure is confined to a specific region of the body, involves a large area of the body where injection of large amounts of an anesthetic might cause side effects that affect the entire body, and does not require general anesthesia. RA has saved lives and it calls for a cooperative patient and much patience on the side of the surgeon and anesthetist. For decades, active research in the operations field focuses on the possibility to introduce new methods that reduce risks and post-operative pain. RA has become an essential routine for pain management. It allows to remove the sensation of a part of the human body on which will be able to perform minor surgical procedures. The main reason of performing RA is to replace general anesthesia and provides an effective technique which offers faster operation settings, a significant reduction in the duration of hospitalization, and most important injecting the patient with a less quantity of anesthetic [12].

Regional anesthesia can be applied to either the central or peripheral nervous systems [13,14]. The central nervous system consists of the brain and spinal cord. The brain transmits and receives information to/from the human body via the spinal cord. The main function of the peripheral nervous system, such as the median nerve, is to circulate information between several organs and the central nervous system [14]. Figure 1.1 shows the peripheral nervous system where each nerve is linked to specific functionality. For example, the median nerve is a branch of the brachial plexus innervating various muscles in the anterior part of the forearm and part of the hand.

RA procedure consists in locating the nerve in order to inject a local anesthetic into its periphery. Traditionally, there are two methods for locating the nerve: one based on neuro-stimulation and one based on imaging guidance [15].

Regional anesthesia based on the neuro-stimulation technique consists in locating the nerve blocks by stimulating the sensitive fibers. This process is linked to the triggering of nerve impulses in a nerve using an electrical impulse. Just as this electrical impulse makes it possible to trigger a muscular reaction which corresponds to the territory of intervention. Once the nerve is located, a needle connected to a stimulator and a syringe is used to inject the anesthetic liquid [15, 16]. Neurostimulation is a very well-known technique in RA, this technique has been used for several years, and has been considered as the reference technique for RA [17, 18]. However, this technique requires precise knowledge of the anatomy of the human body in order to deposit the anesthetic in the vicinity of the nervous block only to avoid the intra-neural injection, source of the lesion, and disabling nerve sequelae. Another drawback is that neurostimulation is an invasive method that has the disadvantage of blind nerve research, the accuracy of which is linked to good anatomical knowledge. In addition, perfect knowledge of the anatomy is sometimes insufficient for locating precision, due to inter-patient morpho-

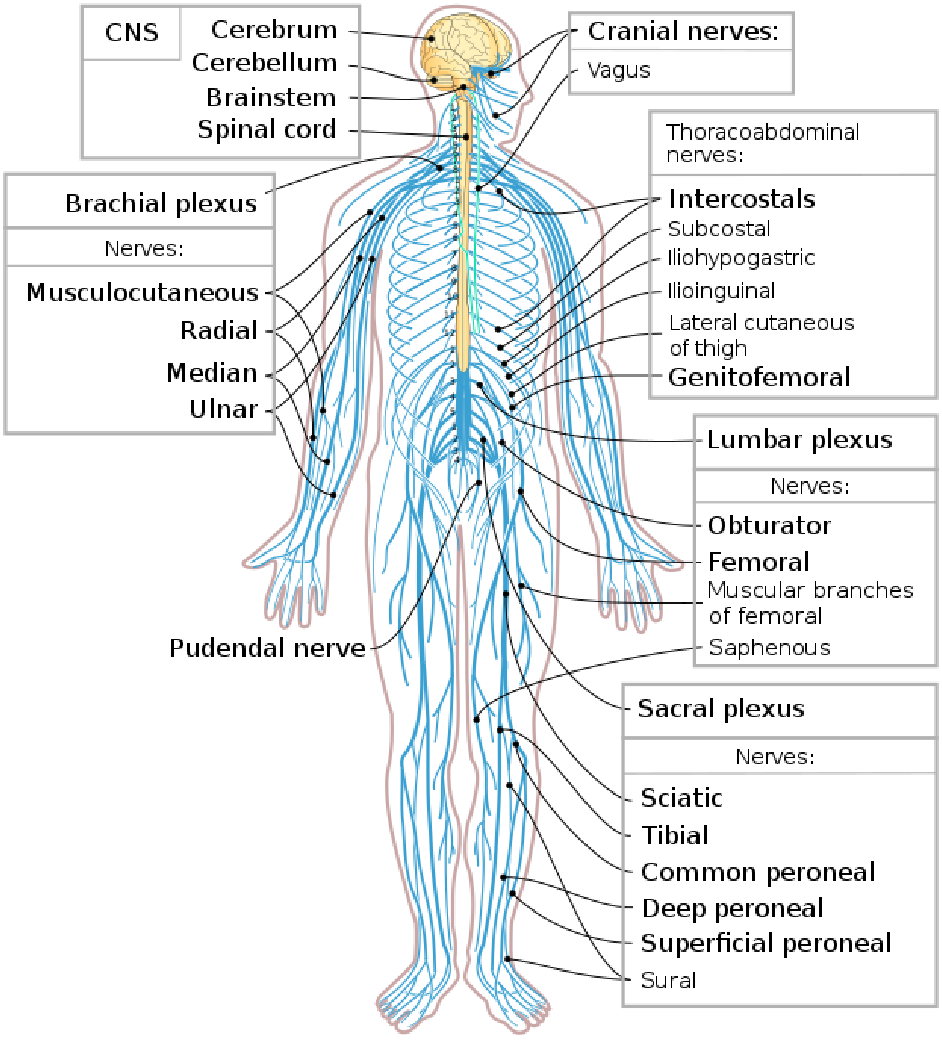


Figure 1.1: Nervous system diagram (Credit: Persian Poet Gal; wikimedia.org).

logical variations. For that RA requires high experience anesthetists to avoid many risks such as block failure, local anesthetic toxicity, nerve trauma, and neurological and vascular injuries [1].

On the other hand, during the last twenty years, several imaging techniques have been used to perform RA [19]. Magnetic Resonance (MR) imaging technique provides a clear image but due to the magnetic field, non-magnetic needles had to be used which makes the operation more expensive and long. Computed tomography (CT) and fluoroscopy techniques provide clear images but due to the x-ray radiation, the operation may harm the patient. The ultrasound (US) imaging is a low cost, radiation free, and real-time acquisitions technique which provides an additional value to RA. The combined technique is called Ultrasound-Guided regional anesthesia (UGRA).

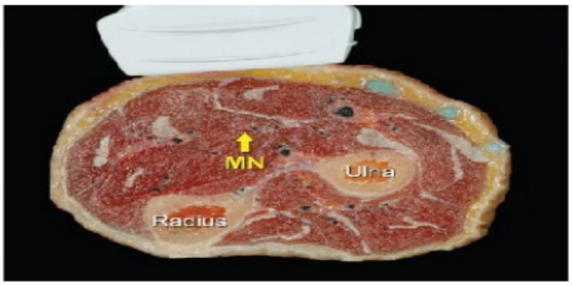
Many studies have shown advantages of using US imaging for RA. The results of these studies have shown in particular a reduction in neurological complications [20]. In [21], an analysis of the RA procedure carried out by anesthetists in training for 44 months showed 6 cases of complications in 9000 blocks based on neuro-stimulation alone, while for ultrasound-guided RA did not give any complications. Today recommendations have been issued by the health services of several countries such as the United States of America and the European Society which recommend the use of the US image for the procedure of RA in order to improve the success rate and reduce the number of complications [20].

1.1.2 The UGRA procedure

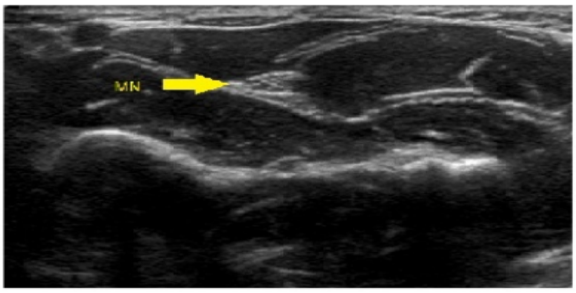
UGRA is becoming a powerful technique in surgical procedures and pain management. US allows a good anatomical structure visualization, which improves the success of the RA operation, and reduces the risk of the RA failure. Figure 1.2 shows the procedure of UGRA of the median nerve at the proximal region of the arm.

Technological developments in the field of US imaging have enabled the use of this non-invasive technique to locate the nerves, identify the needle, and follow the injection of the local anesthetic in real-time. The biggest advantage of this technique is the possibility of having visual feedback, allowing direct localization of the nerves and surrounding structures (i.e. blood vessels, tendons, etc.). US device gives, on one hand, the possibility of choosing the trajectory of the most appropriate needle in order to reduce the risk of vascular puncture and accidental intra-neural injection. On the other hand, this technique brings visual comfort to observe the diffusion of the local anesthetic throughout the RA procedure. Precise control is then possible over the quality injected, thereby considerably reducing the risks of peripheral neuropathy, nerve trauma, and toxicity [23].

In this section, the US device is explained, followed by detailing the UGRA procedure from the expert's point of view.



(b) Anatomical slice



(a) Positioning the US probe and the needle

(c) 2D US plane

Figure 1.2: UGRA of the median nerve [22].

1.1.2.1 The ultrasound device

The Ultrasound (US) device enables the practitioners to have an idea of the current pose of the target region and the needle inside the patient's tissues and to better understand the orientation of the US probe and where the needle should be steered to. The use of US guidance in the practice of RA requires high-performance US equipment and a thorough understanding of anatomical structures. Figure 1.3 shows an example of the practice of UGRA where RA anesthetist needs to acquire both a solid foundation in the field of US and acquire the practical skills necessary to visualize nerve structures [24].

The main function of the US device is to send ultrasonic waves and receive the reflective waves containing the tissues characteristics. These ultrasonic waves are produced by the piezoelectric effect. The piezoelectric crystals that constitute the heart of a US probe act both as signal generators and receivers. The probe is subjected to electrical charges, where the piezoelectric crystals transform the electrical wave into mechanical vibration, in order to generate an ultrasonic wave.

US waves considered a high-frequency sound waves generated in specific frequency ranges and sent through tissues. How sound waves penetrate a tissue depends on the range of the frequency produced. Lower frequencies penetrate deeper than high frequencies. The frequencies for clinical imaging (1-50 MHz) are well above the upper limit of normal human hearing (15-20 kHz). Wave motion transports energy and momentum from one point in space to another without transport of matter [25]. As the sound passes through tissues, it is absorbed, reflected, or allowed to pass through, depending on the echo density of the tissue [26]. Substances with high water content (e.g., blood, cerebrospinal fluid) conduct sound very well and reflect very

Figure 1.3: The UGRA Practice. (credit: clinicalgate.com).

poorly and thus are termed echolucent. Because they reflect very little of the wave, they appear as dark areas. Substances with low water content or high in materials that are poor sound waves conductors (e.g., air, bone) reflect almost all the sound and appear very bright. Substances with sound conduction properties between these extremes appear darker to lighter, depending on the amount of wave energy they reflect.

Each ultrasonic vibration is characterized by a specific frequency and wavelength. The speed of propagation depends on the tissues crossed, for example, the speed of air propagation is $330m/s$, that of bone $3000m/s$ and that of muscle $1600m/s$ [26, 27].

Viewing nerves by the US requires the use of high frequencies providing high-resolution images. Most nerve block location applications require frequencies in the range of 10-14 MHz [23] which make it possible to exploit surface structures.

The process of building the US image starts by emitting ultrasonic waves by the probe and then receives it. These received waves id transformed into an electrical signal under the piezoelectric effect. The electronics of the US system are responsible for amplifying and processing these signals in order to convert it into a digital signal. The transformation of the digital signal into a US image is based on several stages; the received signal is sampled in order to be stored in a matrix. At the end of this sampling, the content of the matrix includes coded values of gray levels obtained according to the intensity of the return echo. The quantification operation is generally based on a scale of 256 gray levels. An interpolation from the values of the closest echoes is carried out in the second step, finally, a spatial smoothing is carried out in order to remedy the interpolation effect.

The US image is affected by the intensity strength of the echo wave which provides a clear idea of the explored tissues. If the reflected wave is of strong intensity, we observe a phenomenon known as hyperechoic (obtaining of the strongly white region) like for example the bone. Conversely, if the reflected wave is of low amplitude, this time we observe a hypoechoic effect (very dark region) as is the case for certain peripheral nerves. Another particular case, which manifests itself when no wave is reflected and therefore no echo is detected giving rise to the phenomenon of an anechoic zone (totally black zone) such as arteries.

Despite the variation in the tissue intensities, the US image still considered as low-quality image. This image is often affected by various artifacts (Speckle noise, hidden anatomical structures due to shadows in the image, absorption, or attenuation of the wave when crossing soft tissues). Despite those limitations, the US is the standard intra-operative imaging modality for UGRA.

Several types of probes are used for the UGRA procedure, where linear transducers stand as the most frequent transducer used in this procedure. Linear transducers have a high scan line density which produces the resolution necessary for direct nerve imaging. Figure 1.4 shows different linear probes which are used in the UGRA procedure.

Figure 1.4: US transducers for regional anesthesia. The photograph includes (left to right) broad linear, small footprint linear, curved, sector, and hockey-stick transducers [25].

A good visualization of the nerve and the needle are a guarantee of the smooth running of an uncomplicated UGRA as shown in Figure 1.5. Unfortunately, the quality of the video is sometimes marred by the limitations linked to the settings of the US system as well as to the probe which generates a degradation of the images. Hence, It is important to optimize the probe settings and also to know the possible artifacts to avoid any complications during the procedure.

1.1.2.2 The medical UGRA procedure technique from the medical aspect

UGRA is a complex procedure involving advanced manipulative skills of the US probe, as well as the needle insertion. The UGRA procedure can be defined by the following steps:

Figure 1.5: US image of the median nerve.

- Searching for the nerve blocks guided by the US. This search should visualize all the anatomical structures in the image and locate the target area (the target nerve). Once the target nerve position is suitable, the position of the probe should remain relatively unchanged for the remainder of the procedure.

- The needle insertion procedure. It starts by finding the needle point of insertion (The interaction pivot point between the needle and the skin of the patient and named the insertion point or the trocar in medical term). The anesthetist starts maneuvering the needle around the insertion point until reaching the target nerve while avoiding risky regions such as arteries.

- Once the needle is located near the target nerve, the anesthetic is administered locally under direct US visualization until the nerve structures are completely surrounded by the anesthetic.

Here, these steps will be discussed in more detail.

1.1.2.2.a Nerve localization techniques

Localization of the nerve is a prerequisite for a successful RA procedure. The first step in the search for nerve blocks guided by the US is to be able to visualize all the anatomical structures to locate the target area (nerve). All the possible settings, namely the penetration depth, the frequencies, and the position of the probe, must be optimized to locate the target. As a general rule, it is possible to identify almost all peripheral nerves in the human body, but this requires the practitioner to have long training and regular practice of the US probe operative gesture [28,29].

Peripheral nerves have a fascicular or “honeycomb” echotexture. This consists of a mixture of a nerve fiber (hypoechoic) and connective tissue (hyperechoic) content within the nerve. Nerves that are surrounded by the hypoechoic muscle are usually easier to visualize than nerves that are surrounded by hyperechoic fat because the nerve borders are more abvious to identify. Nerves can be round, oval, or triangular depending on the size of the nerve, the frequency, and the angle of the wave [30,31]. Furthermore, the nerve shape can change along the nerve path or with strong probe compression [32].

There are two different techniques to hold the US probe, shot-axis and long-axis techniques as shown in Figure 1.6. The long-axis views are useful for demonstrating longitudinal distributions of local anesthetic along the nerve path in one image. The short-axis sliding (sliding the transducer along the known nerve path with the nerve viewed in short axis) is a powerful technique not only to identify small nerves with the US but also to assess the longitudinal distribution of local anesthetic along the nerve. The short-axis technique makes it possible to obtain a cross-sectional image in which the nerves appear in multiple oval hypoechoic zones surrounded by hyperechoic tissue. Most of the nerve blocks are detected using the short axis technique.

Despite the peculiarity of the nerve structure that allows it to be localized, its appearance can be confused with the structures that surround it such as blood vessels, and tendons [33]. For example, the appearance of the median nerve and tendon may be similar during the RA procedure as shown in Figure 1.7.

The difficulty of locating the nerve is not limited to the ambiguity between its appearance and other structures. Indeed, by manipulating the probe to optimize the quality of the image, the visualization of the nerve can disappear from the US image. There are two explanations for this phenomenon. On one hand, the peripheral nerves are very mobile structures that can change position. This phenomenon was well described by [34], they demonstrated the variable location of the median nerve in relation to the axillary artery, due to a slight pressure applied by the probes. On the other hand, the reflection and refraction of ultrasonic waves affect the nerve appearance. The only way to get an optimal view of the nerves in a US image is to have the waves reflected on a nerve. If the conditions for a high reflection rate are not met, the region of the nerve in the US image will either be degraded or completely disappeared.

Therefore, it is important for the practitioner to know how to orient the probe and to determine on which anatomical area it should be placed on the patient. In fact, in order to obtain optimal visualizations, the techniques for handling the probe are essential.

To optimize the US image, the mnemonic PART (pressure, alignment, rotation, tilting) has been recommended as shown in Figure 1.8.

- Pressure is necessary to minimize the distance to the target and compress underlying subcutaneous adipose tissues. It is a useful maneuver that has numerous effects. First, it moves the active face toward deeper structures to make them appear more superficial. It

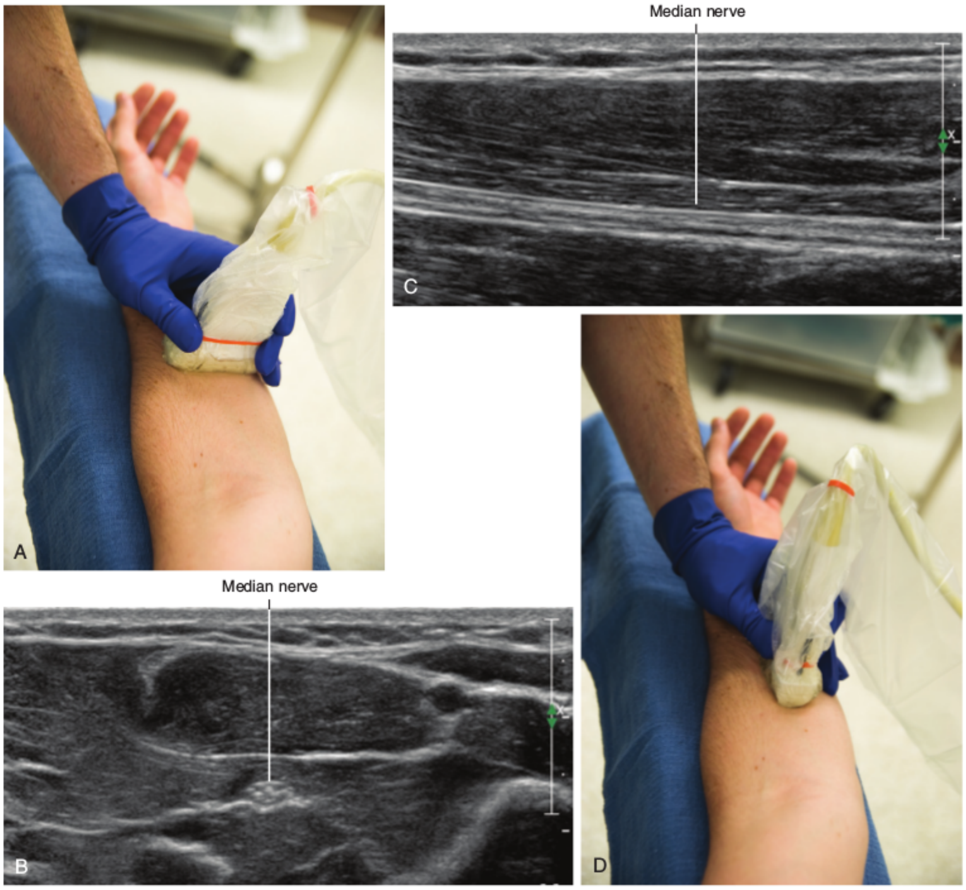


Figure 1.6: Median nerve viewed in short axis (A and B) and in long axis (C and D) [25].

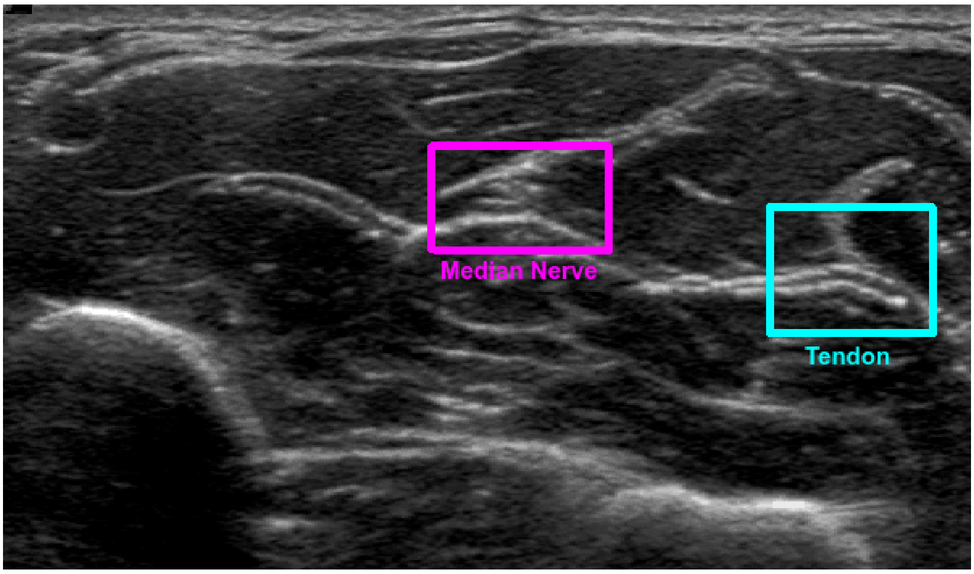


Figure 1.7: Median nerve and tendon US image.

also helps differentiate between different structures by the relative compressibility veins are easy to compress arteries are somewhat compressible and pulsatile and nerves are fairly incompressible.

- Alignment (sliding) refers to placing the transducer in a position over the extremity (or trunk) at which the underlying nerve is expected to be in the field of view.
- Rotation allows fine-tuning of the view of the target structure. As well as, it is used to switch between the short axis and long axis imaging.
- Tilting helps to bring the face of the probe into a perpendicular arrangement with the underlying target to maximize the number of returning echoes and thus provide the best image.

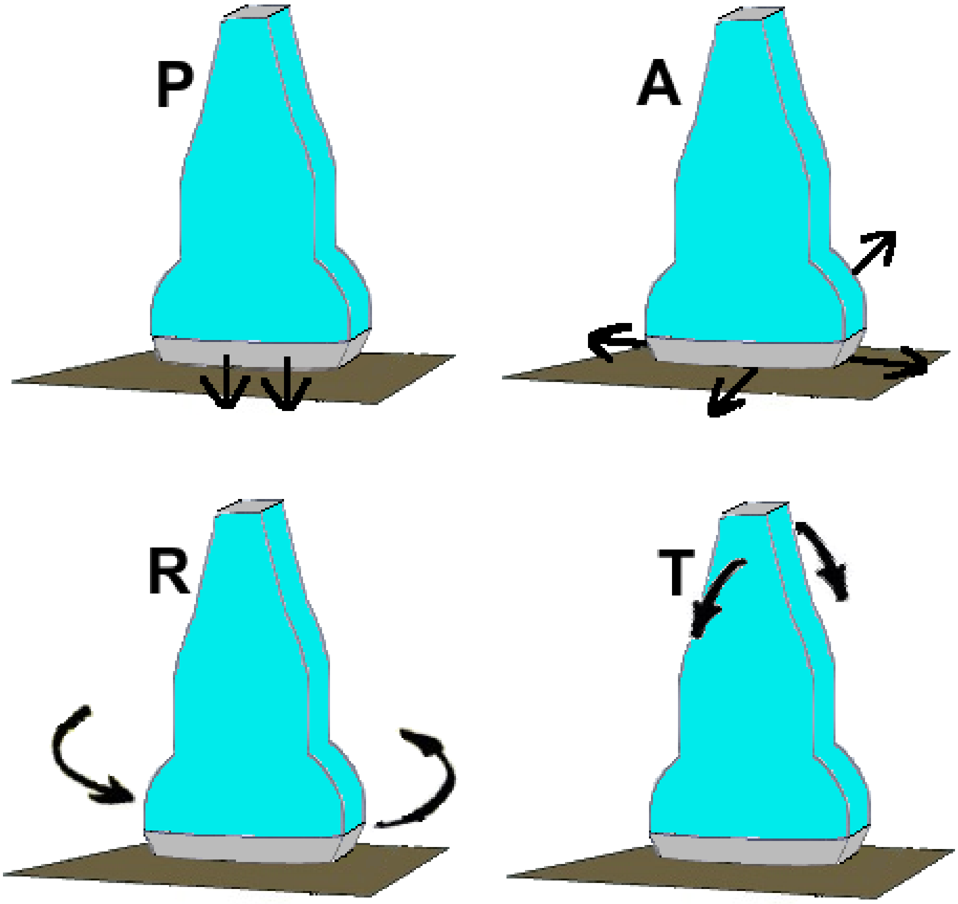


Figure 1.8: PART technique for probe handling(pressure, alignment, rotation, tilting).

In general, probe Alignment (sliding) is often the most regularly performed in order to locate the insertion of the syringe and locate the nerves [33]. Probe tilting can also improve image quality by optimizing the US contrast between the nerve and surrounding tissue. Probe rotation also proves to be effective for clearly visualizing the entire needle.

1.1.2.2.b The needle insertion

Once the target image is optimal, the position of the probe should remain relatively unchanged for the remainder of the procedure. In UGRA, a rigid needle is used for insertion, as there are two basic approaches for needle insertion under US guidance, out-of-plane and in-plane techniques [4]. With the out-of-plane technique, the needle tip crosses the plane of imaging as an echogenic dot. With the in-plane approach, the entire tip and shaft of the advancing needle are visible. Figure 1.9 shows a schematic drawing of the US out-of-plane and in-plane imaging.

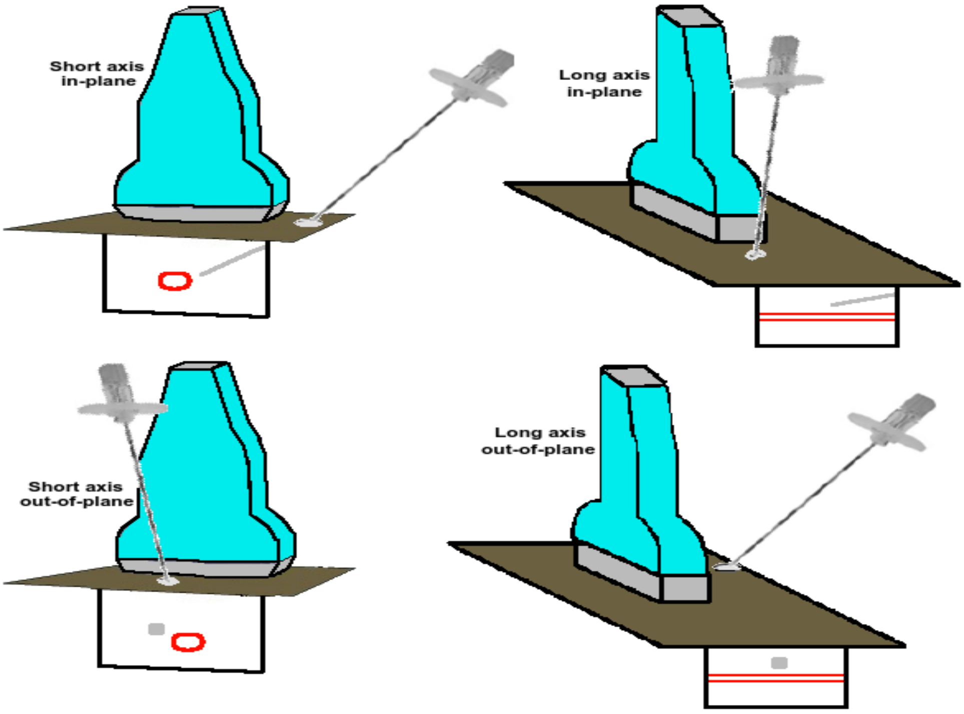


Figure 1.9: Schematic drawing of the short-axis (SAX) and long-axis (LAX) out-of-plane (OOP) imaging (top panels), and SAX and LAX in-plane (IP) imaging (bottom panels). The nerve shape is presented in red while the needle in gray.

There are several advantages to the out-of-plane approach to regional block. The out-of-plane approach uses a shorter needle path than in-plane approaches. One disadvantage of the out-of-plane approach is the extent of the unimaged needle path (structures that may lie short of or beyond the scan plane). If the needle tip crosses the scan plane without recognition, it can be advanced beyond the scan plane into undesired tissue. On the other hand, there are several advantages to use the in-plane approach. It provides the most direct visualization of the needle tip and shaft, and the injection. The needle tip is visualized before advancing. One disadvantage is the long needle path, which results in more tissue for the needle to cross. Partial line-ups (visualization of the needle shaft without visualization of the needle tip in the scan plane) create a false sense of security and therefore compromise the safety of the technique. External marks on the transducer can be used to guide needle placement for in-plane technique.

However, skill is probably more important than the approach alone. There will probably never be a good study comparing the two approaches (out-of-plane versus in-plane) because of strong institutional biases regarding how to perform regional blocks. However, the in-plane approach is most suitable for superficial nerve blocks like the brachial plexus or the femoral block, because the needle insertion angle is relative to the probe, which improves visualization and facilitates the identification of the needle.

One of the interesting characteristics of the procedures in RA is that they only cover small distances and must be performed slowly. However, there still some difficulties in controlling the needle movement while insertion in RA. The insertion point acts as a pivoting point causing the movements of the instrument to be reduced from 6 to 4 degrees of freedom (DOF), translational motion along its axis, and rotational motions about an instantaneous point on the tool as shown in Figure 1.10. It can be inferred easily that two DoFs of the system are lost. Consequently, the anesthetist hand movements about the insertion point are mirrored and scaled relative to the instrument tip. As a result, RA procedures have a long learning curve for the anesthetists because of longer procedure times, more difficult manipulation of instruments, tortuous ergonomics, and patient's anatomical discrepancies [35, 36].

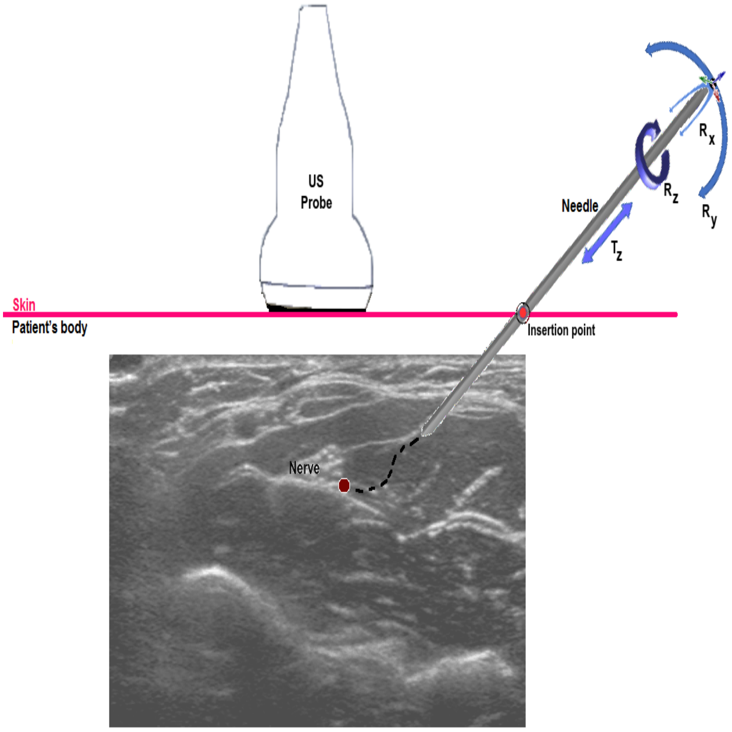


Figure 1.10: Needle insertion movements.

1.1.2.2.c Injection of local anesthetic based on US guidance

Once the needle is close to the nerve, the anesthetic is administered locally under direct US visualization until the nerve structures are completely surrounded by the anesthetic. If the anesthetic spreads in the wrong direction, the needle can be repositioned to ensure the proper anesthetic spread [37]. One of the most significant advantages of UGRA is the ability to identify in real-time the spread of local anesthetic around the peripheral nerves [38]. The possibility of seeing the distribution of the local anesthetic directly by the US minimizes the doses to be administered for the nerve blocks, which is particularly useful in multiple sequence procedures [39].

Local anesthetics work by penetrating tissue and blocking pain signals from being transmitted along nerve endings, preventing the pain signal from reaching the brain. The nerve endings have channels for the electrolyte sodium (Na channels) on them. When tissue is disturbed, the channels open and sodium enters the cell, changing the electrical charge. This electrical change becomes a pain sensation when interpreted by the brain. The local anesthetics are sodium channel blockers. In other words, they prevent sodium from entering the nerve endings, thus, preventing pain. The patient won't be able to feel any pain after having a local anesthetic, although he/she may still feel numbness and some pressure or movement. It normally takes a few minutes for the drug to take effect, and it wears off after a few hours. A stronger and higher dose will last longer.

When performing regional anesthesia, the practitioner must decide on the specific local anesthetic agent(s) as well as the volume, concentration, and mass to be injected. This is based on the desired outcomes of block onset, intensity, duration, and adverse effects [40]. As well as, the type and dose of anesthetic will depend on many factors. These include the patients' age, weight, any allergies, the part of the body to be operated on, and any current medical condition.

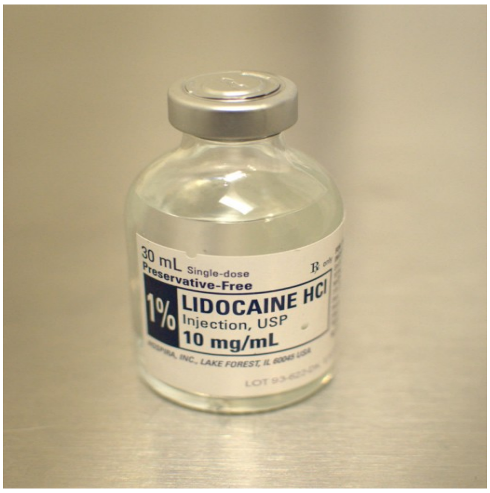


Figure 1.11: Lidocaine is a commonly used local anesthetic (Credit: JL Johnson; Flickr.com).

Various drugs are used to block the pain. The drug works by acting on certain nerve pathways to prevent the nerves in the area of application from sending signals to the brain. Cocaine was the first used anesthetic, but now it is rarely used. Lidocaine is now the most widely used local anesthetic, but different drugs are used for different purposes. For longer procedures, bupivacaine is more suitable, but it can be more painful when first administered. An anesthetist may, therefore, use lidocaine first, then inject with bupivacaine later, if numbness is needed for a longer period.

1.2 Limitations of UGRA

In RA, the anesthetic is injected close to a nerve, a bundle of nerves, or the spinal cord. Skill and experience are needed for the anesthesia specialist to inject the anesthetic at the proper location because the site of injection of the anesthetic has a significant impact on its effectiveness. RA risks and complications happen when the needle's tip touches the nerve which may cause nerve damage. RA also carries the risk of systemic toxicity if the anesthetic is absorbed through the bloodstream into the body. Other complications include heart or lung problems, and infection, swelling, or bruising (hematoma) at the injection site. For that RA requires high experience anesthetists to avoid many risks such as block failure, local anesthetic toxicity, nerve trauma, and neurological and vascular injuries [1].

On the other side, figure 1.12 shows the main difficulties of practicing the UGRA procedure where the anesthetist holds the US probe in one hand and the needle and the injector in the other. The anesthetist searches for the nerve location using the US probe until finding a suitable probe location. While preserving the nerve visualizing all the time in the same plan, the expert inserts the needle toward the nerve and starts maneuvering the needle's tip away from the unwanted risky regions such as arteries and tendons. During the insertion, the anesthetist maintains the needle tip visualization or movement in the image frame until reaching close to the nerve. Therefore, the anesthetist must perform complex hand-eye coordination to keep both the needle and the nerve visible in the 2D plane of the US image. Finally and while keeping the visualization of the nerve and the needle tip, and holding the needle tip position close to the nerve, the anesthetist injects the anesthetic to complete the procedure.

As discussed before and based on these difficulties, practitioners are required to undergo extensive US manipulating and needle control training for UGRA. In [41], the study has shown that learning takes a long time to master this technique. This is due to the difficulty of locating the nerve and the needle in the US image [42]. Another study in [43] aims to characterize the behavior of novices as they undertook the challenges of learning the UGRA technique. In this study, there were a total of 398 errors committed out of 520 operations, with the 2 most common errors consisting of the failure to visualize the needle before advancement and unintentional probe movement.

In [43], five quality-compromising patterns of behavior were identified: (1) failure to recognize the maldistribution of local anesthesia, (2) failure to recognize an intramuscular location of the needle tip before injection, (3) fatigue, (4) failure to correctly correlate the sidedness of the patient with the sidedness of the US image, and (5) poor choice of needle-insertion site and angle with respect to the probe preventing accurate needle visualization.

Overall, the UGRA procedure implies a high level of surgical skills and requires a long

Figure 1.12: The main difficulties of practicing the UGRA procedure.

learning process. Therefore, Robotic-assisted systems for UGRA is considered a great solution that can provide the anesthetists with better control in nerve detection and needle trajectory which leads to assist the overall procedure in the success rate for patient safety.

1.3 Robotic systems for medical surgeries

In the past few decades, the surgical practice has been revolutionized by the introduction of advanced instrumentation enabling a more flexible procedure such as the minimally invasive surgery (MIS). Similar to RA, the main advantage of MIS is attributed to a reduction in patient trauma, resulting in faster recovery and lower hospitalization costs. MIS offers greater challenges to surgeons. Due to the inherent kinematic constraints at the incision points where the motions of MIS instruments are restricted to four degrees of freedom. Despite the lack of dexterity and perception, all surgeries are moving toward MIS to give more benefits to patients at the expense of a more stressful environment to surgeons.

Several major problems can be identified during the MIS practice: the need of an “insertion point” for an instrument causes kinematic restrictions [44]; the use of screens to provides visual feedback leads to a loss of depth due to the 2D visual feedback [45]; maintaining the imaging system and instrument control as precisely as possible cause human tremors and/or fatigue due to the duration of the operation [46]; installation during surgery sometimes leads to poor ergonomics for surgeons due to the length of the instruments which leads to fatigue and sometimes tremors [47].

Along with the fine development of hands-on surgeries, surgical robots have been introduced to assist surgeons in performing various surgeries. For the past 20 years or so, robotic minimally invasive surgery has emerged to improve operating conditions in the operating room by assisting the surgeon with specific tasks.

In 1920, the term "robot" was first used by Czech writer Karel Čapek in his science fiction play “Rossum’s Universal Robots”. It introduced machines with a human appearance as artificial workers. In the 1980s, robots have become a reality in several fields of application such as industry, medical, etc. Robotic arms have been developed and used to replace or assist humans in tasks that are difficult, arduous, or risky for the operator. Significant efforts have been made to move from traditional robotics that are based on the idea of replacing humans, to collaborative robotics which are based on interaction with human.

Compared to the hands-on MIS surgeries, the robotically assisted surgery is a more precise operation that minimizes any potential damages which may be incurred from the negligence of the surgeon’s hands. A rigorous definition of “surgical robot” that “the surgical robot would be a powered computer-controlled manipulator with artificial sensing that can be reprogrammed to move and position tools to carry out a range of surgical tasks” [48]. Briefly speaking, the surgical robot is a robotic manipulator that is used to assist surgeons to perform surgical operations. Historically, the first surgical robot is introduced by Kwoh in 1985 which called PUMA 560 and aimed to hold a fixture next to the patient’s head to locate a biopsy tool for neurosurgery [49]. Since then, many surgeries began to attempt surgical robots as the assistive devices in operation rooms.

The key difficulty of using medical robots is that the surgeon has to deal with several constraints. These constraints are exclusively presented to which the MIS robot should adhere to guard safety. In an MIS, the surgical instrument, usually held by a robotic wrist, is moved with relatively large angular mobility about a single point or within a limited spatial volume. This constraint is called “fulcrum effect” [50, 51]. In UGRA, for example, the instrument pivots at the point at which they enter the patient’s body. Such constraint enforces the surgical tool being manipulated with four degrees-of-freedom (DOF), including three rotational DOFs pivoted at a point and one translational DOF whose axis moves through this point. This design consideration has inspired researchers to design robots to articulate a mechanism that can mechanically decouple rotational and translational motions of tools at the insertion point. This mechanism is called the “remote center-of-motion (RCM)” mechanism [52]. Thus, the RCM mechanism plays a principal role in the kinematic design of MIS robots.

In the context of surgical robotics, and whatever approach is chosen (remote handling or co-handling), many robotic systems focus on the response to the RCM problem to avoid damaging the tissue around the insertion point and to reduce patient pain in the event of RA [53–55]. Different robots have been used for assisting the MIS practice, where the RCM mechanically constrained, controlled by free-wrist robots, or command constrained.

For mechanically constrained robots, the RCM is defined and mechanically imposed by the kinematic structure of the robot, such as [56–58]. For robots with a free wrist, it is possible to design a robot with its last two (or three) joints such as the AESOP system [59]. In this case, the RCM is tied to the trocar/insertion area.

(a) AESOP

(c) Zeus

(b) Probot

(d) Da vinci

Figure 1.13: Example of robotic systems for medical surgeries.

Constrained by command robots allow a precise control of the surgical instrument. However, in order to be able to apply these robots to minimally invasive surgery, it is necessary to calculate a virtual rotation point that constrains the robot control. Robots constrained by control were built by designing a system specific for MIS or use an existing robotic arm for the same purpose. Some notable and well-known systems are proposed in the last few decades such as Zeus [60] and da vinci [61]. These robots are the most widely accepted and sold robotic systems that applied to the medical field. Thanks to the full master-slave mechanism, the surgeon can operate remotely from a “master” console that is physically separate from the patient. On the other side, robotic arms were designed with the primary objective of imitating human gestures and achieving their performance, such as Franka Emika [62], Kuka [63] or universal robot [64]. Many researches were developed using a robotic arm for an MIS application. Thus following the rapid development in the past years, the robotically assisted MIS has been widely accepted by worldwide surgeons and patients nowadays.

To the best of our knowledge, this study proposed the first complete robotic-assisted system for the UGRA. In the next section, the robotic system is presented in detail.

1.4 The proposed robotic-assisted UGRA system

UGRA offers a huge performance impact on the practice of regional anesthesia [3,4]. For several years, there has been a growing demand for this technique in France and in other European

(a) KUKA

(c) Franka

(b) P-Rob

(d) Universal Robot

Figure 1.14: Commonly used robotic arms in research projects.

countries [65]. As discussed in the Section 1.2, there are several limitations the anesthetists may face during the UGRA procedure. Despite the long training that the anesthetists take, the procedure can take the benefits from robotics and image processing researches that may give assistance for the anesthetists during the procedure.

Robotic assistance are already used in some medical settings and show their huge impacts [7]. Robotic-assisted UGRA is a solution that can provide the anesthetists with better control which leads to reduce the UGRA limitations. A ‘CoBot’ (from collaborative and robot) is a robot intended to physically interact with the human in a shared workspace [66]. This integration strategy leverages the ‘strength’ and ‘endurance’ of robots with the flexibility and decision making of human participants. Therefore, the goal of this research is to provide the anesthetist’s act with a robotic system that is CoBotically manipulated to facilitates performing the UGRA procedure. It must be emphasized that this system is designed to be a complementary working tool for human operators. For the intented goal, the proposed system does not replace the anesthetist but assists him/her to perform UGRA. Furthermore, this robotic system can provide great assistance by helping the experts with techniques and tools that improve the procedure accuracy and safety such as avoiding nerve trauma or damage to healthy tissues. Moreover, it could increase the anesthetist’s experience by more teaching and learning to avoid unintentional injuries.

Forthat, the objective of this thesis is to propose a complete robotic system for the UGRA procedure. This system is divided into two connected robotic systems, a US probe holder robot (PHR) and a needle holder robot (NHR). As shown in Figure 1.15, the PHR aims to be CoBotically manipulated by the anesthetist to place the US probe on the patient’s skin. PHR must perform the following consequential tasks: co-manipulation of the US probe; nerve visual servoing; follow the orientation of the needle. The NHR must also be co-manipulated by the anesthetist while using his/her expertise gestures. It will assist him/her to position the needle on the insertion pointon the patient's skin, and to keep a safe needle motion inside the patient's body.

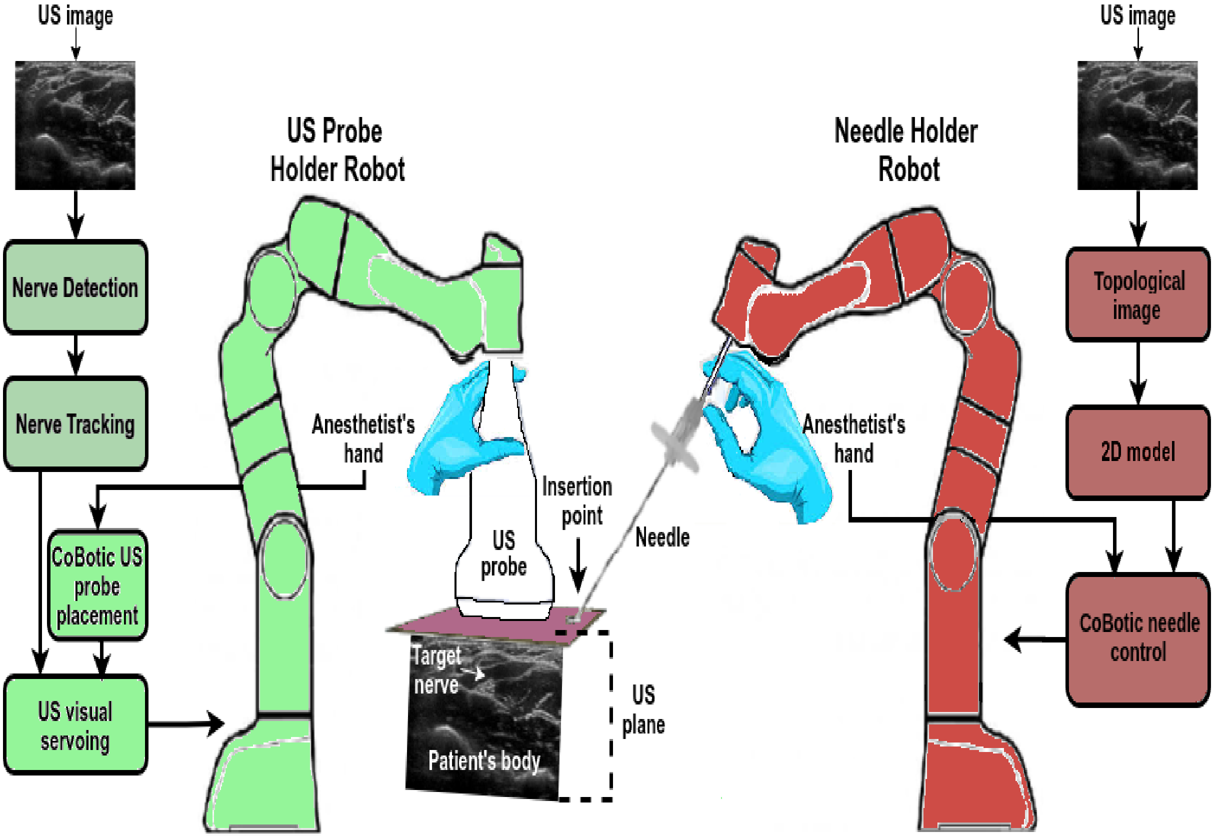


Figure 1.15: The proposed robotic-assisted UGRA procedure.

In this thesis, we focused on some aspects (as highlighted in Figure 1.15) for which we propose novel techniques and methods. For the other aspects, such as visual servoing, it will be a part of our future work. Overall, the work conducted during this thesis leads to three main contributions:

- The first contribution is to propose a tool that consists of processing information (feature) from a US image in order to automatically detect and track the nerve blocks. Hence, to facilitate the detection and tracking process, the target object should be differentiable from other objects using image features. Nevertheless, choosing and extracting relevant features is a complex task due to the nerve texture appearance and the noisy nature of this type of image. In Chapter 2, we address these criteria by introducing a novel texture descriptor, Robust Adaptive Median Binary Pattern (RAMBP). This descriptor is evaluated for high noisy textures because choosing the suitable robust feature is a crucial step in detection and tracking applications. This contribution is presented in Chapter 2.

- The second contribution is to propose various techniques for nerve detection and tracking in the US image. These techniques are based on: texture descriptors (RAMBP); CNNs models; and a new novel merged CNNs model (RAMBP with CNNs model). The results of nerve detection and tracking will be used as an input for visual servoing. Hence, this would facilitate the anesthetist's daily routine and allow him/her to focus on needle insertion and the anesthetic delivery. This contribution is introduced in Chapter 3.

- The third contribution proposes a robotic-control system for needle insertion. The robotic control strategy leverages the 'strength' and 'endurance' of robots with the flexibility and decision making of the anesthetist to facilitates performing the UGRA procedure. This system provides a safe and flexible solution for the robot and the medical staff in a shared workspace. Moreover, this system allows the experts to control the NHR for a better accuracy of the needle insertion. This contribution is presented in Chapter 4.

Chapter 2

Robust Features Extraction

Outline:
In this chapter, the ultrasound image is analyzed by highlighting the importance of finding robust and strong features in the image. These features will be used as the basis for detection and tracking.

Contents

2.1 Introduction

Ultrasound (US) imaging has been used in the last few decades and has an excellent safety record. It is based on non-ionizing radiation, so it does not have the same risks as other types of imaging systems that use ionizing radiation. Although the US image is generally considered a good clear image, it suffers from different noises and artifacts [67]. The US image is affected by the intensity strength of the echo wave which provides a kind of structural information of the explored tissues. If the reflected wave is of strong intensity, we obtain a bright regions called a hyperechoic regions. while, if the reflected wave is of low amplitude, we obtain a dark regions called a hypoechoic regions. The peripheral nerves, such as the median nerve, are considered as hypoechoic regions (bee comb-like structure) that is surrounded by hyperechoic regions. Furthermore, US images are acquired by US scatter echo signals which are correlated in the local medium and came from body tissues, structures, tissue microstructures, etc. But some parts of the scatters come from multiplicative noises caused by scatters of locally correlated areas which wavelength is smaller than US beams wavelength these noises called speckles. Another cause of speckle-noise is when using coherent source and non-coherent detector in the US [68–70].

There exist different studies on filtering the speckle noise [71]. Nevertheless, prior image enhancements would remove important information in the US image, such as the speckle information. This information is important for the anesthetists for locating the needle and recognizing different regions in the US image. Therefore, robust nerve detection and tracking should be made directly on the original image without any prior image enhancements such as removing the hypoechoic region or by filtering the image. For that, accomplishing detection or tracking tasks is very challenging in US images.

Detection and tracking problems in the image and signal analysis require to take into account complex information embedded in the data. Images might contain many thousands of pixel values that represent different objects. Even though humans can deal with both physical objects and abstract notions in day-to-day activities while making decisions in various situations, it is not possible for the computer to handle them directly.

A feature should contain the information required to distinguish between classes, be insensitive to irrelevant variability in the input [72]. As well as, the features should be limited in number, to permit efficient computation of discriminant functions and to limit the amount of training data required. Image features, such as edges and interest points, provide rich information on the image content. It corresponds to local regions in the image and it is fundamental in many applications in image analysis, such as recognition, matching, reconstruction, etc [72]. A feature descriptor is an algorithm that takes an image and outputs descriptors/feature vectors. Feature descriptors encode interesting information into a series of numbers and act as a sort of numerical “fingerprint” that can be used to differentiate one feature from another. Feature descriptors help to improve the efficiency of many tasks such as detection and tracking. For that, choosing a suitable feature is a crucial step in detection and tracking applications.

Texture feature is an important cue for detection in many US images applications [73] since some tissues show specific noisy texture properties, such as certain nerves. In this chapter, we propose a novel and robust texture descriptor "Robust Adaptive Median Binary Patterns" (RAMBP). As the US image considered a noisy image with texture regions, we evaluate the proposed descriptor on well-known noisy textures for the problems of texture classification.

The structure of this chapter is as follows. Section 2.2 introduces the state of the art for texture descriptors. Followed by detailing the proposed RAMBP descriptor in Section 2.3. A performance evaluation of the RAMBP descriptor is presented in Section 2.4. The chapter ends with conclusions in Section 2.5.

2.2 State of the art

Texture classification is one of the major problems in image analysis and a well-known research topic for its significant rule to understand the texture recognition process. Given training samples, texture classification categorizes an input image to one of a set of known textures. Texture classification has a crucial value in the fields of computer vision and pattern recognition, including medical imaging [74], document analysis [75], environment modeling [76], and object recognition [77].

Texture classification is divided into two major problems, feature extraction and classification [78]. Most researches target the feature extraction problem, due to the fact that having a strong texture features are more crucial and critical than having a strong classifier. The long-standing need for efficient and effective data classification indicates the important role of powerful and appropriate features. For any durable texture classification application, the feature extraction problem depends on two important points, descriptor quality, and time complexity [78, 79]. The high-quality descriptor must be distinctive for different texture classes and should reach the highest robustness for different texture variations such as rotation, scale, blur, and different kinds of noise.

One of the texture classification methods that gained huge attention has been extensively studied is the Local Binary Patterns (LBP). LBP is a simple yet powerful operator to describe local image patterns and shows robustness to illumination, rotation, and scale [80]. within a specific local area, LBP encodes the comparisons of neighboring pixels. And due to its low computational complexity, LBP has been used widely as a solution for many problems, such as texture classification [81], object detection [82], image matching [83], image retrieval [84], biomedical image analysis [85], face recognition [86], etc. For general texture classification purposes, LBP derivatives have been introduced with surveys given in [79, 87–91]. These derivatives fuse LBP with other visual cues to improve LBP robustness, discriminativeness and applicability such as ILBP [92], CLBP [93], RLBP [94], DLBP [81], etc.

However, despite the increase in discriminativeness, LBP derivatives have their weaknesses in terms of feature dimensionality and robustness to noise such as Gaussian noise, Gaussian blur, and impulse noise. These weaknesses have led to introduce several studies that aimed to present a noise-robust operator. For example, in [95], the authors proposed Median Binary Patterns (MBP) to add more sensitivity to microstructure and impulse noise robustness. Nevertheless, MBP does not handle other types of noise and showed low performance for the high levels of impulse noise. In [96], the authors introduced Binary Rotation Invariant and Noise Tolerant (BRINT). Although BRINT samples the points in a scaled circular neighborhood which made it more distinctive and robust to noise, it suffers from limitations in terms of robustness under high noisy textures. In [97], Schaefer et al. proposed Multi-Dimensional Local Binary Patterns (MDLBP), which added more information from different radii and concatenated it in one histogram. This makes the histogram more effective and robust to noise, but, on the other side, it increases features dimensionality which leads to high computational complexity. Furthermore, this approach suffers in robustness under high noise corruption. In [98], the authors introduced Adaptive Median Binary Patterns (AMBP). AMBP used a self-adaptive analysis window size depending on the local microstructure of the texture which made the descriptor more robust to impulse noise. Despite the noise robustness of this approach, it has limitations for textures with a very high level of noise.

To overcome the problem coming from high noisy texture, few studies had been introduced. For instance, Scale Selective Local Binary Patterns (SSLBP) and Median Robust Extended Local Binary Patterns (MRELBP). Guo et al. [99] proposed SSLBP which uses a Gaussian filter to produce a scale space of a texture image. For each image in the scale space, the pre-learned dominant binary patterns histogram is built. Then the scale-invariant feature for each pattern is found by taking the maximal frequency among different scales. SSLBP considered an effective descriptor for textures under Gaussian noise. Nevertheless, SSLBP filtering procedure failed under impulse noise. On the other hand, Liu et al. [100] performed median filtering and compared it with a sampling scheme to introduce MRELBP. MRELBP adds more microstructure and macrostructure information, but it uses a simple median filter procedure which leads to failure under Gaussian noise and extremely high level of impulse noise.

On the other hand, deep learnings have shown outstanding performances in texture classification where it learns features from the data of interest such as [101, 102]. In [103], Cimpoi et al. used filter banks and Convolutional Neural Networks (CNNs) for texture recognition and segmentation. The descriptor built on a pre-trained VGG-VD (very deep) model which improves the performance. This approach is an effective texture descriptor and produces more robustness for variable images recognitions, but it has weaknesses to the median and high noisy textures, and some shortcomings in term of time complexity.

Although the binary patterns family has huge success in the computer vision field, there are several weaknesses with these methods. In [91, 104], the authors performed extensive comparisons for the existing local binary features for texture classification, where many of the existing local binary approaches suffer from a serious limitation. These limitations can be concluded in the descriptor ability to handle textures with a high level of noise and to handle different types of noise which remains unsatisfactory.

Based on these limitations, the obvious question being raised here is how to reach high noise robustness without any prior knowledge of the noise type, without any prior learning process, and for different kinds of noise such as Gaussian noise, Gaussian blur, and impulse noise. In other words, performing the descriptor in noise-free data then try to classify the noisy and noise-free textures under different geometric and illumination condition.

Gaussian noise, Gaussian blur, and impulse noise are considered as the most frequent and challenging noises in image processing, computer vision, and pattern recognition fields. For that, to improve and ensure the best performance of the image processes such as classification, these noises should be detected, reduced, or removed. Some descriptors incorporate filtering procedure to improve the performance, such as Gaussian and median filtering. Many techniques have been developed to suppress Gaussian noise, such as mean filter, wavelet denoising [105], and kernel regression [106]. Nevertheless, these filters are suitable for Gaussian noise but not for other noises such as impulse noise. On the other hand, various filters have been proposed to remove impulse noise, such as median filter [107] and adapted median filter [108]. Hence median and adaptive median filters consider all pixels as noisy corrupted pixels, the filter will fail under images with a high level of noise. To avoid this drawback, the switching techniques were introduced such as Boundary Discriminative Noise Detection (BDND), which takes the advantages of detecting which pixel is corrupted and which one is not [109–111]. In this context, using pixel classification from switching techniques with binary pattern methods can lead to better texture analysis for different types of noise.

In the following section, an efficient and simple local binary descriptor (RAMBP) is proposed. It takes the advantages of switching techniques and median adaptive scheme to include more robustness in features for texture with a high level of noise. RAMBP captures both microstructure and macrostructure texture information and provides a better representation of the local structures. RAMBP effectiveness and robustness will be examined for high noisy textures classification.

2.3 Robust Adaptive Median Binary Patterns (RAMBP)

To provide an efficient texture descriptor, it should be discriminative and robust to noise. All state-of-the-art descriptors share one or more weaknesses of sensitivity for high noisy textures. RAMBP uses noisy pixel classification, an adaptive window for the threshold and binary modules, and regional values instead of using pixel intensities. Here in this section, a background on binary patterns is presented. Followed by explaining all RAMBP descriptor steps.

2.3.1 Background

RAMBP is derived from the binary patterns familly. To facilitate understanding the RAMBP principle in the next section, this section provides a backgound on the binary patterns derivatives that leads to RAMBP.

2.3.1.1 Local Binary Patterns

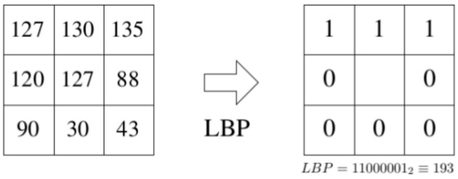
The Local Binary Patterns (LBP) [80] is one of the powerful operators for describing image texture features. The simplest, yet very efficient, LBP feature vector can be created by two steps. First, compare P neighbors of a central pixel to get the corresponding P-bit binary code with regard to the center pixel's gray value as a threshold. Second, the image LBP can be defined as

$$LBP_P(x_c, y_c) = \sum_{p=0}^{P-1} s(i_p, i_c) 2^p \tag{2.1}$$

where i_c is the central pixel value, i_p is the pixel value of the p neighbor, and $s(i_p, i_c)$ can be expressed as

$$s(i_p, i_c) = \begin{cases} 1 & i_p \geq i_c \\ 0 & i_p < i_c \end{cases} \tag{2.2}$$

Figure 2.1 shows a simple example of applying an LBP operator on a 3 × 3 neighborhood. Where the output central pixel after applying LBP is 193.



127	130	135
120	127	88
90	30	43

1	1	1
0		0
0	0	0

$LBP = 11000001_2 \equiv 193$

Figure 2.1: LBP operator applied to a 3 × 3 neighborhood.

2.3.1.2 Median Binary Patterns

Median Binary Patterns (MBP) demonstrates very good discrimination properties and used to provide more sensitivity to microstructure and noise robustness [95]. MBP operates maps by comparing P neighbors and the central pixel to get the corresponding P-bit binary code with regard to their median gray value as a threshold. MBP operates the same as LBP, but it takes the median value of the selected window as a threshold instead of the value of the central pixel. The indicator of Eq. 2.2 replaced by

$$s(i_p, i_c) = \begin{cases} 1 & i_p \geq Med \\ 0 & i_p < Med \end{cases} \tag{2.3}$$

where Med is the median value of the 3 × 3 neighborhood.

Figure 2.2 shows a simple example of applying an MBP operator on a 3 × 3 neighborhood. Where the output central pixel after applying MBP is 225.

2.3.1.3 Adaptive Median Binary Patterns

Adaptive Median Binary Patterns (AMBP) is an adaptive approach that incorporates a filtering process to obtain a robust representation of the local context [112] and considers image local

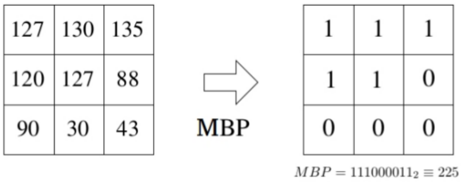


Figure 2.2: MBP operator applied to a 3×3 neighborhood.

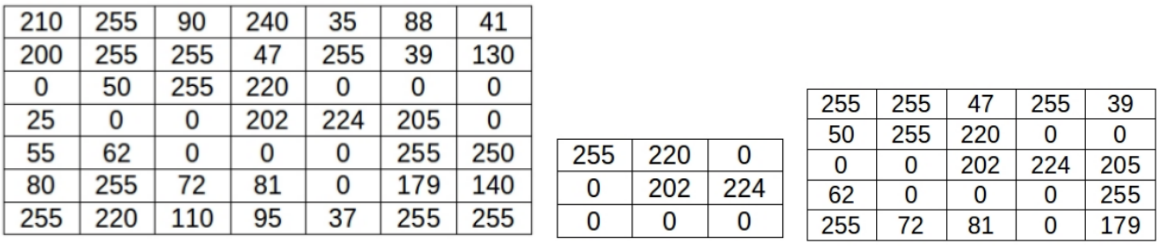
variations as additional information. AMBP uses a median adaptive filtering procedure which made it more robust to noise. AMBP descriptor combines both LBP and MBP depending on the noise and the microstructure information.

AMBP procedure is described in Algorithm 1. It starts by taking 3×3 neighborhood window, then computes the median, minimum, and maximum values. If the median value is greater than the minimum value and smaller than the maximum value, then the threshold will depend only on this window, else the window will expand by one pixel in all directions to be 5x5, and so on. After finding the acceptable threshold window, if the central pixel is greater than the minimum value and smaller than the maximum value, then the threshold will equal the central pixel, else the threshold will equal the median value. After obtaining the threshold, AMBP computes the binary value of the central pixel within the 3×3 window.

Algorithm 1: AMBP algorithm

```
Input: For each pixel(I) and maximum window size $w_{max}$.
Output: AMBP value for the central pixel.
for $w < w_{max}$ do
    Extract $w \times w$ window around I.
    Find the median $Z_{med}$, minimum $Z_{min}$ and maximum $Z_{max}$ values in the $w \times w$
      window.
    if $Z_{min} < Z_{med} < Z_{max}$ then
        break;
    else
        $w = w + 1$
    end
end
if $Z_{min} < I < Z_{max}$ then
    $\tau = I$ (LBP);
else
    $\tau = Z_{med}$ (MBP);
end
AMBP(I) obtained by using $\tau$ as threshold in the neighborhood $\Omega$.
```

Figure 2.3a shows an example of a hypothetical image. AMBP starts by taking the 3×3 window around the central pixel, as can be seen in Figure 2.3b. The pixel in are sorted in the ascending order, the median value equal to 0, the minimum value equal to 0, and the maximum value is equal to 255. as can be seen that the median value is not greater than the minimum value, which means that this window is not acceptable and needs to expand by one pixel in all directions. Figure 2.3c shows the expanded window 5 × 5. AMBP computes the median value of the new window, which equal to 72, the minimum and maximum values are 0 and 255 respectively. In this window, the median value is greater than the minimum value and smaller than the maximum value, which means this window is accepted to find the threshold in it. To compute the threshold, AMBP checks, in the 5 × 5 window, if the central pixel value is greater than the minimum value and smaller than the maximum value, the central pixel equal to 202 which is greater than the minimum value 0 and smaller than the maximum value 255. So, the threshold of this window will be set as 202. The threshold of AMBP binary value is applied to the 3×3 window and the binary pattern is equal to 010000101 which equal 67 in decimal.



210	255	90	240	35	88	41
200	255	255	47	255	39	130
0	50	255	220	0	0	0
25	0	0	202	224	205	0
55	62	0	0	0	255	250
80	255	72	81	0	179	140
255	220	110	95	37	255	255

(a) 7 × 7

255	220	0
0	202	224
0	0	0

(b) 3 × 3

255	255	47	255	39
50	255	220	0	0
0	0	202	224	205
62	0	0	0	255
255	72	81	0	179

(c) 5 × 5

Figure 2.3: hypothetical image.

2.3.2 RAMBP overview

Figure 2.4 shows the scheme of the proposed RAMBP descriptor, where it can be seen it's divided into three stages, the classification process of noisy pixels detection, threshold process, and generating the binary pattern. Here, these stages are presented in details.

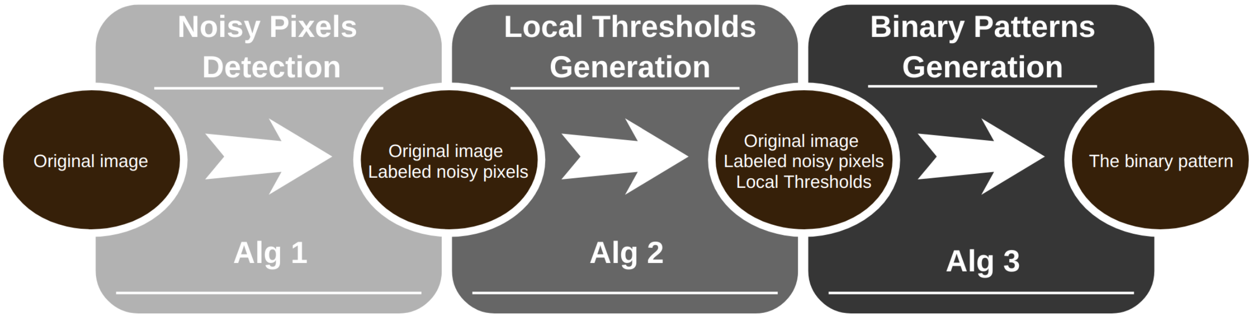


Figure 2.4: Illustration for the RAMBP descriptor.

2.3.2.1 Classification process for noisy pixels detection

As this section aims to perform texture classification without any prior noise knowledge, the first step consists of classifying each pixel in the image as corrupted or uncorrupted pixels. for that, the detection step of BDND algorithm [109] has been adopted.

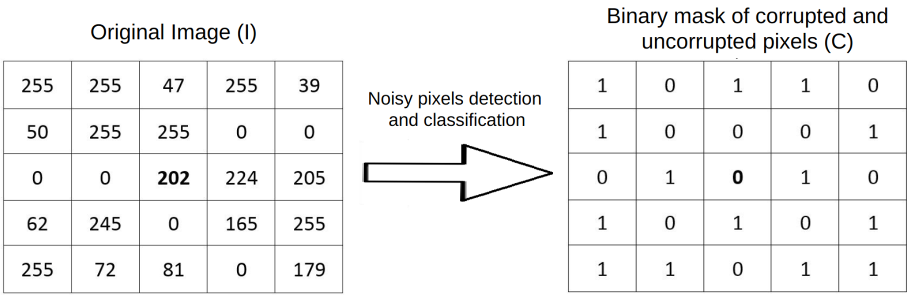
Pixel classification starts by taking a 21×21 window around the central pixel, then examine the pixel whether it meets the condition as an uncorrupted pixel. If the pixel considered as a corrupted pixel in the first stage, another examination will be invoked by imposing a 3×3 window around the central pixel to ensure the examination for more confined local statistics. A pixel classified as a corrupted pixel, if it fails in both examinations. Algorithm 2 provides a full explanation about pixel classification step.

Algorithm 2: Noisy pixels detection

```
   Input: The original image $I$.
   Output: The image of labeled pixels $C$.
 1 for each pixel position $x_0$ do
 2     For the current pixel $I(x_0)$, impose a $21 \times 21$ window.
 3     Compute $V_0$ by sorting the pixels in the window.
 4     Find the median value($med$) of $V_0$.
 5     From $V_0$, obtain the difference vector $D_V$. $D_V[i] = V_0[i+1] - V_0[i]$, where the index
         $i = 1 : length(V_0) - 1$
 6     Compute the left cluster range $v_L$, where $v_L = V_0[i_l]$ ($i_l$ is the index of
         $max(D_V\{0, i_{med}\})$).
 7     Compute the right cluster range $v_R$, where $v_R = V_0[i_r]$ ($i_r$ is the index of
         $max(D_V\{i_{med}, end\})$).
 8     Initialize three clusters of $V_0$, $\{0, v_L\}$, $\{v_L, v_R\}$, and $\{v_R, 255\}$.
 9     if $I(x_0) \in \{v_L, v_R\}$ then
10         $x_0$ labeled as uncorrupted pixel.
11     else
12         Repeat $2 - 8$ steps with $3 \times 3$ window around $x_0$. If $I(x_0) \notin \{v_L, v_R\}$, label $x_0$ as
             corrupted pixel. Otherwise, $x_0$ labeled as uncorrupted pixel.
13     end
14 end
```

Figure 2.5 provides an example of 5×5 window instead of 21×21 to facilitate understanding pixel classification algorithm (Algorithm 2) using the following procedure,

- The first step after choosing the window is to sort all the pixels in the window to obtain V_0, in the given example (Figure 2.5) $V_0 =$ [0 0 0 0 0 0 39 47 50 62 72 81 **165** 179 202 205 224 245 255 255 255 255 255 255 255], and the median value (med) of V_0 is **165**.
- Then the difference vector is obtained $D_V =$ [0 0 0 0 0 39 8 3 12 10 9 **84** 14 **23** 3 19 21 10 0 0 0 0 0 0].
- Find v_L, the correspondence pixel in V_0 that gives the maximum intensity differences in D_V left interval. (left interval of D_V is located between the 0 and med in V_0). In this example $v_L =$ **81**.



Original Image (I)

255	255	47	255	39
50	255	255	0	0
0	0	**202**	224	205
62	245	0	165	255
255	72	81	0	179

Binary mask of corrupted and uncorrupted pixels (C)

1	0	1	1	0
1	0	0	0	1
0	1	**0**	1	0
1	0	1	0	1
1	1	0	1	1

Figure 2.5: Pixel classification example of 5 × 5 window using the procedure of Algorithm 2. In the image C, corrupted pixels are represented as 0 and uncorrupted pixels as 1.

- In the same manner, find v_R, the correspondence pixel in V_0 that gives the maximum intensity differences in D_V right interval. (right interval of D_V is located between the *med* and 255 in V_0). And in this example $v_R = \mathbf{179}$.
- Then the three clusters are $\{0,0,0,0,0,0,39,47,50 \ \ ,62,72,81\}$, $\{165,179\}$, and $\{202,205,224,245,255,\ 255,255,255,255,255,255,255\}$. The central pixel $I(x_0) = \mathbf{202}$ belongs to the third cluster which considered as corrupted pixel and the pixel needs to re-examine on 3 × 3 window .
- As can be seen in Figure 2.4, sorting the pixels in the 3 × 3 window gives $V_0 =$ [0 0 0 165 **202** 224 245 255 255] with $med = \mathbf{202}$. $D_V =$ [0 0 **165** **37** 22 10 0]. $v_L = \mathbf{0}$ and $v_R = \mathbf{165}$, which provide three clusters $\{0,0,0\}$, $\{165\}$, and $\{202,224,245,255,255\}$. The central pixel $I(x_0) = \mathbf{202}$ still belongs to the corrupted pixels clusters, which concludes that is a corrupted pixel and $C(x_0) = \mathbf{0}$.

2.3.2.2 Threshold Process

Finding the threshold value for each pixel is a crucial point for generating the binary pattern. Using a corrupted central pixel as a threshold value, as LBP, will affect the noise robustness of the descriptor. As well as, using a small or large region to obtain the median as a threshold value will affect the descriptor. This leads to biased median value, due to missing information for the small region or including a large number of pixels for the large region. For obtaining the threshold value, adaptive window and pixel classification are used to reach the maximum robustness.

Algorithm 3 represents the threshold process of the proposed descriptor, which starts by checking if the current pixel is classified as a corrupted or an uncorrupted pixel. If the current pixel classified as an uncorrupted pixel, the pixel threshold value is equal to the current pixel value (same as LBP). Otherwise, a 3 × 3 window is imposed around the current pixel and the number of the uncorrupted pixels is counted. If the number of the uncorrupted pixel is more than the corrupted ones, the threshold value is equal to the median of the uncorrupted pixels inside this window. Otherwise, the window will be enlarged by 1 pixel in all directions (5 × 5). This process will be repeated until the maximum window size is reached, where the threshold value is equal to the median value of all uncorrupted pixel inside that window.

Algorithm 3: Generation of local thresholds

```
Input: The original image $I$, the image of labeled pixels $C$, maximum window size $W_m$.
Output: Pixels threshold values $T$, and pixels corresponding window size $WS$.
for each pixel position $\boldsymbol{x_0}$ do
    if $I(x_0)$ is classified as uncorrupted pixel in $C$ then
        $T(x_0) = I(x_0)$
        $WS(x_0) = 1$
    else
        Initialize $w = 3$
        Impose a window $W$ $(w \times w)$ around $x_0$ $(W \in I)$
        Intialize $WS(x_0) = W_m$
        while $W < W_m$ do
            Find $N_{un} = \#$ uncorrupted pixels in $W$
            if $N_{un} \geq \frac{W^2}{2}$ then
                $WS(x_0) = W$
                Break
            else
                Update $W(w \times w)$, where $w = w + 2$
            end
        end
        Find $I_{un}$ (uncorrupted pixels in $WS(x_0)$)
        $T(x_0) = med(I_{un})$
    end
end
```

Figure 2.6 illustrates an example of obtaining the threshold value after classifying the pixels using the following procedure,

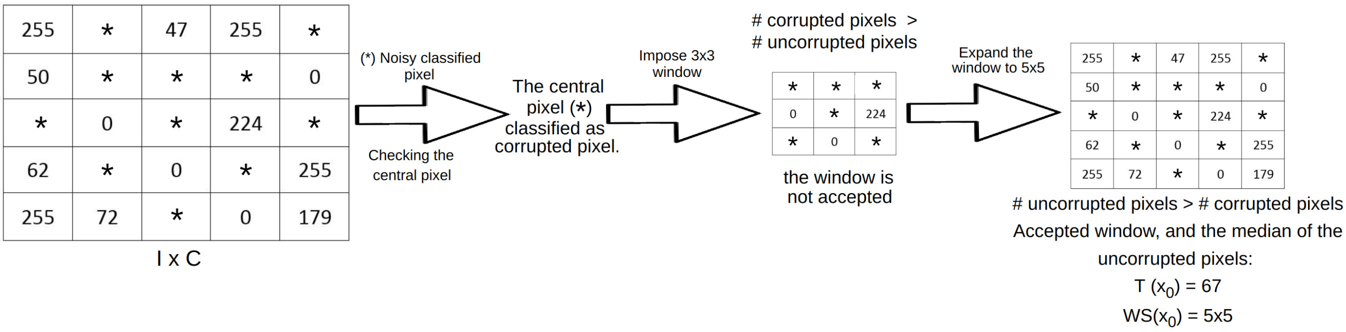


Figure 2.6: Threshold process example to facilitate understanding Algorithm 3.

- The current central pixel is classified as a corrupted pixel, which leads to impose a 3×3 window around it.
- The next step consists of checking whether the number of uncorrupted pixels is greater than the number of corrupted one. In the given example, $\#$ *uncorrupted pixel* $= 3$ while $\#$ *corrupted pixel* $= 4$, which followed by ignoring this window and enlarge it to be 5×5 window.
- In 5×5 window, $\#$ *uncorrupted pixel* $= 14$ while $\#$ *corrupted pixel* $= 11$. This window considered accepted window, and the threshold value will be obtained by taking the median value of the uncorrupted pixels, $T = med\{255, 47, 255, 50, 0, 0, 224, 62, 0, 255, 255, 72, 0, 179\}$ and equal to 67.

2.3.2.3 Generate the binary pattern

To reach the highest performance in texture classification, the descriptor should balance the classification goals such as robustness to noise, discriminativeness, and low computational cost. LBP descriptor conveys local structures, but to achieve better performance, discriminative properties should be used by considering the effect of image patches instead of taking a single pixel. To provide more information to the descriptor, these patches do not intersect with central pixel threshold window (Section 2.3.2.2). As well as, and each patch size will be found using an adaptive way that depends on each patch pixels. Figure 2.7 and Algorithm 4 demonstrates the binary pattern module of the proposed descriptor.

The binary pattern module (Algorithm 4) represents the procedure of forming the binary pattern. The module starts by finding the neighborhood patches with a maximum size around its central pixel. For each patch, a 3×3 window imposed around its central pixel. If the number of uncorrupted pixels is more than the corrupted pixels, this window considered accepted window and the value of the patch is the median of the uncorrupted pixels in that window. Otherwise, the window is enlarged to be 5×5 window. The process continues until reaching

Algorithm 4: Generation of binary patterns

```
Input: The original image $I$, the image of labeled pixels $C$, maximum window size $W_m$,
       pixels threshold values $T$, and pixels corresponding window size $WS$.
Output: The binary pattern (RAMBP).
for each pixel position $\boldsymbol{x_0}$ do
    The distance between the central pixel and each patch center $(n_i)$: $R = W_m + WS(x_0)$
    for each patch $P_i$ $(i \in 0:7)$ do
        Initialize $w = 3$
        Impose a window $W$ $(w \times w)$ around $P_i$ center $(n_i)$ $(W \in I)$
        Intialize patch window size $W_{P_i} = W_m$
        while $W < W_m$ do
            Find $N_{un} = \#$ uncorrupted pixels in $W$
            if $N_{un} \geq \frac{W^2}{2}$ then
                $W_{P_i} = W$
                Break.
            else
                Update $W(w \times w)$, where $w = w + 2$
            end
        end
        Find $I_{un}$ (uncorrupted pixels in $W_{P_i}$)
        $\beta = med(I_{un})$
        $S(P_i) = \begin{cases} 1, & T(x_0) \geq \beta \\ 0, & T(x_0) < \beta \end{cases}$
    end
    The binary pattern$(x_0) = \sum_{i=0}^{7} S(P_i) 2^i$
end
```

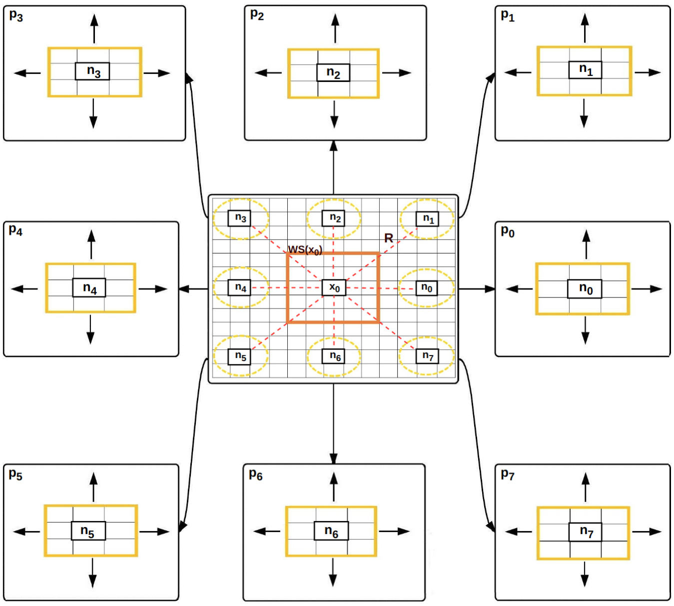


Figure 2.7: Binary module scheme (Algorithm 4). Where x_0 is the current central pixel, $WS(x_0)$ is x_0 corresponding window size (Algorithm 4), and P are the neighborhood patches with each central pixel (n).

the predefined maximum window size. After finding each neighborhood patch value, the binary pattern ($8bits$) is computed with a simple procedure between the patches values and the central pixel threshold value, where each patch represented in the binary pattern by 0 or 1.

2.4 Performance evaluation of RAMBP

The experiments were carried out with a core 7 Duo 3.50 GHz processor with 32GB RAM under Matlab. Nine texture datasets were conducted in these experiments, which considered from the most commonly used texture datasets. Table 2.1 summarized the used texture datasets, number of classes, number of images, images size, and each texture challenges.

To evaluate the robustness of the proposed approach, k-nearest neighbor (k-NN) had been used. The k-NN classifier recognized as one of the most popular and simplest methods, the k-NN is used with χ^2 distance defined as

$$\chi^2(x, y) = \frac{1}{2}\sum_i \frac{(x_i - y_i)^2}{x_i + y_i} \tag{2.4}$$

where x and y are the features vectors of two different textures. k-NN is adopted with k value equal to 1 for most experiments, but this parameter has been varied to test its influence on the performance consistency.

Table 2.1: Summary of the used Datasets.

Texture datasets	# of classes	# of images	Image size($pxls$)	Challenges
$Outex\_TC10$ [113]	24	4320	128×128	Rotation changes
$Outex\_TC11$ [113]	24	960	128×128	Inca illuminant, rotations (0°)
$Outex\_TC12$ [113]	24	4800	128×128	Illumination variations, rotation changes
$Outex\_TC23$ [113]	68	2720	128×128	Inca illuminant, rotations (0°)
Curet [114]	61	5612	200×200	Illumination variation, rotations and pose changes, specularities, shadowing
Brodatz [115]	111	999	215×215	Large number of classes, lack of intraclass variations
BrodatzRot [115]	111	999	128×128	Rotation changes, large number of classes, lack of intraclass variations
$KTH-TIPS2b$ [116]	11	4752	200×200	Pose changes, illumination changes, scale changes
ALOT [117]	250	25000	384×256	Strong illumination changes, rotation changes, large number of classes

In order to study the effect of the adaptive window maximum size, RAMBP performance has been tested on $Outex\_TC11$ dataset with different maximum window sizes. Figure 2.8 shows the classification score in different applied noises for different maximum window size values. As can be seen, the larger window size gives the higher score, but the time complexity will grow exponentially. Therefore, a good trade-off should be taken between the accuracy and the time complexity. In the experiments, 5×5 max window size is adopted since it gives high classification score and makes the algorithm run faster. In comparison with traditional LBP, the proposed method is slower but it has less computational complexity and dimensionality than many LBP descriptors used to address the noisy textures.

In this section, we start by evaluation of the proposed method on high noisy textures, including Salt-and-Pepper (impulse) noise, Gaussian noise, and Gaussian blur. Followed by experiment results of the proposed method on noise-free textures. In this paper, some of state-of-the-art descriptors results have been reported from [104].

2.4.1 Noisy texture classification

Noise robustness is a crucial point for evaluating descriptors. In this experiment, in order to test the noisy textures and evaluate the descriptor robustness in a more accurate way, the random noise generation has been repeated 10 times over the dataset, and the classification results had been noted by taking the average of these 10 tests. Noise-free images have been used for the training step while testing step performed on the noisy images. Choosing this scheme makes the noisy texture classification very difficult since the descriptor does not use any noise information

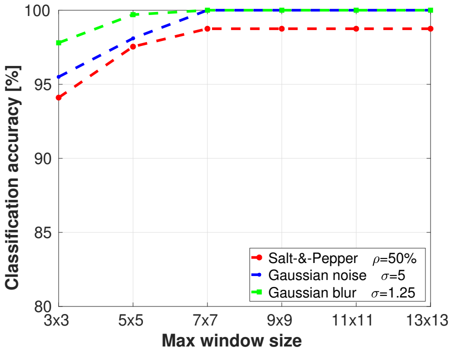
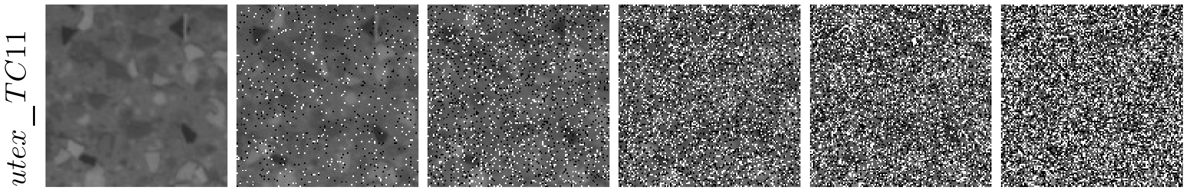


Figure 2.8: Illustration of the performance according to the maximum window size that the adaptive window could reach.

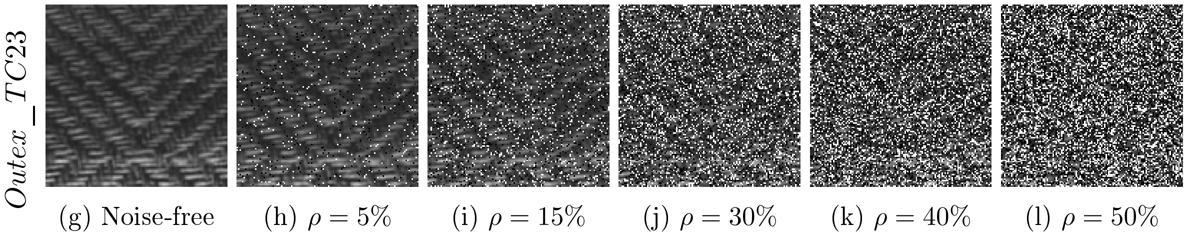
and any prior learning process.

2.4.1.1 Salt-and-Pepper noise

Salt-and-Pepper (impulse) noise introduces high or low values randomly distributed over the image. Salt-and-Pepper noise has been applied to $Outex\_TC11$ and $Outex\_TC23$ datasets with different noise densities ρ. High noisy textures are very challenging as it can be seen in Figure 2.9 where textures are visually unrecognizable from 30% of noise.



(a) Noise-free (b) $\rho = 5\%$ (c) $\rho = 15\%$ (d) $\rho = 30\%$ (e) $\rho = 40\%$ (f) $\rho = 50\%$



(g) Noise-free (h) $\rho = 5\%$ (i) $\rho = 15\%$ (j) $\rho = 30\%$ (k) $\rho = 40\%$ (l) $\rho = 50\%$

Figure 2.9: Example of $Outex\_TC11$ and $Outex\_TC23$ textures with different impulse noise densities.

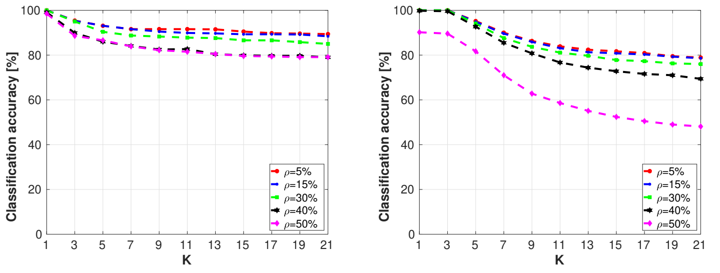
The results of the proposed algorithm are listed in Table 2.2. It can be observed that the classification accuracy is improved after using the proposed method. Compared to the different state-of-the-art techniques, RAMBP yields the best results and outperforms other techniques, especially on high noisy textures.

Table 2.2: Classification scores (%) comparison between the proposed descriptor (RAMBP) and state-of-the-art descriptors for Salt-and-Pepper noise.

Dataset	$Outex\_TC11$					$Outex\_TC23$				
Noise parameter	Noise density ρ					Noise density ρ				
Method	5%	15%	30%	40%	50%	5%	15%	30%	40%	50%
LBP [80]	85.4	15.5	5.4	4.2	4.2	66.0	9.9	3.8	1.8	1.5
$LBPriu2$ [80]	31.7	4.2	4.2	4.4	4.2	11.8	1.5	1.5	1.5	1.5
$LBPri$ [80]	47.1	10.0	4.2	4.2	4.2	26.5	4.7	2.2	1.5	1.5
$ILBPrui2$ [92]	27.3	4.2	4.2	4.2	4.2	10.7	2.1	1.5	1.5	1.5
$CLBP$ [93]	17.3	8.3	4.2	4.2	4.2	7.6	2.9	1.5	1.6	1.5
$MBPriu2$ [95]	31.0	8.3	4.2	4.2	4.2	17.0	2.5	1.5	1.5	1.5
MBP [95]	95.8	38.6	20.5	16.6	16.1	76.8	18.6	6.0	4.9	4.2
$RLBPriu2$ [94]	39.2	4.2	4.2	4.2	4.2	18.5	1.5	1.5	1.5	1.5
$EXLBP$ [118]	27.3	4.2	4.2	4.2	4.2	12.2	1.5	1.5	1.5	1.5
$NTLBP$ [119]	74.4	22.1	4.8	5.0	6.3	40.5	4.7	3.8	2.6	2.7
$MDLBPriu2$ [97]	71.9	13.5	8.3	4.2	4.2	38.2	3.7	2.9	2.5	1.9
$DLBP$ [81]	29.8	5.4	4.2	4.2	4.2	16.5	4.9	1.5	1.5	1.5
$BRINT$ [96]	30.8	7.1	6.0	4.4	4.2	15.9	1.5	1.5	1.3	1.5
$LBPD$ [120]	25.2	8.3	4.2	4.2	4.2	10.3	2.9	1.5	1.5	0.1
$SSLBP$ [99]	29.0	9.6	4.2	4.2	4.2	24.5	2.8	1.5	1.5	1.5
$AMBP$ [98]	100.0	95.4	20.7	13.8	10.7	100.0	85.0	4.8	1.8	1.5
$MRELBP$ [100]	100.0	100.0	100.0	85.8	50.2	100.0	99.9	94.0	54.6	19.2
$FV-VGGVD(SVM)$ [103]	21.0	12.1	6.0	6.5	4.2	10.3	5.2	2.3	1.5	1.8
$RAMBP$	**100.0**	**100.0**	**100.0**	**99.1**	**98.5**	**100.0**	**100.0**	**100.0**	**99.8**	**90.2**

As can be seen from the results, using rotational uniform scheme decreases the performance of the LBP based descriptors. Using LBP and MBP gave better results than $LBPriu2$ and $MBPriu2$, respectively. It can also be noticed from Table 2.2 that, $MRELBP$ offers the second best performance but its accuracy drops drastically with high noise densities (e.g. 50%). Also, $AMBP$ gives good results and noise robustness under low-density impulse noise but not for high noise.

Although RAMBP previously mentioned performance shows a high score where KNN ($k = 1$) provides the best match among all images, it is important to study the matched percentage of the same class images. This percentage can be computed using different k values in k-NN. In other words, an image is classified by the majority votes and assigned to the most common class. For example $k = 1$, KNN provides the nearest image, then the examined image will be classified as that image class. For $k = 3$, KNN provides the nearest three images and the examined image will be classified to class with the majority votes between the three images classes. For that, RAMBP performance has been tested with different k values in k-NN. We can notice from Figure 2.10a the stability and robustness of RAMBP in different Salt-and-Pepper noise densities where it keeps good accuracy even with high noise density and large value of k. Also shown in Figure 2.10b, the descriptor performance over k has a more decreasing rate which is proportional to the noise density. This happens may be due to the number of classes (i.e. 68) in $Outex\_TC23$ dataset. Nevertheless, the accuracy stays good for different k values.



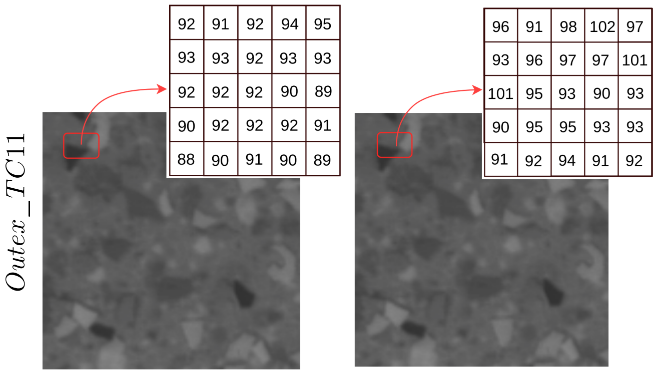
(a) $Outex\_TC11$ (b) $Outex\_TC23$

Figure 2.10: The performance of RAMBP for Salt-and-Pepper noise according to k values in k-NN. Where k starts from 1 to 21 with a step of two to avoid tie problem, and ρ parameter indicates Salt-and-Pepper density.

2.4.1.2 Gaussian noise

Gaussian noise is an additive noise affects digital images gray values. Gaussian noise has been added to $Outex\_TC11$ and $Outex\_TC23$ datasets with standard deviation $\sigma = 5$. Figure 2.11 provides an example of the used datasets after adding Gaussian noise, where visually it is difficult to see the global effect and the difference between noise-free and noisy textures, but it can be seen the local information and pixels intensity are affected.

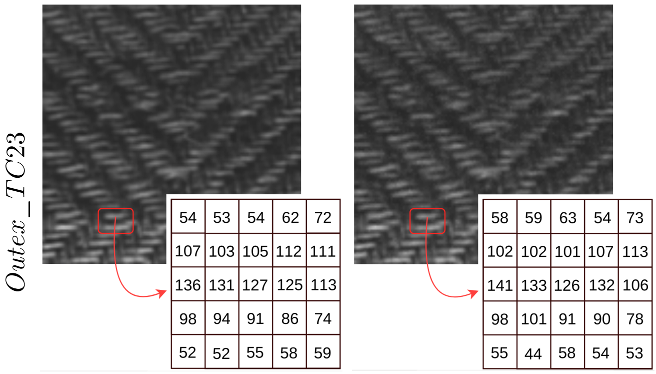
Table 2.3 shows the classification results of the proposed method as well as the state-of-the-art descriptors, where RAMBP provides the best performance among other descriptors. SSLBP descriptor gives the second best results, followed by MRELBP, AMBP, and deep learning



92	91	92	94	95
93	93	92	93	93
92	92	92	90	89
90	92	92	92	91
88	90	91	90	89

96	91	98	102	97
93	96	97	97	101
101	95	93	90	93
90	95	95	93	93
91	92	94	91	92

(a) Noise-free (b) $\sigma = 5$



54	53	54	62	72
107	103	105	112	111
136	131	127	125	113
98	94	91	86	74
52	52	55	58	59

58	59	63	54	73
102	102	101	107	113
141	133	126	132	106
98	101	91	90	78
55	44	58	54	53

(c) Noise-free (d) $\sigma = 5$

Figure 2.11: Example of $Outex\_TC11$ and $Outex\_TC23$ textures with Gaussian noise standard deviation $\sigma = 5$, where it shows the changes in pixels values.

techniques. But SSLBP yielded poor accuracy under Salt-and-Pepper as indicated in Table 2.2. As can be observed from Table 2.2 and Table 2.3, the proposed method achieved the best results in both experiments and showed nice consistency in different types of noise.

Table 2.3: Classification scores (%) comparison between the proposed descriptor (RAMBP) and state-of-the-art descriptors for Gaussian noise with standard deviation σ.

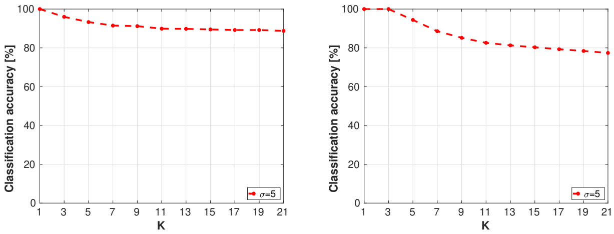
Dataset	$Outex\_TC11$	$Outex\_TC23$
Method	$\sigma = 5$	$\sigma = 5$
LBP [80]	35.0	09.8
$LBPriu2$ [80]	17.7	8.4
$LBPri$ [80]	16.0	7.9
$ILBPrui2$ [92]	17.5	10.4
$CLBP$ [93]	11.9	5.6
$MBPriu2$ [95]	12.1	5.2
MBP [95]	59.4	22.0
$RLBPriu2$ [94]	22.1	11.9
$EXLBP$ [118]	19.2	10.3
$NTLBP$ [119]	24.0	9.0
$MDLBPriu2$ [97]	12.5	6.1
$DLBP$ [81]	14.8	8.2
$BRINT$ [96]	61.9	27.4
$LBPD$ [120]	24.6	14.8
$SSLBP$ [99]	97.1	91.5
$AMBP$ [98]	96.5	74.3
$MRELBP$ [100]	91.5	79.2
$FV-VGGVD(SVM)$ [103]	93.1	71.5
$RAMBP$	**99.0**	**95.9**

To illustrate the stability of RAMBP for Gaussian noise, different k values in k-NN have been tested as shown in Figure 2.12. We can notice, a small decrease of RAMBP accuracy with increasing the value of k. Overall, RAMBP provides good stability and robustness even at large values of k in k-NN.

2.4.1.3 Gaussian blur

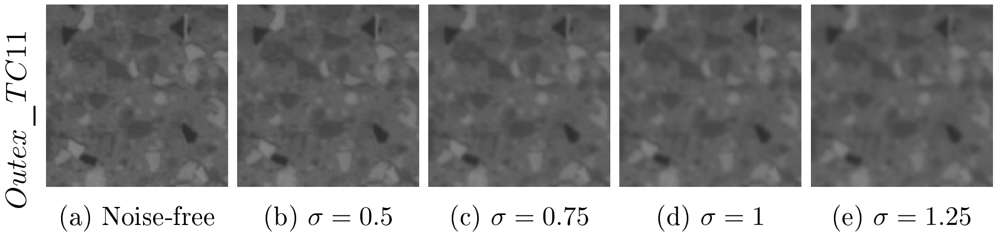
Gaussian blur, known also as Gaussian smoothing, is another kind of effects happened to images, which results in removing image detail. This type of noise also modify the local structure which affects the local binary patterns. In these experiments, Gaussian blur has been applied to $Outex\_TC11$ and $Outex\_TC23$ datasets with different standard deviations σ. Figure 2.13 illustrates an example of the used datasets with Gaussian blur.

Table 2.4 depicts the classification scores after applying Gaussian blur. The proposed method shows the best score with SSLBP method. The latest method performs nicely here

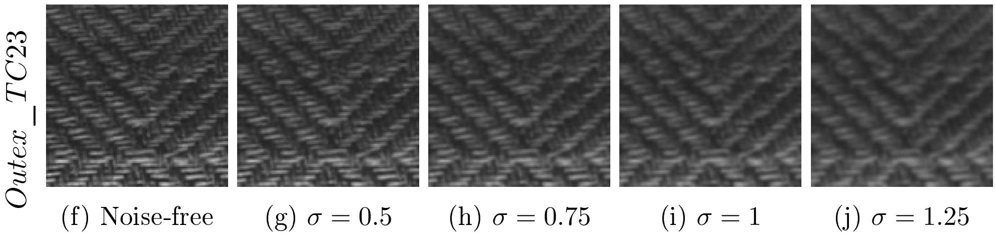


(a) *Outex_TC*11 (b) *Outex_TC*23

Figure 2.12: The performance of RAMBP for Gaussian noise with different k values in k-NN. Where k starts from 1 to 21 with a step of two to avoid tie problem, and σ parameter indicates Gaussian noise standard deviation.



(a) Noise-free (b) $\sigma = 0.5$ (c) $\sigma = 0.75$ (d) $\sigma = 1$ (e) $\sigma = 1.25$



(f) Noise-free (g) $\sigma = 0.5$ (h) $\sigma = 0.75$ (i) $\sigma = 1$ (j) $\sigma = 1.25$

Figure 2.13: Example of *Outex_TC*11 and *Outex_TC*23 textures with different Gaussian blur standard deviation.

because it includes the blurring process in descriptor generation. However, it must be recalled the poor performance of SSLBP in Salt-and-Pepper noise as evidenced in Table 2.2. MRELBP and FV-CNN have good performance under low noisy textures, but the accuracy vastly decreases under higher noise.

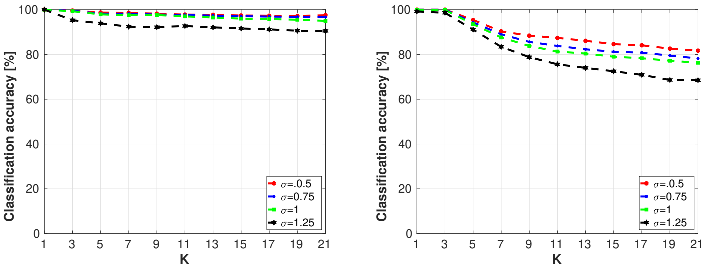
Table 2.4: Classification scores (%) comparison between the proposed descriptor (RAMBP) and state-of-the-art descriptors for Gaussian blur with standard deviation σ.

Dataset	$Outex\_TC11$				$Outex\_TC23$			
Method	$\sigma = 0.5$	$\sigma = 0.75$	$\sigma = 1$	$\sigma = 1.25$	$\sigma = 0.5$	$\sigma = 0.75$	$\sigma = 1$	$\sigma = 1.25$
LBP [80]	99.1	71.8	53.8	39.8	99.9	55.0	37.8	29.0
$LBPriu2$ [80]	94.2	46.5	24.6	12.7	72.4	30.3	16.6	9.7
$LBPri$ [80]	86.9	44.6	26.0	18.1	57.7	28.3	16.0	9.4
$ILBPrui2$ [92]	97.3	59.8	29.4	20.4	81.7	43.2	25.1	16.7
$CLBP$ [93]	98.8	74.8	49.6	23.1	86.6	55.4	36.1	21.2
$MBPriu2$ [95]	85.4	29.0	18.5	11.9	58.7	22.5	13.5	10.6
MBP [95]	97.5	73.0	50.9	39.5	99.0	58.2	40.7	28.3
$RLBPriu2$ [94]	95.0	49.8	28.7	16.5	75.4	33.2	18.4	10.7
$EXLBP$ [118]	94.0	47.7	28.3	17.1	73.3	32.0	17.8	10.5
$NTLBP$ [119]	96.3	49.0	33.1	19.4	80.1	35.7	21.7	14.1
$MDLBPriu2$ [97]	100.0	60.2	36.9	23.8	95.7	35.1	20.6	12.2
$DLBP$ [81]	90.4	61.5	21.9	13.1	67.7	31.3	16.5	8.7
$BRINT$ [96]	100.0	97.1	80.4	44.6	100	97.5	59.1	39.1
$LBPD$ [120]	99.4	85.8	65.2	45.4	87.7	56.0	40.2	30.6
$SSLBP$ [99]	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	90.6
$AMBP$ [98]	100.0	99.0	88.7	52.6	100.0	99.5	81.4	53.8
$MRELBP$ [100]	100.0	100.0	93.8	75.4	99.9	97.9	85.8	61.8
$FV - VGGVD(SVM)$ [103]	100.0	100.0	96.5	89.8	99.6	94.1	83.1	71.8
$RAMBP$	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**99.2**

The accuracy of the proposed method can also be observed in Figure 2.14 after varying k values in k-NN. In the same Figure 2.14, the classification accuracy is high and gets affected with a small decrease after increasing the standard deviation and the value of k. It can be seen that RAMBP has high stability and robustness among different k values. In general, RAMBP achieved the best results compared to the different state-of-the-art techniques as apparent in Table 2.2, 2.3 and 2.4.

2.4.1.4 Mixed noises

Most state-of-the-art methods deal with a single kind of noise such as Gaussian noise or impulse noise. For that reason, it is interesting to examine the ability of the RAMBP descriptor to classify textures with a mixture of a Gaussian noise, Gaussian blur, and impulse noise. In these experiments, Gaussian noise, Gaussian blur and impulse noise were applied to $Outex\_TC11$ with different standard deviations σ and different noise densities ρ. After applying mixed-noise,



(a) *Outex_TC*11 (b) *Outex_TC*23

Figure 2.14: The performance of RAMBP for Gaussian blur with different k values in k-NN. Where k starts from 1 to 21 with a step of two to avoid tie problem, and σ parameter indicates Gaussian blur standard deviation.

Tab. 2.5 shows the classification performances of the proposed method and some state-of-the-art methods as a baseline, where it is shown that the proposed method achieved the best performance. It can also be seen in Tab. 2.5 that the order in which each noise is applied has a certain impact on the results: thus, applying Salt-and-Pepper noise followed by Gaussian blur gives a lower score than the other way round. This is due to smoothing of the local high impulse noisy structure which leads to spreading the noise. Overall, Tab. 2.5 shows promising results and opens a new challenge of mixed noise texture classification.

2.4.2 Noise-free texture classification

Noise-free texture classification is challenging due to datasets properties mentioned in Table 2.1. In these experiments, seven texture datasets have been used. The training and testing schemes are different from one dataset to another. For $TC10$ and $TC12$ Outex datasets, testing and training samples are well-defined by [80]. For $TC10$ the training set has no rotation, and the testing set is rotated by $\{5°, 10°, 15°, 30°, 45°, 60°, 75°, 90°\}$ rotation angles. Also, $TC23$ the training set has no rotation, while the testing set is rotated by $\{0°, 5°, 10°, 15°, 30°, 45°, 60°, 75°, 90°\}$ rotation angles.

BrodatzRot is generated to test rotation invariance by applying a random rotation angle for each sample in *Brodatz*. For *Curet*, *Brodatz*, *BrodatzRot*, and *ALOT* datasets, each class samples was divided equally (50% train/test) using a random selection of the samples. 100 random couple train/test sets were generated and the classification results are averaged over the 100 random partitionings. For *KTH2b* dataset which has four samples of 11 classes each, training is performed on three samples and testing on the remaining one, the results are obtained by performing the experiment four times.

The result of texture classification is depicted in Table 2.6, where RAMBP provides the best results in some datasets and high performance for the others. Also, FV-CNN, SSLBP and MRELBP techniques show high and competitive performances. However, since RAMBP does not use any learning process and provides high performance for different kind of noises, RAMBP stands out the best descriptor in noisy and noise-free texture classification.

Table 2.5: Classification scores (%) comparison between the proposed descriptor (RAMBP) and some state-of-the-art descriptors as a baseline for mixed noisy $Outex\_TC11$ texture with a standard deviation σ and a noise density ρ (Gaussian noise(GN), Gaussian blur(GB) and Salt-and-Pepper(SP)).

Method	$GN - GB$	$GB - GN$	$GN - SP$	$SP - GN$	$GB - SP$
Order	$GN_\sigma = 5$ $GB_\sigma = 1$	$GB_\sigma = 1$ $GN_\sigma = 5$	$GN_\sigma = 5$ $SP_\rho = 30$	$SP_\rho = 30$ $GN_\sigma = 5$	$GB_\sigma = 1$ $SP_\rho = 30$
LBP [80]	53.65	32.61	7.51	8.33	10.42
$LBPriu2$ [80]	19.69	16.15	4.16	4.16	4.16
MBP [95]	75.0	46.25	6.67	7.29	8.3
$AMBP$ [98]	98.06	92.9	7.71	8.96	27.40
$RAMBP$	**99.4**	**99.6**	**97.9**	**93.4**	**95.7**

Method	$SP - GB$	ALL	ALL	ALL	ALL
Order	$SP_\rho = 30$ $GB_\sigma = 1$	$GN_\sigma = 5$ $GB_\sigma = 1$ $SP_\rho = 30$	$GB_\sigma = 1$ $GN_\sigma = 5$ $SP_\rho = 30$	$SP_\rho = 30$ $GN_\sigma = 5$ $GB_\sigma = 1$	$SP_\rho = 30$ $GB_\sigma = 1$ $GN_\sigma = 5$
LBP [80]	1.16	6.36	5.21	1.78	4.16
$LBPriu2$ [80]	4.16	4.16	4.16	4.16	4.16
MBP [95]	6.98	4.16	4.16	6.77	6.77
$AMBP$ [98]	4.16	21.88	13.96	4.16	4.16
$RAMBP$	**56.7**	**93.3**	**91.5**	**49.3**	**52.1**

Table 2.6: Classification scores (%) comparison between the proposed descriptor (RAMBP) and state-of-the-art descriptors for noise-free texture classification.

Method	$TC10$	$TC12$	$Curet$	$Brodatz$	$BrodatzRot$	$KTH2b$	$ALOT$
$\#classes$	(24)	(24)	(61)	(111)	(111)	(11)	(250)
LBP [80]	99.36	90.55	92.77	88.67	76.48	60.33	86.58
$LBPriu2$ [80]	99.69	92.16	97.03	90.70	79.22	62.69	94.15
$LBPri$ [80]	86.69	83.68	95.38	89.93	71.73	61.48	93.29
$ILBPrui2$ [92]	99.66	93.34	94.66	91.66	82.27	61.93	95.71
$CLBP$ [93]	99.45	95.78	97.33	92.34	84.35	64.18	96.74
$MBPriu2$ [95]	95.29	86.69	92.09	87.25	74.57	61.49	88.23
MBP [95]	98.52	97.17	91.24	89.27	76.67	60.19	91.30
$RLBPriu2$ [94]	99.66	93.53	97.20	91.09	79.59	61.20	94.23
$EXLBP$ [118]	99.64	93.55	96.85	90.19	80.08	62.39	95.20
$NTLBP$ [119]	99.32	95.27	96.11	89.31	80.25	61.30	94.47
$MDLBPriu2$ [97]	99.22	95.64	96.92	93.40	82.31	66.52	95.81
$DLBP$ [81]	99.46	91.97	94.38	88.73	75.04	61.72	NO
$BRINT$ [96]	99.35	98.13	97.02	90.83	78.77	66.67	96.13
$LBPD$ [120]	98.78	96.67	94.23	89.74	74.79	63.47	92.82
$SSLBP$ [99]	99.82	99.36	98.79	89.94	80.03	65.57	96.68
$AMBP$ [98]	99.68	98.12	95.64	90.67	79.86	62.73	95.82
$MRELBP$ [100]	99.82	99.58	97.10	90.86	81.92	**68.98**	97.28
$FV-VGGVD(SVM)$ [103]	80.00	82.30	**99.00**	**98.70**	**92.10**	**88.20**	**99.50**
$RAMBP$	**99.90**	**99.70**	**98.50**	**94.05**	**86.98**	68,86	**97.59**

2.4.3 Computational complexity

Time complexity is considered a crucial point, especially for texture classification. Important aspects that affect the running time for any descriptor are feature extraction time and the feature dimensionality. In the literature, it is generally the performance results of the descriptor that are focused on, while time complexity has received less attention. Table 2.7 demonstrates the feature extraction time and dimensionality of RAMBP for the $Outex\_TC11$ database. In Table 2.7, the average time is reported for each method. This average is calculated over 480 images of size 128×128 without including the training time for some methods marked as ($\star$). Different schemes are defined for parameters radius (r) and number of neighbor members (p), which affect the feature dimensionality and feature extraction time. Feature dimensionality, which is the final dimensionality of each method provided to the classifier, has been noted. It can be observed from Table 2.7 that $FV - VGG$ is the most computationally expensive method for feature extraction and has a very high feature dimensionality. RAMBP is slower than few state-of-the-art methods, but due to its strength and robustness in different kinds of noise and noise free classification, as well as its low dimensionality, it provides promising and competitive results, in addition to the possibilibity of using GPUs and parallel programming which ensure a real-time process.

Table 2.7: Computational complexity comparison between the proposed descriptor (RAMBP) and state-of-the-art descriptors. Different schemes for parameters (r, p) are defined. Sch 1: $(1, 8)$, $(2, 16)$, $(r, 24)$ for $3 \leq r \leq 9$; Sch 2: $(r, 8)$ for $1 \leq r \leq 9$; Sch 3 $(1, 8)$ and $(r, 24)$ for $2 \leq r \leq 9$; Sch 4: $(1, 8)$, $(2, 16)$ and $(3, 24)$; Sch 5: $(1, 8)$, $(3, 8)$ and $(5, 8)$; Sch 6: $(r, 8)$, $r = 2, 4, 6, 8$; Sch 7: $(3, 24)$, $(9, 24)$. Methods labeled with ($\star$) are those requiring extra computation time at the feature extraction stage.

Method	Scheme	Feature extraction time (ms)	Feature dimensionality
$LBPriu2$ [80]	Sch 1	87.2	210
$LBPri$ [80]	Sch 2	47.5	108
$ILBPrui2$ [92]	Sch 1	90.8	420
$CLBP$ [93]	Sch 1	127.9	3552
$MBPriu2$ [95]	Sch 1	215.6	420
$RLBPriu2$ [94]	Sch 1	488.6	210
$EXLBP$ [118]	-	91.3	273
$NTLBP$ [119]	Sch 1	332.3	388
$MDLBPriu2$ [97]	Sch 5	26.3	1000
$DLBP$ [81]	Sch 1	($\star$)565.3	14150
$BRINT$ [96]	Sch 3	248.8	1296
$LBPD$ [120]	-	54.2	289
$SSLBP$ [99]	Sch 7	($\star$)180.0	2400
$AMBP$ [98]	Sch 4	92.5	1536
$MRELBP$ [100]	Sch 6	416.6	800
$FV - VGGVD(SVM)$ [121]	-	($\star$)2655.4	65536
$RAMBP$	-	225.4	256

2.5 Conclusion

It is well known that the visual properties of US images are degraded by many effects such as artifacts, signal degradation, and speckle noise. These effects make detection and tracking very challenging. For that, the use of powerful and noise-robust features is a crucial step for accurate detection and tracking. In this chapter, we proposed a new efficient and robust descriptor, Robust Adaptive Median Binary patterns (RAMBP).

The RAMBP descriptor uses pixel classification and adaptive analysis to ensure strong discriminativeness and noise robustness. Since the US image shares textures properties, RAMBP was tested against different descriptors for noisy texture classification. The proposed descriptor was evaluated on noisy textures including Salt-and-Pepper, Gaussian noise, Gaussian blur, and mixed noise. Experimental results indicated that RAMBP outperforms other existing descriptors in handling high-noise texture classification and performs as one of the best in noise-free texture classification. Moreover, RAMBP achieves a good running time and low feature dimensionality.

RAMBP descriptor will be used and evaluated in the next chapter for the detection and tracking in US images.

Chapter 3

Nerve Detection & Tracking

Outline:
Accurate and consistent nerve detection and tracking is essential for safe and efficient robotic-assisted UGRA system. In this chapter, different nerve detection and tracking techniques are evaluated in terms of accuracy, consistency, time complexity, and handling different nerve situations.

Contents

3.1 Introduction

The UGRA procedure requires anatomical knowledge and skills in the use of US imaging, as well as a high degree of concentration by the anesthetist in locating the nerve [5, 6]. In addition, it is not always easy to locate certain nerves in the practice of UGRA. In this work, a robotic-assisted UGRA system is proposed where this chapter aims to develop tools to assist the anesthetists with accurate nerve detection and tracking.

Although there has been extensive development of detection, and tracking algorithms for US images [122–125], it is still an open problem, especially for RA. So far, very little attention has been paid to the nerve detection and tracking. In the previous chapter, we proposed Robust Adaptive Median Binary Patterns (RAMBP) which has the advantages of effectiveness and robustness to textures with different high noises. Using detectors or trackers based on RAMBP would provide a good performance, since the nerve regions considered as a noisy texture. However, we believe Convolutional Neural Networks (CNNs) models are effective too for US image textures. Thus, to learn more optimal features, it is more interesting to merge RAMBP with CNN models to achieve better performance for nerve detection and tracking.

In this chapter, we propose different methods to detect and track the nerve in US images in Section 3.2 and 3.3 respectively. These methods are based on texture descriptor (RAMBP), CNNs models, and the proposed merged model (texture descriptor with CNNs). In Section 3.4, experimental setups, results, and discussion are detailed. The chapter ends with the final conclusions in Section 3.5.

3.2 Nerve detection

In the literature, several methods have been implemented to deal with the problem of object detection. These works accomplished in various applications such as medical imaging [126], surveillance [127], robotics [128], etc. There exist few researches aiming for nerve region detection, where these studies based on using texture descriptors [71] or CNN [129] architectures to differentiate the nerve region from other regions in the US images.

Texture is an important cue for detection in many US images applications since some tissues show specific texture properties, such as certain nerves. The traditional approaches use handcrafted texture features to classify textures in US images [130]. In [131], the authors proposed a descriptor based on the combination of median binary pattern and Gabor filter to characterize and classify the median nerve tissues.

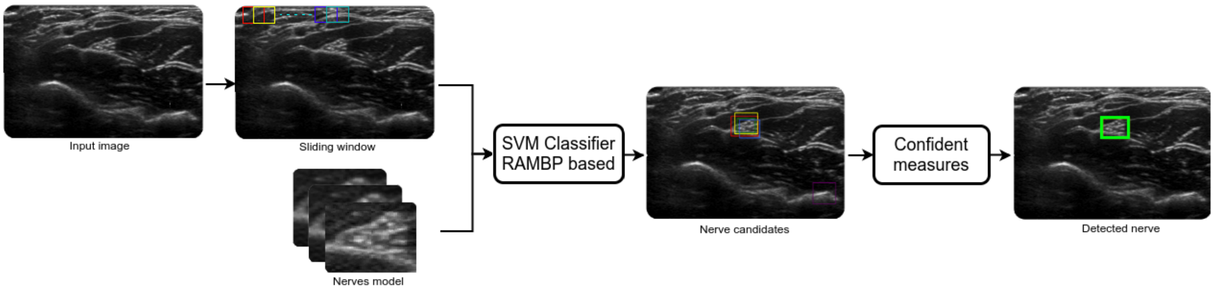
On the other hand, CNN [132] have received significant attention in computer vision and machine learning applications such as object detection [133], image classification [134], and image segmentation [135]. Motivated by these breakthroughs, several CNN based methods have been developed in order to significantly improve object detection performance in the US medical images [136, 137]. However, very little attention has been paid to nerves in the US images [125, 138, 139]. In [129], the authors used a convolutional neural network and spatiotemporal consistency to segment efficiently the nerve region.

Despite the promising results obtained by the texture descriptors or the CNN based detectors, still, the nerve detection topic requires further development and examination. In this section, we propose new techniques for detecting the nerve in the US images. These techniques are based on texture descriptor (RAMBP), CNNs models, and merged model (RAMBP with CNNs models).

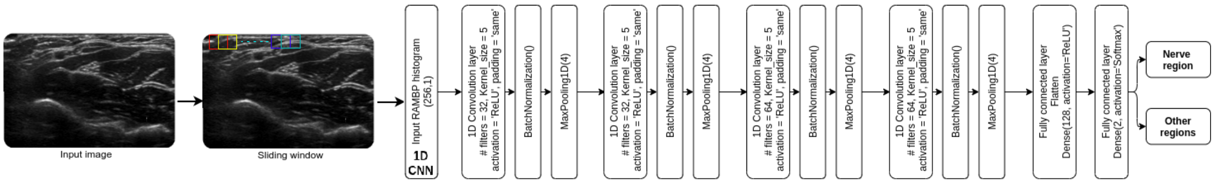
3.2.1 Nerve detection using RAMBP descriptor

In the current section, we present nerve detection using RAMBP descriptor using two techniques, Support Vector Machine (SVM) [140] and 1D CNN [132]. Although 1D CNN considered a CNNs model but, here in this section, we use RAMBP descriptor as input for the 1D CNN. Figure 3.1 illustrates the entire processing chain of the two techniques.

Extracting the features is one of the essential steps in detecting the nerve in the US image. This step consists of extracting significant information from the nerve. The visual properties (features) of nerve tissues in US images can vary from patient to patient. In addition, the position of the probe can significantly affect these properties. To deal with such a situation, the SVM and 1D CNN models are trained on different nerve appearances.



(a) SVM



(b) 1D-CNN

Figure 3.1: Schematic of nerve detection using RAMBP descriptor. (a) SVM classifier. (b) 1D CNN classifier.

As shown in Figure 3.1a and for the SVM classifier, first, feature extraction was performed to represent the nerve texture where the RAMBP is presented in Chapter 2. Then, the 1D RAMBP histogram is used with an SVM algorithm for the training and testing phase which results in building an SVM model.

For 1D CNN and similar to SVM classifier, RAMBP descriptor was performed to represent the texture of the nerve. The 1D CNN architecture is constructed to learn deep visual features from RAMBP histogram. As shown in Figure 3.1b, the 1D architecture has four 1D convolutional layers, four 1D max-pooling layers, four batch normalization layers, a fully connected layer, and the top layer of this architecture is the ReLU layer. The first convolution layer has 32 convolution kernels of size 5 with the 'same' padding (the output feature map has the same spatial dimensions as the input feature map). When this 1D vector is passed into the first convolution layer, 32 various learned filters produce different responses considered as the extracted features. This followed by normalizing the features by the first batch normalization layer. After that, these features are passed to the first pooling layer with a size of 4. As shown in Figure 3.1b, the reduced features from the first max-pooling layer are passed to the second convolution layer. After that, the sequence of layers is repeated by taking the second, third, and fourth convolution, batch normalization, and max-pooling layers, but the third and fourth convolution layers have kernels of size 5 and produce 64 different features. The outputted features by the fourth max-pooling layer are flattened and inputted into the fully connected layer. Finally, the features are passed through the last fully connected layer where the nerve images will be recognized by the softmax classifier in the top layer of this architecture.

As illustrated in Figure 3.1, several US images were used as a training set in order to represent different aspects of the nerve. To detect the nerve, the SVM and 1D CNN models are used to compare the sliding windows in the input image (test) and the models. For SVM, the classification procedure generates a list of candidate region positions for each nerve. From these positions, the final nerve position is assigned to the region where the confidence level is the highest as shown in Figure 3.1a. On the other side and for 1D CNN, the classification procedure output is the prediction score of each sliding window in the image. From these scores, the final nerve position is assigned to the region with the highest score.

3.2.2 Nerve detection using CNNs model

In this section, the 2D CNN architecture is constructed to learn deep visual features from US images. As observed in Figure 3.2, this architecture has four 2D convolutional layers, four batch normalization layers, two 2D max-pooling layers, and a fully connected layer as shown in Figure 3.2. The first convolution layer has 32 convolution kernels of size 3 × 3 with the 'same' padding. The output of the first convolution layer produces 32 various feature maps. This followed by normalizing the features by the first batch normalization layer. Then, the normalized features are inputted into the second convolution and batch normalization layers, which have the same parameters as the first layers. The normalized learned features are reduced by the first max pooling layer with a size of 2 × 2. After that, the sequence of layers is repeated by taking the third and fourth convolution and batch normalization layers, followed by the second max pooling layer with a size of 4 × 4. The third and fourth convolution layers have the same parameters as the first two layers but with 64 various learned filters that produce 64 different feature maps. Then, the features produced by the second max-pooling layer are flattened and passed into the fully connected layer. Finally, the features are passed through the last fully connected layer where the nerve images will be recognized by the softmax classifier in the top layer of this architecture.

As illustrated in Figure 3.2, several US images were used as a training set in order to

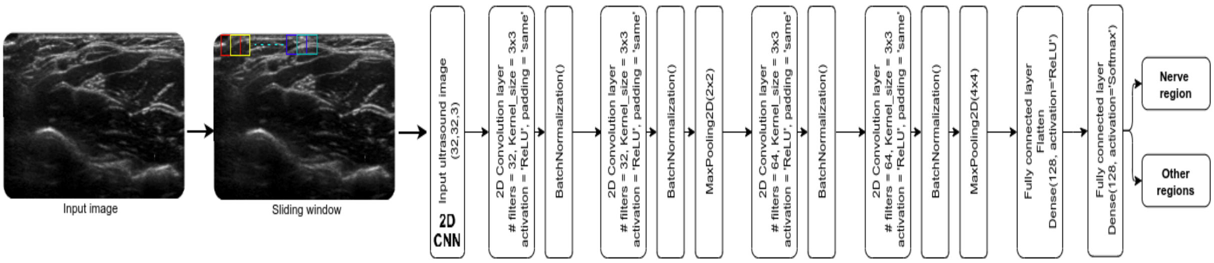


Figure 3.2: Schematic of the 2D CNN architecture.

represent different aspects of the nerve. To detect the nerve, the 2D CNN models are used to compare the sliding windows in the input image (test) and the models. The classification procedure output is the prediction score of each sliding window in the image. From these scores, the final nerve position is assigned to the region with the highest score.

3.2.3 Nerve detection using the merged model

Despite the strong characteristics of texture descriptors or the CNN based detectors, still, the nerve detection topic requires further development and examination. Thus, to learn more optimal features, we propose to learn high-level features from different dimensional data using different-dimensional CNNs. The merged deep CNN architecture has two branches, a one-dimensional (1D) CNN branch (RAMBP 1D histograms) and a 2D CNN branch (2D US images) as shown in Figure 3.3. These two architectures aim to learn deep features and concatenate them to form the merged deep network.

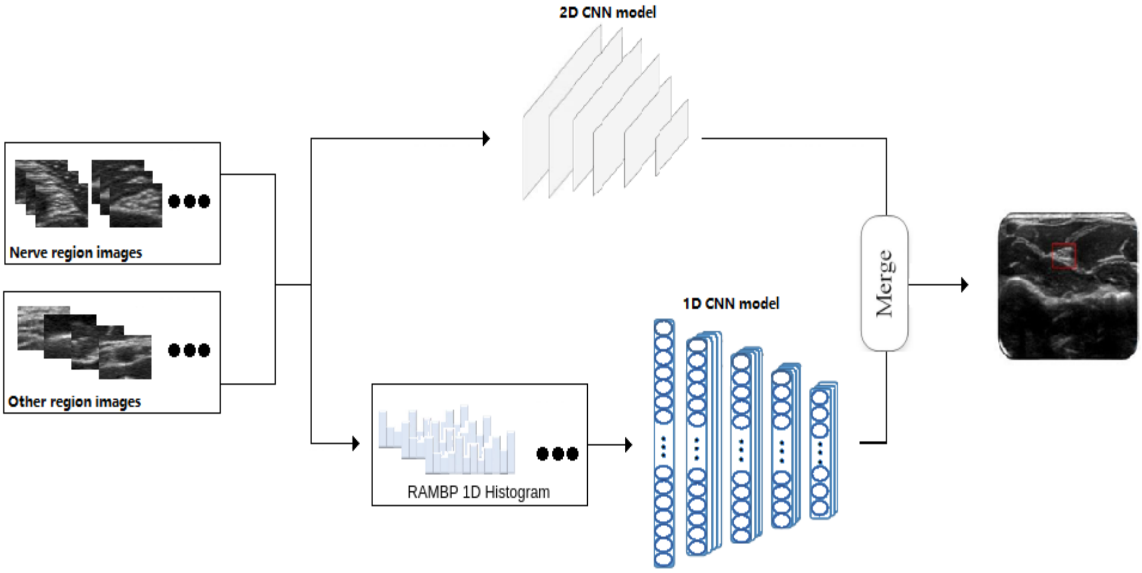


Figure 3.3: Schematic of the merged 1D-2D CNN architecture.

As shown in Figure 3.4, the merged model is build from the 1D and 2D CNN model which are presented in Section 3.2.1 and 3.2.2, respectively. The merged model remove the second fully connected layer from each model and replace it by a merge layer. After the deletion of the last fully connected layer, the 1D CNN and 2D CNN are merged by a merged layer. The merged layer can be expressed as

$$z^l = \left[z_{1D}^{l-1} \;\; z_{2D}^{l-1} \right] \tag{3.1}$$

where z_{1D}^{l-1} and z_{2D}^{l-1} are the learned features produced by the fully connected layers of the 1D and 2D CNN branches, respectively.

When the concatenated features are passed through the last fully connected layer, nerve region will be recognized by the softmax classifier in the top layer of this architecture.

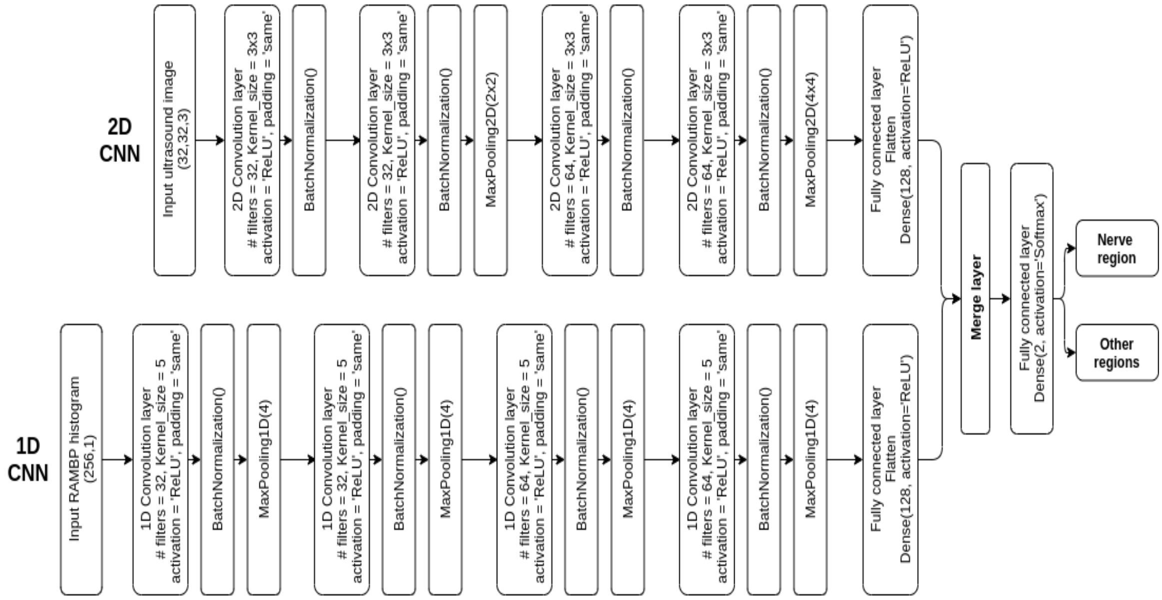


Figure 3.4: An illustration of the merged CNN model.

3.3 Nerve tracking

Tracking is one of the fundamental tasks in computer vision and image analysis, and it is used in a wide range of applications such as video surveillance, medical imaging, robotics, etc. Tracking is an easy task when the target objects are isolated and easily distinguishable from the background, but it is a very challenging task when the image suffers from illumination changes, shape deformation, object disappearance, viewpoint variation, etc [141]. Tracking failure could happen easily under the noise, illumination changes, occlusion, and deformation of the target. Therefore, tracking the nerve in US imaging modality is a very challenging task.

Tracking in US images is a very challenging task due to the degradation of the visual property of US images. Various methods have been proposed in the literature regarding tracking in US images. In [142], Li et al. improved active contour by incorporate intensity information with edge gradient to track the tongue in US images. In [143], the author used Markov random field for segmenting and tracking automatically the tongue contour. Nascimento et al. used multiple model data association tracker with nonlinear filters to track the left ventricle in the US images [144]. In [145], the authors proposed a variant of active appearance modeling to detect and track the tongue in US images. The technique proposed in [146]based on building an observation distribution using deep neural networks, which have been used with the systole and diastole motion patterns to track the left ventricle in the US images. In [147], a new technique is proposed using a Kalman filter with an elliptical model to track the center of vessels in US images. Novotny et al. searched for long straight objects by using generalize Radon transform and Graphics Processor Unit (GPU) for tracking medical instruments in three-dimensional US images [148]. In [149], the authors used region-based tracking to track endocardium in three-dimensional US images. As far as we know, there is no tracking study addresses nerve regions in US images.

In this section, we propose different techniques for tracking the nerve in the US images. These techniques are based on texture descriptor (RAMBP), CNNs models, and the proposed merged model (RAMBP and CNNs model).

3.3.1 Nerve tracking based on RAMBP descriptor

Feature tracking offers a wide range of application possibilities in computer vision and control theories, such as medical robotics, surveillance, etc. A robust extraction and real-time tracking of features is a big step to the success of other tasks (e.g. visual servoing). Among popular feature tracking methods, there are three robust and efficient tracking algorithms, particle filter [150], mean-shift [151] and Kanade-Lucas-Tomasi (KLT) [152]. Particle filter and mean-shift algorithms used to find the similarity between the current image and the target nerve, and it is based on histogram representations. KLT used the frame to frame object tracking, which consists of matching object feature points or descriptors between these images and calculate the displacement vector.

LBP descriptor family have been used for tracking in many US images applications, it presents many advantages such as discrimination efficiency, illumination changes, and rotation invariance, and low computation cost [80]. In [153], Ning et al. used LBP for describing color-texture histogram with a mean-shift algorithm. In [154], the author improved the particle filter to handle illumination changes by using LBP as a color-texture histogram. Histograms of Oriented Gradients (HOG) is also a powerful descriptor, Bilinski et al. performed multi-object tracking under occlusion using HOG descriptors [155]. However, the previously mentioned texture-based tracking methods have some notable limitations, mostly the sensitivity to noise. To the best of our knowledge, there is no study that addresses the problem of noisy texture tracking, particularly nerve tracking for RA application.

Since we deal with texture regions in US images it would be more interesting to use robust descriptors. Several studies aim to increase the robustness of noisy textures classification using robust local binary pattern approach [95, 96, 112, 156, 157]. Figure 3.5 describes the tracking process where the used deterministic tracking algorithms are combined with different kinds of feature extraction methods. The process starts by taking the nerve previous location and input it to the feature descriptor (RAMBP). This followed by applying the feature tracker. These trackers compare the previous nerve location features with the features that had been taken from the candidate nerve location.

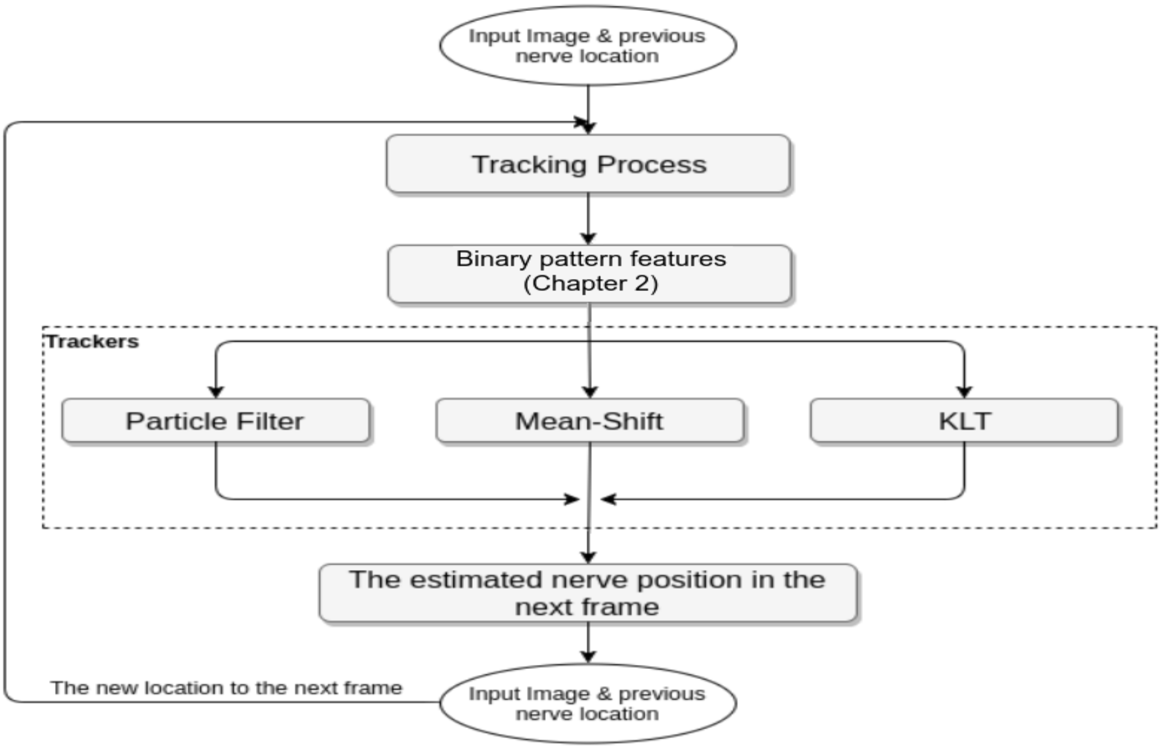


Figure 3.5: The flowchart of the tracking process.

3.3.2 Nerve tracking based on CNNs models

It is more beneficial to exploit recent trackers based on deep learning processes, since it has shown excellent performance in many computer vision applications. Motivated by CNNs breakthroughs, several deep-learning based trackers have been developed in order to significantly improve the tracking performance. These research works showed promising results for different tracking applications, such as [158–169].

In the literature, few methods introduced the tracking using deep-learning in US images. In [146], the authors built a deep neural network observation distribution to track the left ventricle endocardium in US images. In [170], The authors used deep neural networks to build a new observation model in a particle filter to track and segment the left ventricle in US images.

This section aims to track nerves in US images using deep-learning methods, and as such these methods should be robust enough to track different nerve situations. The visual tracker starts by generating the target model in the first frame, then extracts features in the next frame to find candidates' models, and finds the best match between target and candidate models. Most existing deep trackers use CNN either to generate appearance models, to match object model with its candidates, or to distinguish the object from the surrounding areas. Therefore, 13 deep-learning tracking methods are tested to track the nerve in the US images.

In this work, we conducted a comparative study of thirteen deep trackers for tracking the nerve in the US images. These trackers are built by stacking different CNN layers to track the nerve in US images. The trackers are Continuous Convolution Operators Tracker (C-COT) [158], Efficient convolution operators (ECO) [159], Convolutional Network based Tracker (CNT) [160], Multi-Domain convolutional neural Networks (MDNet) [161], Structure-Aware Network (SANet) [162], Fully-Convolutional Siamese Networks (SiameFC) [163], Correlation Filter Network (CFNet) [164], Discriminant Correlation Filters Network (DCFNet) [165], Multi-task Correlation Particle Filter (MCPF) [166], Hedged Deep Tracking (HDT) [167], Hierarchical Convolutional Features Tracker (HCFT) [168], Deep-Learning Tracker (DLT) [141], and Convolutional RESidual Tracker (CREST) [169]. Experimental detail is presented in Section refexper.

3.3.3 Nerve tracking based on 1D-2D merged model

As Section 3.2.3 discussed the possibility of merging texture descriptors with CNNs model for nerve detection, it is interesting, also, to use the merged model the tracking task. Herein, we propose to use the merged 1D-2D CNNs model (Section 3.2.3) with three tracking techniques to track the nerve in US images. The first technique is based on detection using the merged CNNs model in the next frame search area bigger than the nerve previous location. The second technique is based on particle filter (PF) tracker [150] where the merged CNNs model is responsible for providing the particle weights. The third technique combines the first two trackers.

The first tracker (the merged 1D-2D CNNs based search tracker) starts by finding the search window (area). The search window is 4 times bigger than the previous nerve location where this location represented by a window around the nerve in the previous frame. To track the nerve, the merged CNNs output is used to compare the sliding windows in the search area. The tracking procedure consists in the prediction score of each sliding window in the searching area. From these scores, the final nerve position is assigned to the region with the highest score.

The merged CNNs based PF tracker starts by generating the particles where each particle will produce a candidate window. This followed by applying the merged 1D-2D CNNs model to each candidate window. The process will update the particles weight using the responses of the merged CNNs model. Finally, the predicted nerve position is obtained using,

$$P_n = \sum_{i=0}^{n} w_i c \tag{3.2}$$

where P_n is the predicted nerve location, n is the number of particles, w_i is the weight of the i^{th} particle calculated from the merged CNNs model, and c is the particle location in the image.

Updating the weights and particles is an important step in PF tracker which leads to avoid the degeneracy problem. A re-sampling algorithm is applied [171] where the weights are set to $w_i = 1/n$. Different resampling methods have been used to update the particles, such as multinomial, residual, stratified, and systematic methods. After extensive experiments, we used the residual method for updating the particles.

The merged CNNs based mix tracker is a combination of the previous two trackers. It starts by using the merged CNNs based PF tracker as explained before. If the highest score (the maximum similarity) of the particles' weight, retrieved from the merged CNNs model, is lower than a certain threshold, the merged CNNs based search tracker will be triggered for one frame to track the nerve more precisely.

For the three techniques, we adopt an incremental strategy for updating the merged CNN model with the new nerve appearances. The incremental strategy only uses new samples in the current frame to update the model. Hence, this updated model is able to adapt to the target appearance variations while alleviating the drift problem.

3.4 Experiments, results and discussion

This experiment provides a performance comparison and evaluation of the proposed approaches. Each method is analyzed in terms of accuracy, consistency, time cost, and handling different nerve situations. In this study, we conduct the experiment on the median nerve. As shown in Figure 3.6, the median nerve is one of the major nerves in the arm, it starts from the brachial plexus to innervate the intrinsic muscles of the hand. More precisely, the median nerve is located at the proximal elbow, mid-forearm, and wrist. The median nerve presents a circular, oval or elliptic shape [172].

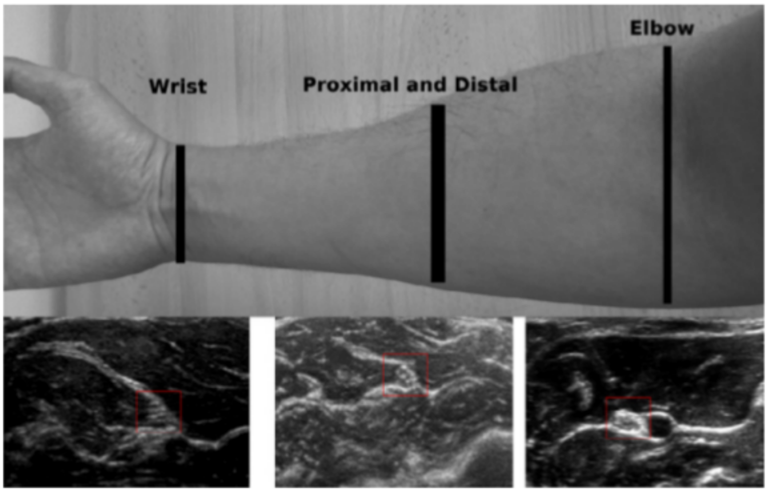


Figure 3.6: US images of the median nerve in the elbow, proximal and distal, and wrist forearm.

Next section describes the used dataset and setup. Then, the detection and tracking experiments are presented in next sections.

3.4.1 Dataset

In UGRA, the anesthetist starts by using the US probe to scan a part of the body back and forth in order to locate and track the nerve. This step is important to stabilize the probe in a good position to visualize the nerve and insert the needle. Nerve detection and tracking in US images require an in-depth study of the most effective detection and tracking techniques, as well as it relies on the large dataset to include intra and inter-patient variations. These steps are essential to assess the robustness of nerve detection and tracking methods in US images. However, the location of the nerve for UGRA is a very delicate subject, due to the lack of data. To our knowledge, no database of US images of the median nerves has been published. For this reason, data collection is inevitable to assess the performance of nerve detection and tracking methods.

For that, the dataset was acquired in real conditions at the Medipole Garonne hospital in Toulouse (France), and it is ethically approved. US videos of the median nerve were obtained from several volunteer patients under real conditions where the ground truth was provided by two regional anesthesia experts. Experiments were conducted on sonographic videos of the median nerve obtained from 20 anonymous adult patients (12 men and 8 women) using a US machine with a 5-12 MHZ transducer frequency. The databases are in the form of a video for each patient. Each video consists of 650-750 frames and a total number of 14,000 US images of the median nerve were used. The used images in our experiments have a dimension of 600 × 350 pixels. The experiments were carried out with a core 7 Duo 3.50GHz processor with 32GB RAM under Matlab and Python.

Very few studies have tackled the problem of nerve detection and tracking in the US images. Therefore, the results obtained by these methods were regularly presented to the anesthetists to collect their feedback and validate our results. The ground truth validated by anesthetists was compared with the results obtained by the computer. This comparison allows the measurement of the performance of the proposed methods. Furthermore, with the aim of performing in real-time, nerve detection and tracking performed directly on the original US image without any prior image enhancements.

3.4.2 Nerve detection experiments

The objective of this experiment is to evaluate each technique of the proposed detection models in order to study their role and impact on nerve detection. Figure 3.7 shows the scheme of the testing process. For testing an image, a sliding window of size of 64 × 64 is passed through the image with 50% overlapping to search the nerve.

As can be seen from Figure 3.7, the SVM and the 1D CNN models take the 1D RAMBP histogram as their input, the 2D CNN model takes the 2D window directly, while the merged CNN model takes both the 2D window and its 1D RAMBP histogram. The 1D RAMBP histogram is computed under Python and the implementations of the designed models are done under Python with the public deep learning libraries TensorFlow and Keras [173].

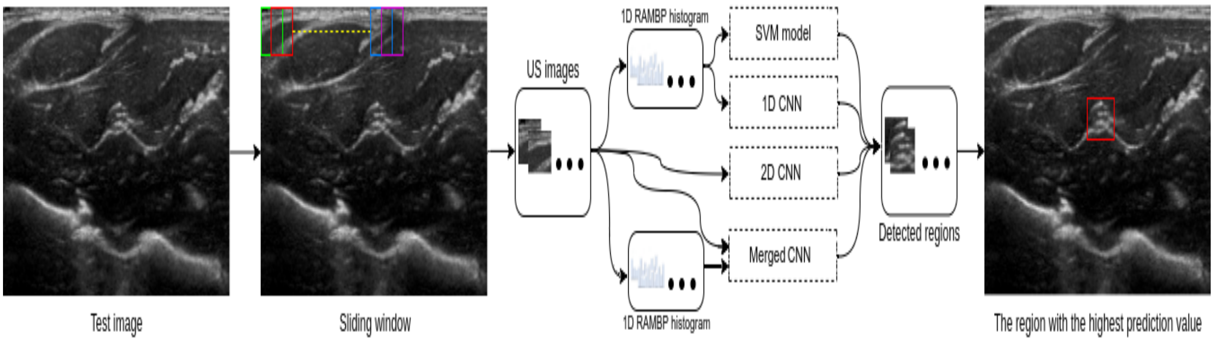


Figure 3.7: The scheme of the testing process.

In these experiments, 10 videos were selected randomly as a training dataset and the remaining 10 videos as a testing dataset. For setting the training dataset positive class, each labeled nerve is extracted with a bounding box around it. While for the negative class, a sliding window of size of 64×64 is passed randomly through the image except for the nerve regions.

For the training and testing process, the images are resized to 32×32 which will be sent to the CNN and SVM models. For all models and after trying different parameters, we choose a batch size of 256 and the number of epochs is 500. For CNN models and to find the output size of the first dense layer of 1D and 2D CNN, we choose 128 (dimensionality of the output space) after trying different values. As well as, the Adaptive Moment Estimation (Adam) [174] is adopted for the CNN learning procedure.

Figure 3.7 demonstrates the testing process to detect the nerve in US images. It started by exploring the testing images by passing a 50% overlapping sliding window which is the input for the 2D CNN model. RAMBP histogram is also computed for each window which is the input for the 1D CNN and SVM models. The output of the models provides all possible nerve locations, where the location with the highest prediction value is set as the detected nerve location. We consider the image as true positive if it overlapped with the exact nerve region (ground truth) at least 50%, otherwise, it considered as a negative sample (other regions).

Accuracy is the most common performance measure, but the main drawback is that accuracy hides some details that can help understanding better the detection model performance. For that, Recall, Precision, and F1-score provide better performance understanding by taking both false positives and false negatives into account. Precision is the correct proportion of positive identifications, Recall is the correct proportion of actual positives, and F1-score is the weighted average of precision and recall. Precision, Recall, and F1-score are expressed as,

$$Precision = \frac{TP}{TP + FP} \tag{3.3}$$

$$Recall = \frac{TP}{TP + FN} \tag{3.4}$$

$$F1 - score = \frac{2 \times (Precision \times Recall)}{Precision + Recall} \tag{3.5}$$

where TP, TN, FP, and FN are the number of samples for "True Positive", "True Negative", "False Positive" and "False Negative", respectively.

To show the effectiveness of the merged method, Table 3.1 depicts comparative results between the three CNN architectures (1D CNN, 2D CNN, and the merged CNN). It can be seen that the 1D CNN alone or the SVM model can detect nerve location with a precision of 82% and 81%, respectively. The performance of 2D CNN is better than that of 1D CNN and the SVM models with a precision of (85%). When compared with SVM and the two single-input CNNs, the merged CNN reaches the highest precision (96%) and shows a certain advantage in the overall performance. Furthermore, the merged CNN shows the importance of the RAMBP histogram which provides more accurate and robust results. Figure 3.8 shows the results of the merged CNN model for nerve detection.

Table 3.1: Comparison between different models and the merged method. The results are expressed in percentage between [0 1].

Method	**precision**	**recall**	**F1-score**
RAMBP 1D CNN	0.82	0.84	0.83
RAMBP SVM	0.81	0.82	0.82
2D CNN	0.85	0.87	0.86
The merged 1D-2D CNNs	**0.96**	**0.95**	**0.96**

In this work, we chose a simple architecture to demonstrate the effectiveness of the texture features for nerve detection in the US image. Furthermore, to show the robustness of the RAMBP descriptor with a more complex architecture, we have tested DenseNets [175] (2D - 121 layers) to detect the median nerve. After extensive experiments, as shown in Table 3.2, the precision of this network to detect the nerve did not pass 84%. At the same time, we implement a 1D architecture similar to the DenseNet but for 1D RAMBP input, it reaches a precision of 80%. Nevertheless, after doing a simple merge layer between the previous two networks, the precision reached 94%. We believe that the 2D CNN architecture is not enough to detect the nerve in US images due to its difficulties to capture textures information. Moreover, the use of a powerful texture descriptor such as RAMBP the accuracy would vastly be improved.

To show the robustness of the RAMBP descriptor against different texture descriptors, they have been used in the merged model as an input to the 1D CNN branch. Table 3.3 lists comparative results against different base-line descriptors such as LBP. It can be observed that the detection accuracy is improved after using the proposed model. Although AMBP precision

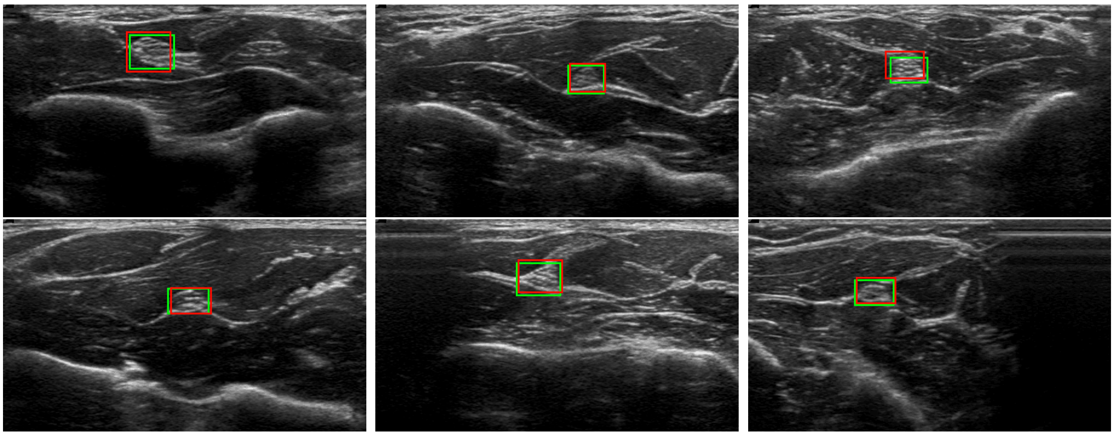


Figure 3.8: Nerve detection using the merged CNN model (red rectangle for detection results and green rectangle for the ground truth).

Table 3.2: Comparison between different CNN models and the merged method. The results are expressed in percentage between [0 1].

Method	**precision**	**recall**	**F1-score**
RAMBP 1D DenseNet	0.80	0.80	0.79
2D DenseNet	0.84	0.83	0.84
The merged 1D-2D DenseNet	**0.94**	**0.92**	**0.93**

is as good as RAMBP, the recall and F1-score results show the higher performance and stability of the RAMBP. Compared to the different base-line techniques, RAMBP yields the best results and outperforms other techniques.

The confusion matrix is a common way to evaluate the performance of a multiclass classification model, which plots the actual class label against the predicted class label. Figure 3.9 shows the corresponding confusion matrix of the merged model where about 5% of the nerve regions (label *nerve*) are wrongly classified. Likewise, for other regions (label *others*), less than 1% are wrongly assigned. From the confusion matrix, we found that the proposed system provides promising results for detecting the nerves in the US images.

Time complexity is an important point, especially in UGRA applications. While the tested models take a long time in the learning phase to build the models, it is fast in the detection phase. This has an advantage for real-time applications since the learning phase can be performed offline. Although the 1D CNN and the proposed merged model are considered slower than the 2D CNN and the SVM models due to the extra step of extracting the 1D texture descriptor from the sliding window, it is possible to make the detection faster by parallelizing RAMBP using, for example, a graphics processor GPU.

Table 3.3: A comparison between different descriptors as an input for the 1D CNN branch in the merged model. The results are expressed in percentage between [0 1].

Method	precision	recall	F1-score
LBP [80]	0.93	0.90	0.92
MBP [95]	0.94	0.92	0.93
CLBP [93]	0.94	0.91	0.93
HOG [176]	0.93	0.84	0.88
AMBP [98]	**0.96**	0.91	0.93
RAMBP [156]	**0.96**	**0.95**	**0.96**

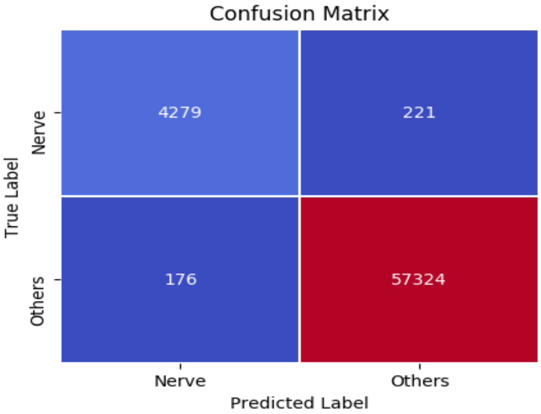


Figure 3.9: Confusion matrix for the merged model with RAMBP descriptor as an input for the 1D CNN branch.

3.4.3 Nerve tracking experiments

This section shows the feasibility of nerve tracking in US images using different approaches. These experiments provide a performance comparison and evaluation of the tracking approaches for nerve tracking in US images. The accuracy is assessed by the bounding box overlap ratio between the estimated nerve position and the ground truth. The overlap ratio is based on pixels percentage in the intersection area.

The structure of this section is as followed, first, we show the texture descriptors evaluations. Then, CNNs based trackers evaluations. Finally, we take the best methods in texture descriptors and CNNs based trackers and compare it with the merged model trackers.

3.4.3.1 Nerve tracking using texture descriptors

The objective of this experiment is to evaluate the nerve tracking techniques using texture descriptors. For setting up the trackers, we used for the particle filter 50 particles. In the mean-shift algorithm, we set the threshold to 0.0001. While the size of the descriptor in the Kanade-Lucas-Tomasi (KLT) algorithm is set to 16 × 16 pixels around the feature points.

In this validation part, the particle filter, mean-shift, and KLT algorithms are applied to US images. Figure 3.10 shows the accuracy of texture descriptors based trackers. We can observe that particle filter with RAMBP descriptor (PF-RAMBP) achieved the best results, while mean-shift and KLT algorithms had less stability and less performance accuracy. Also, it can be seen that AMBP, MBP, LBP, and CLBP gave good results with particle filter but with less stability. Figure 3.11 shows the results of the particle filter with RAMBP and the ground truth of the nerve location.

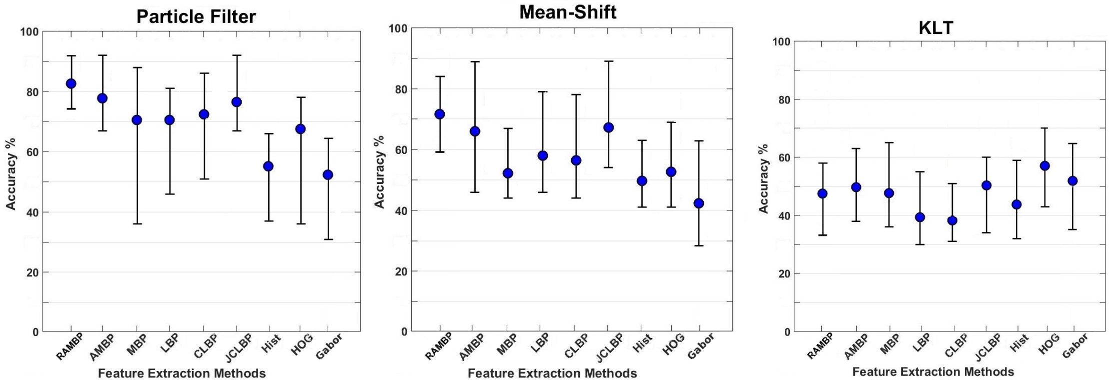
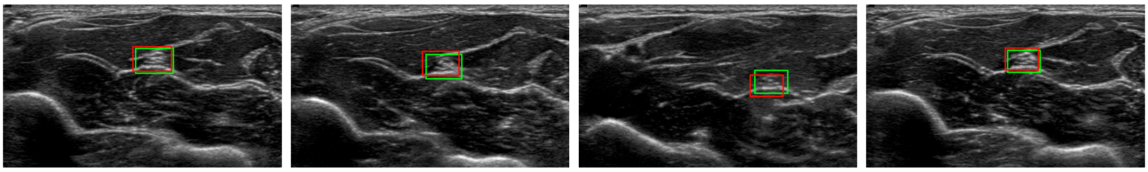


Figure 3.10: Accuracy of nerve tracking with bar error.



(a) $10^{th} frame$ (b) $200^{th} frame$ (c) $400^{th} frame$ (d) $600^{th} frame$

Figure 3.11: Nerve tracking using particle filter with RAMBP descriptor (red rectangle for particle tracking and green rectangle for the ground truth).

Table 3.4 shows the performance of the nine feature extraction methods. As can be seen, using a particle filter or mean-shift algorithm, RAMBP provides the best results. On the other hand, HOG, and AMBP perform better while applying KLT. These results are obtained thanks to RAMBP stability and discriminative properties. In general, the particle filter is more robust to noise, while the mean-shift and KLT do not perform well under the presence of noise.

The computational cost of particle filter depends on the number of particles, while the mean-shift on the threshold value, and for KLT on the number of features and for the size of the descriptor. Therefore in our experiments, particle filter and mean-shift algorithm require less computation time.Table 3.5 shows the running time for each method, where it can be seen that the particle filter algorithm (best tracking method) provides good processing time with Matlab, which can be optimized to be much faster using real-time platforms. In the same table, it can be seen that RAMBP is slower than LBP, AMBP, MBP, and normal histogram. LBP uses the central pixel as a threshold value, MBP uses the median of a fixed

Table 3.4: Tracking scores (%) comparison between RAMBP, AMBP, MBP, LBP, CLBP, Gabor, HOG, and normal histogram with the proposed tracking algorithms. The results are expressed in percentage between [0 1].

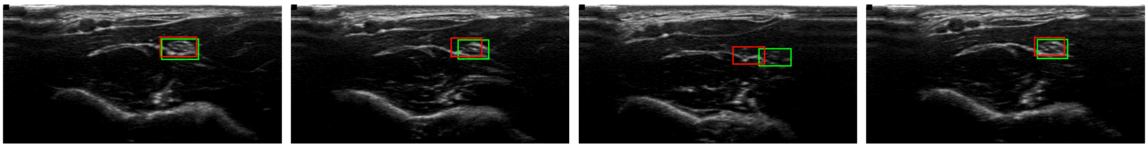
Method	Particle filter	Mean-shift	KLT
RAMBP [156]	**0.83**	**0.72**	0.47
AMBP [98]	0.78	0.66	0.49
MBP [95]	0.71	0.52	0.47
LBP [80]	0.70	0.58	0.39
CLBP [93]	0.72	0.57	0.38
HOG [155]	0.67	0.53	**0.57**
Gabor [177]	0.51	0.43	0.53
Hist	0.55	0.50	0.43

window, AMBP uses extra procedure which consists of using an adaptive window to obtain the threshold value, while RAMBP uses an extra process of classifying each pixel in the image, as well as an adaptive window. For that, RAMBP is slower than LBP, MBP, and AMBP, but at the same time it provides a good trade-off between tracking accuracy and time complexity.

Table 3.5: Tracking Speed (Second Per Frame spf) between the proposed tracking methods.

	Particle filter	Mean-shift	KLT
RAMBP [156]	0.15	0.30	0.47
AMBP [98]	0.12	0.24	0.38
MBP [95]	0.11	0.15	0.40
LBP [80]	0.08	0.08	0.09
CLBP [93]	0.18	0.27	0.22
HOG [155]	0.17	0.32	0.42
Gabor [177]	0.20	0.45	0.52
Hist	0.05	0.03	0.05

The experiments faced some failures in a few frames where the nerve appearance is almost the same as the surrounding areas, but these cases are rarely encountered in the tests. Besides, the proposed method faces some challenges such as real-time process, one of the future works consists of using optimization methods and GPU's to ensure and guarantee the reduction in time complexity. Another challenge is how to reduce the effect of maximizing the error after a high number of iteration, and one of the possible solutions is by using detection step after a fixed number of frames to correct the initial location of the nerve. Figure 3.12 shows an example of a few frames of failure after using RAMBP with a particle filter, which happened due to nerve disappearance. In the same figure, it can be seen that the system succeeded to re-track the nerve after a few frames of losing the target.



(a) $470^{th} frame$ (b) $490^{th} frame$ (c) $510^{th} frame$ (d) $530^{th} frame$

Figure 3.12: RAMBP Nerve tracking with particle filter. The drastic change in nerve appearance may lead to detection failure, but the system re-tracked then nerve successfully.

3.4.3.2 Nerve tracking using CNNs models

In this section, we evaluate CNNs based trackers for nerve tracking in the US image. The same parameters provided by the original papers were used along with the source codes that have been made available by the original authors. In this experiment, VGG-Net [178], very deep convolutional networks (up to 19 layers) are adopted for feature extraction.

Figure 3.13 illustrates the tracking methods accuracy for the median nerve, where ECO, C-COT, and SANet achieved the best results, while other methods suffer from less stability and less performance accuracy. Also, it can be seen that CREST gave good results but with less stability. Figure 3.14 shows qualitative results of tracking the median nerve using the ECO method.

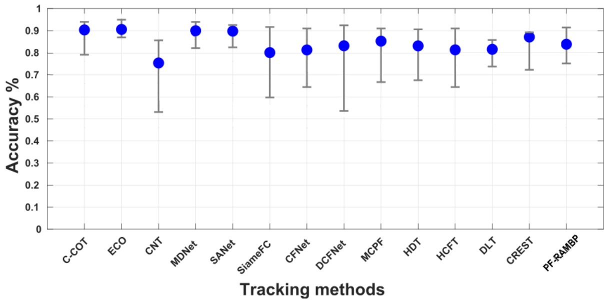
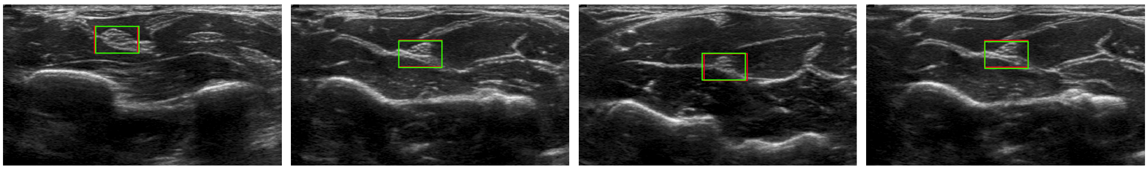


Figure 3.13: The performance of deep-learning trackers for the median nerve, where accuracy and stability are shown.



$10^{th} frame$ $200^{th} frame$ $400^{th} frame$ $600^{th} frame$

Figure 3.14: Nerve tracking using ECO tracker (red rectangle for ECO method and a green rectangle for the ground-truth).

Table 3.6 depicts tracking methods performance for median nerves. Using ECO provides the best results, where these results are obtained due to transferring prior visual via pre-training and capturing any appearance changes via online learning. C-COT adopts the same maneuver as ECO, but ECO provides a better generalization of the target by avoiding over-fitting. Other good trackers are SANet and MDNet which achieved a good score caused by using a particle filter framework in its design. As well as this, SANet incorporates an RNN scheme which leads to an increase in the tracking accuracy.

Table 3.6: Tracking scores (%) comparison between the proposed tracking methods. The results are expressed in percentage between [0 1].

$C-COT$ [158]	0.90
ECO [159]	**0.91**
CNT [160]	0.77
$MDNet$ [161]	0.88
$SANet$ [162]	0.89
$SiameFC$ [163]	0.80
$CFNet$ [164]	0.81
$DCFNet$ [165]	0.83
$MCPF$ [166]	0.86
HDT [167]	0.83
$HCFT$ [168]	0.82
DLT [141]	0.82
$CREST$ [169]	0.87
$PF-RAMBP$	0.83

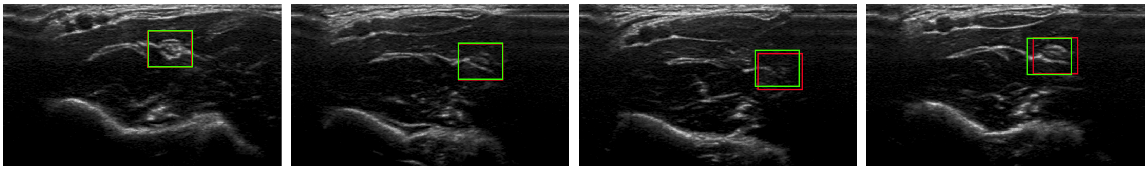
CNT tracker uses one convolutional layer, while others use deeper convolutional layers such as ECO and HCFT. In this experiment, it was observed that using more deep layers results in better performance and improves the tracking accuracy. Comparing between CNN-based deep trackers and traditional texture based trackers such as Particle Filter (PF) with RAMBP features [156], PF-RAMBP achieves good results and outperforms few deep-learning trackers. Finally, it can be observed that the ECO tracker provides the best results among CNN-based deep trackers for nerve tracking and gives the best stable results. Overall, the accuracy of CNN-based deep trackers is competitive and provides good performance for tracking the median nerve.

Table 3.7 demonstrates the running time for each method where it shows that DCFNet provides the best processing time. While ECO is slow, C-COT, SANet, and HDT are much slower. Important aspects affect the running time for CNN-based deep tracking algorithms, which are the number of layers and model update strategy. Some trackers use more deep layers while others use fewer layers which makes the tracker run faster. The other important aspect is the tracker model update strategy, where it can be noticed that updating the model after each frame is time-consuming. For that, ECO updates its model every few frames which makes the process run faster. Another strategy to update the model is using the Siamese network to model prior information that accelerates the running process such as CFNet, DCFNet, and SiameseFCs. While ECO is not the fastest method but, at the same time, it provides a good trade-off between tracking accuracy and time complexity.

Table 3.7: Tracking Speed (spf) between the proposed tracking methods.

$C-COT$ [158]	0.65
ECO [159]	0.13
CNT [160]	0.63
$MDNet$ [161]	1.00
$SANet$ [162]	1.63
$SiameFC$ [163]	0.26
$CFNet$ [164]	0.45
$DCFNet$ [165]	0.04
$MCPF$ [166]	0.60
HDT [167]	0.89
$HCFT$ [168]	0.27
DLT [141]	0.33
$CREST$ [169]	0.9
$PF-RAMBP$	0.15

The experiments faced some challenges when the nerve disappeared or appeared to be almost as identical as the surrounding areas. The MCPF tracker uses a particle filter principle which gives it the ability to re-track the nerve in case of disappearance. Other trackers expand their localization to re-track the nerve once it appears again such as ECO and DCFNet. On the other hand, the CREST tracker failed to re-track the nerve rapidly after it appeared again. Figure 3.15 shows an example of how MCPF succeeds in tracking the nerve even when the nerve almost disappeared.



(a) $470^{th} frame$ (b) $490^{th} frame$ (c) $510^{th} frame$ (d) $530^{th} frame$

Figure 3.15: Nerve tracking using MCPF tracker. Although the existence of nerve disappearance, the tracker succeeded to predict the nerve location (red rectangle for MCPF tracker and a green rectangle for the ground-truth).

3.4.3.3 Nerve tracking using merged features

The objective of this section is to evaluate the nerve trackers using the merged CNNs model in order to study their role and impact. Three tracker were proposed based on the 1D-2D merged CNNs model: the merged CNNs based search tracker; the merged CNNs based PF tracker; and the merged CNNs based mix tracker. These techniques are evaluated against the best-performed methods in texture based trackers (PF-RAMBP) and the CNNs based trackers (ECO tracker).

For the merged model, the images are resized to 32 × 32 which will be inputted to the CNN branch. After trying different parameters, we choose a batch size of 256 and the number of epochs is 500. To find the output size of the first dense layer, we choose 128 (dimensionality of the output space) after trying different values. As well as, the Adaptive Moment Estimation (Adam) is adopted for the CNN learning procedure.

For the merged CNNs based search tracker, a window which is 4 times bigger than the previous nerve location is initialized, followed by a sliding window of size of 64 × 64 is passed through the searched window with 50% overlapping to search the nerve. For the merged CNNs based PF tracker, we set the number of particles to 30. While for the merged CNNs based mix tracker, we set the threshold to 60% which means if the similarity between the previous nerve location and PF candidates is less than 60%, the merged CNNs based search tracker will be activated to track the nerve for one frame.

Figure 3.16 illustrates the tracking methods' accuracy for the median nerve, where the merged CNNs based search tracker achieved the best results and stability. While PF-RAMBP has good accuracy, the PF with the merged CNNs model has a better performance. As can be observed, the merged CNNs based trackers outperform the CNN and texture descriptors based trackers. Moreover, combining between the PF and the search trackers provides good performance with high stability. Figure 3.17 shows qualitative results of tracking the median nerve using the merged CNNs based search tracker.

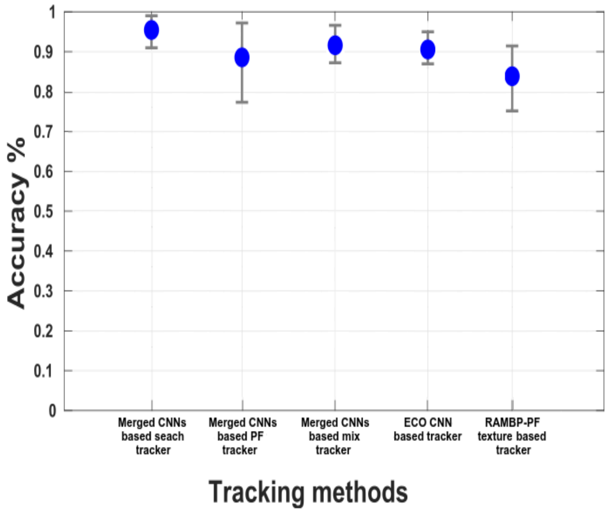


Figure 3.16: The performance of nerve trackers for the median nerve, where accuracy and stability are shown.

Table 3.8 depicts the tracking methods performance for median nerves. Using the merged CNNs based search tracker provides the best results, due to the fact that this tracker work as a detection method in the nerve surrounding region. the merged CNNs based mix tracker has good results and outperforms the merged CNNs based PF tracker. Comparing between merged CNNs-based trackers, CNN trackers, and traditional texture descriptor trackers, the merged CNNs-based trackers achieve the best results.

As discussed in previous experiments, the time complexity is considered an important point for a more successful UGRA. Table 3.9 depicts the running time for each method where it shows

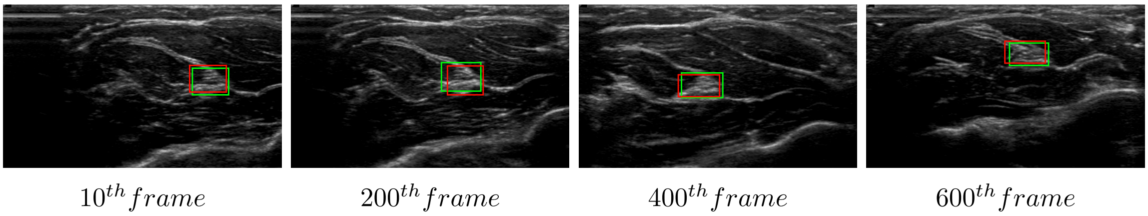


Figure 3.17: The merged CNNs based search tracker for median nerve tracking. (red rectangle for the merged CNNs based search tracker and a green rectangle for the ground-truth)

Table 3.8: Tracking scores (%) comparison between the proposed tracking methods. The results are expressed in percentage between [0 1].

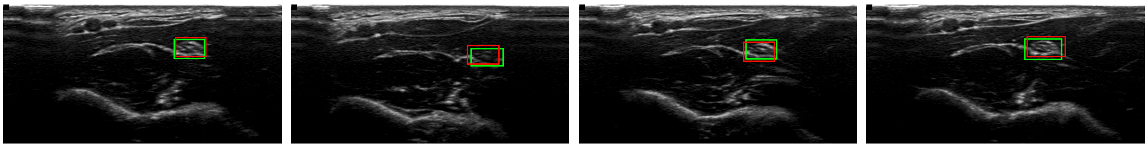
The merged CNNs based search tracker	**0.96**
The merged CNNs based PF tracker	0.88
The merged CNNs based mix tracker	0.93
ECO (CNN-based tracker) [159]	0.91
PF-RAMBP (texture descriptor-based tracker)	0.83

that PF provides the best processing time. Important aspects affect the running time for the tracking methods. For The merged CNNs based search tracker, it suffers under time cost due to the search process around the nerve previous location. For PF, the time cost is affected by the number of particles. While the merged CNNs based mix tracker is not the fastest method but, at the same time, it provides a good trade-off between tracking accuracy and time complexity.

Table 3.9: Tracking Speed (spf) between the proposed tracking methods.

The merged CNNs based search tracker	0.25
The merged CNNs based PF tracker	0.10
The merged CNNs based mix tracker	0.19
ECO (CNN-based tracker) [159]	0.13
PF-RAMBP (texture descriptor-based tracker)	0.15

The challenge of nerve disappearance is crucial for successful nerve tracking. Since the nerve region appears similar to the other surrounding regions, the succession or the failure to track the nerve can distinguish a robust tracker from another. The PF principle has the ability to track the nerve but it depends on the number of particles that affect the time cost. ECO method expands the search area once the target nerve is disappeared. On the other hand, the merged CNNs search mix tracker guarantees nerve tracking but at the cost of time. The merged CNNs based mix tracker provides a good trade-off since it has the fast PF principles and the accurate search tracker. Fig. 3.18 shows an example of how the merged CNNs based mix tracker succeeds in tracking the nerve even at nerve disappearance.



(a) $470^{th} frame$ (b) $490^{th} frame$ (c) $510^{th} frame$ (d) $530^{th} frame$

Figure 3.18: Nerve tracking using the merged CNNs based mix tracker. Even with the drastic change in nerve appearance, this tracker succeeded to track the nerve (red rectangle for the merged CNNs based mix tracker and a green rectangle for the ground-truth).

3.5 Conclusion

To conclude, nerve detection and tracking is an important step to stabilize the ultrasound probe in a good position to visualize the nerve and insert the needle. Although there has been extensive development of detection, and tracking algorithms for medical images, it is still an open problem especially for ultrasound images in regional anesthesia. In this chapter, we have examined and proposed techniques based on RAMBP and CNNs models for detection and tracking in the ultrasound images. Furthermore, we presented a new merged CNN architecture to learn high-level features from hand crafted 1D descriptor RAMBP and 2D low level visual features.

In this chapter, first, we introduced the detection and tracking techniques in detail. This followed by evaluating each technique and discussing its advantages and limitations. Extensive experiments were held in terms of accuracy, consistency, time complexity, and handling different nerve situations.

It has been shown that nerve detection and tracking techniques that are base on the merged model outperformed other techniques. Although the texture based and CNNs-based detectors and trackers perform good in the ultrasound images, the results show that the merged model obtained higher accuracy in the experiments. Furthermore, different techniques were presented for detecing and tracking the nerve in the ultrasound image. The obtained detection results show that the merged features based detector outperforms other techniques and achieved best performance with 96% precision. On the other hand and for the tracking techniques, we found that the merged CNNs based trackers acheived the best results where we beleive that the mix tracker (particle filter and search technique) provides a good trade-off between tracking accuracy (93%) and time complexity.

Chapter 4

Robotic Control

Outline:
In Ultrasound-Guided Regional Anesthesia, needle insertion imposes a very challenging task, where it is important to ensure that the needle moves within the constraints of the insertion point and obstacle avoidance in order to prevent patient harm and reach the targeted nerve. In this chapter, a robotized needle insertion control system is introduced. This robotic system can provide great assistance to the experts with techniques and tools that improve the procedure accuracy and safety such as avoiding nerve trauma or damage healthy tissues.

Contents

4.1 Tasks of robotic-assisted UGRA

As discussed in Chapter 1, performing the UGRA routine requires a long learning process, mainly due to the risk of nerve trauma while inserting the needle and due to the difficulty of processing the poor quality US image [5, 6]. Therefore, the DANIEAL2 project aims to provide anesthetists with a collaborative expert-robot-environment platform that significantly improves the UGRA practice; that is to study and develop an intrinsically safe robotic medical device, where the expert is always in the control loop. This involves assisting the medical gesture when inserting the needle towards the targeted nerve under ultrasound (US) imaging using the collaboration of a robotic arm.

The proposed robotic system designed for the UGRA procedure is divided into two connected robotic systems, a US Probe Holder Robot (PHR) and a Needle Holder Robot (NHR), as shown in Figure 4.1.

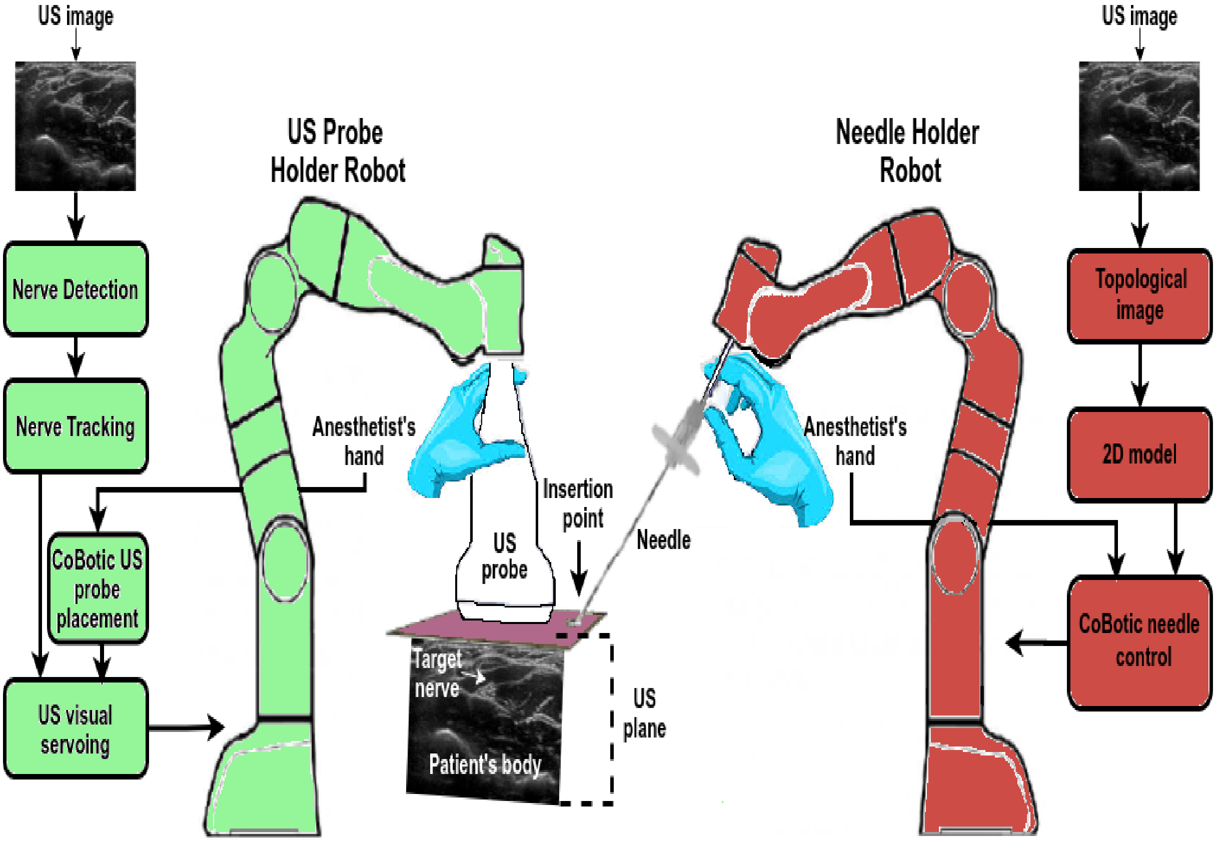


Figure 4.1: The proposed robotic-assisted UGRA procedure.

The proposed robotic system is divided into 3 **sequential phases** where each one represents an action of the UGRA act related to the NHR and the PHR. These phases shown in Figure 4.2 start with the preliminary CoBotic US probe positioning defined as φ_0. In phase φ_1: the nerve is detected and visualized at all time by visual servoing approach applied to the US probe; then the needle is positioned on the patient's skin at the so-called the insertion point and CoBotically aligned within the US plane. Finally, in phase φ_2: the nerve and the needle are visualized at all time in the same US plane by applying visual servoing; the needle is then CoBotically inserted towards the targeted nerve through the insertion point. The anesthetist manually injects the drug once the needle's tip is close to the nerve area. These phases are explained in details as follows:

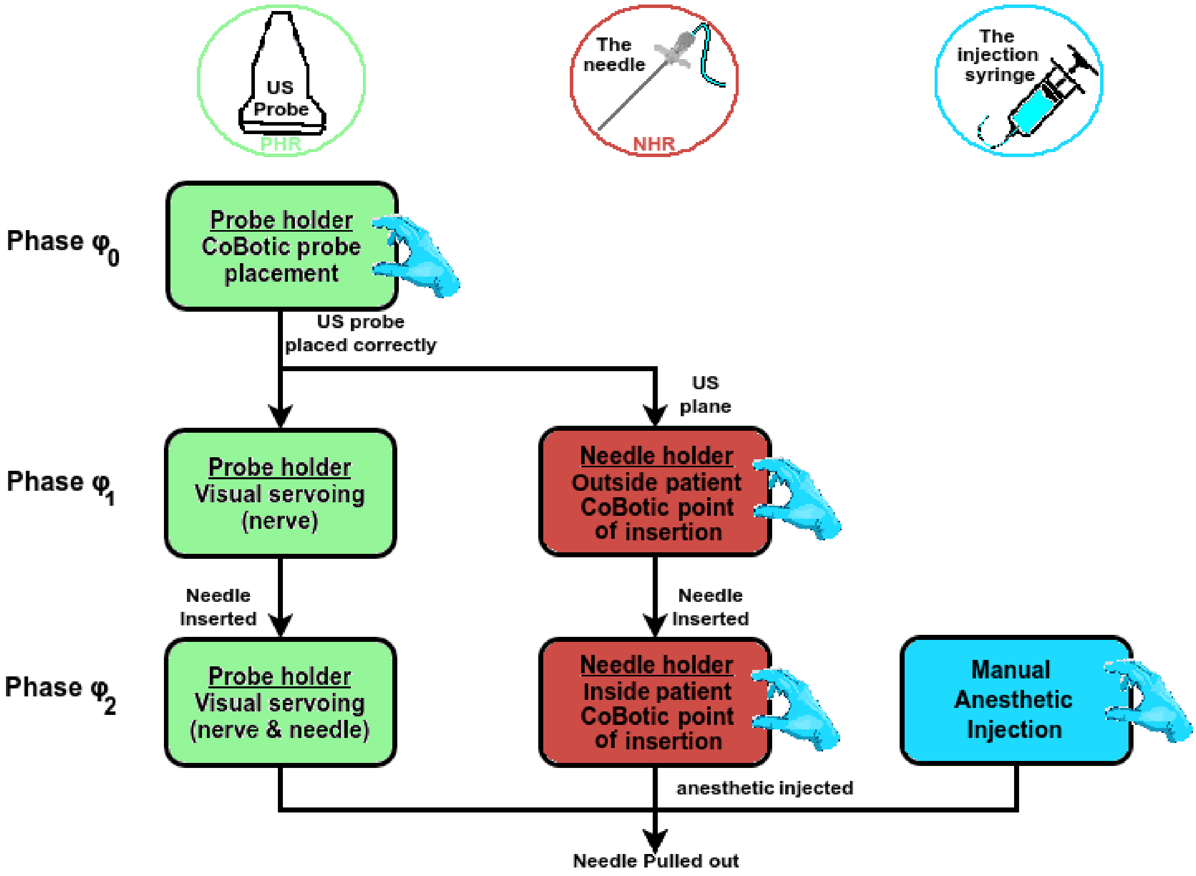


Figure 4.2: The proposed robotic-assisted UGRA procedure.

- Phase φ_0 - PHR CoBotic US probe placement: this phase starts by setting up the patient, followed by a CoBotically placement of the US probe by the anesthetist on the patient's skin.

  When the anesthetist has positioned the probe at the chosen location, then Phase φ_1 is triggered.

- Phase φ_1 - this phase is composed of 2 **simultaneous parts**, one for the PHR and the other for the NHR:
  - PHR part: controls the PHR by visual servoing to unmanned maintain the desired regions of interest (e.g. the nerve) in the 2D US image obtained during phase φ_0. Then, automatic nerve detection and tracking, and a visual servoing technique are

performed to keep the targeted nerve within the 2D US plane even when physiological movements occur.

- NHR part: when the needle is outside the patient's body (so called "Outside patient control" process), the anesthetist's goal is to find a needle insertion point on the patient's skin that could provide the best trajectory in order to reach the nerve. But the expert faces two constraints: the insertion point must be in the US plane (note that the needle cannot appear on the US image until it is inside the patient's body); the needle pose is constrained by the need of being seen within the US plane. Knowing the geometrical transformations of the PHR and NHR, we can offer the anesthetist a CoBotic control of the NHR that respect the two constraints with an adapted force feedback over the 6-Dimensions. Finally, the NHR CoBotically ensures needle visualization within the US plane.

Once the anesthetist inserts the needle in the well-chosen location (i.e. insertion point) on the patient's body, phase φ_2 is triggered.

- Phase φ_2: this phase is composed of 3 **simultaneous parts**, one for the PHR, one for the NHR, and one for the manual injection:
  - PHR part: the aim is to unmanned keep the targeted nerve and the inserted needle within the 2D US image plane using US visual servoing approach. To complete this action, we need to track the nerve in the US images and to retrieve the position/orientation of the needle's tip using the geometrical transformations between the two robots. Hence, when the anesthetist tilts the needle outside the US plane, the PHR has to follow the needle's tip and keep it within the US plane.
  - NHR part: when the needle is inserted in the patient's body (so called "Inside patient control" process), the anesthetist's goal is to move the needle as close as possible to the nerve with respect to Remote Center of Motion (RCM) constraints. The NHR is CoBotically controlled to manipulate the needle by reducing the DoF around the RCM point. In order for the anesthetist to avoid risky regions such as artery, vein, or tendon, a topological image, based on the US image, is processed on-line to build a 2D model (mesh) that provides interaction among various tissues and the needle's tip and shaft. This model allows the NHR to apply repulsive forces once the needle's is near risky regions. Moreover, with the 2D model, we can propose to the anesthetist safe trajectories to reach the nerve area.
  - Manual injection part: This part is manually performed by the anesthetist with no robotic assistance.

  Once the anesthetic drug is injected around the nerve, the anesthetist pulls out the needle; once the needle is outside the patient's body, the NHR and PHR assistance process stop.

The folllowing scheme details the various **tasks** that compose the PHR and NHR control phases.

Phase φ_0

PHR part

- Ⓐ PHR CoBotic control for the probe placement on the patient's body.

Phase φ_1

PHR part

- Ⓑ Nerve detection in one US image.
- Ⓒ Nerve tracking in the streamed US images.
- Ⓓ Visual servoing to control the PHR to keep the visualization of the nerve.

NHR part

- Ⓔ NHR CoBotic control for needle placement in the US plane.

Phase φ_2

PHR part

- Ⓑ Nerve detection in one US image.
- Ⓒ Nerve tracking in the streamed US images.
- (D') Visual servoing to control the PHR to keep the nerve AND the needle visualization in the US image.

NHR part

- Ⓕ NHR CoBotic control for needle insertion under RCM constraints.
- Ⓖ Topological image from US image.
- Ⓗ 2D model.
- Ⓘ NHR CoBotic control for needle insertion under RCM constraints with risky regions avoidance.

Manual injection part

- Ⓚ No robotic assistance needed.

For Phase φ_0 task Ⓐ, the anesthetist control the NHR CoBotically, this control does not require any movement constraints. The output of this task is the US image and the US plane which are inputted to Phase φ_1, in more specific into tasks Ⓑ and Ⓔ, respectively.

Task Ⓑ uses the US image to automatically detect the target nerve, as discussed in Chapter 3. Task Ⓒ takes the output of task Ⓑ (the detected nerve location) and starts tracking the nerve in real-time streamed US images, as discussed in Chapter 3. The result of task Ⓒ is the tracked nerve in each US frame which is inputted into the task Ⓓ. Ⓓ aimed to keep the nerve visualization at all time by taking the tracked nerved position and applying visual servoing techniques.

At the same time, task Ⓔ NHR CoBotic control for needle placement in the US plane is performed. In this task, the anesthetist moves the NHR CoBotically toward the desired point of insertion location. To guarantee the visualization of the needle within the US plane, the needle should be CoBotically aligned within the US plane. To do so, the PHR has to share its workspace with the NHR. Hence, the positions of the US probe and plane will be referenced to the NHR base frame. Therefore, forces are applied to the NHR to guarantee the proper placement of the needle within the US plane.

At this stage, the targeted nerve is visualized within the US image, and the needle is positioned on the patient's skin in a position that guarantees inserting the needle within the US plane (visualized in the US image). Then, Phase φ_2 will be triggered.

In phase φ_2, task Ⓑ and Ⓒ are repeated, where the output nerve position is entered in task Ⓓ′. Task Ⓓ′ uses a visual servoing technique to keep visualization of the tracked nerve and the needle within the US plane. To keep visualizing the needle in the US plane, the NHR has to share its workspace with the PHR. Therefore, the position of the needle plane will be referenced to the PHR base frame.

Simultaneously, task Ⓕ NHR CoBotic control for needle insertion under RCM constraints is performed. The anesthetist CoBotically controls the needle insertion while taking into account the constraints of the insertion point. However, using task Ⓕ alone is not sufficient as it lacks the avoidance process of risky regions such as arteries or bones. Hence, tasks Ⓖ and Ⓗ aim to locate all regions in a given US image. In task Ⓘ, the NHR is CoBotically controlled with the addition of applying small forces on the robot end-effector which are rendered to the anesthetist's hand once the needle's tip is close to a risky region such as an artery, a bone, and even the nerve.

While the PHR control keeps visualizing the target nerve and the inserted needle, and once the needle's tip is close to the nerve, the anesthetist injects the anesthetic manually to complete the UGRA procedure in task Ⓚ.

As this thesis is a part of DANIEAL2 project, we do not focus on proposing new visual servoing techniques (task Ⓓ and Ⓓ′), wherein this project proposes to use some existing visual servoing methods for US medical applications. In this thesis, we focus on proposing new methods and techniques for nerve detection and tracking (tasks Ⓑ and Ⓒ) in Chapter 3, and in this chapter, we focus on proposing methods for phase φ_2 NHR part (tasks Ⓕ, Ⓖ, Ⓗ, and Ⓘ) where the next section will introduce the proposed work.

4.2 The needle holder robot control in phase φ_2 (NHR part)

In this section, we present methods for tasks Ⓕ, Ⓖ, Ⓗ, Ⓘ of part 2 NHR of phase φ_2. These tasks focus on controlling the needle inside the patient's body. This control aims to fix the interaction point between the patient's skin and the needle where the needle movement will be around that point. The interaction pivot point between the needle and the skin of the patient is named the insertion point. This point restricts the medical instrument movements inside the patient's body. These movements may different from one type of instruments to another. For example, the trocar is a well known instrument hugly used during laparoscopic surgery. Due to trocar's characteristics, movements around the insertion point are restricted into only four degrees of freedom (DoF) motion: three rotational DoF and one translational DoF. For UGRA procedure where a rigid symetric needle is used, movements around the insertion point are restricted into only three DoF motion: two rotational DoF and one translational DoF. Therefore in this work, the rotation on z-axis is blocked due to the symetrical aspect of the needle.

In UGRA, the insertion point constitutes an essential issue constraining the anesthetist gesture when manipulating a needle inside the human body. However, manipulating the needle imposes a very challenging task, where it is important to ensure that the needle should move within the constraints of the insertion point in order to prevent harm to the patient. While using a robotic-assisted procedure, the inserted needle movements are limited to a translation along its axis and rotations around at the insertion point [179]. The manipulator motion is then constrained with respect to that point on the end-effector known as Remote Center of Motion (RCM) as shown in Figure 4.3 [52, 180].

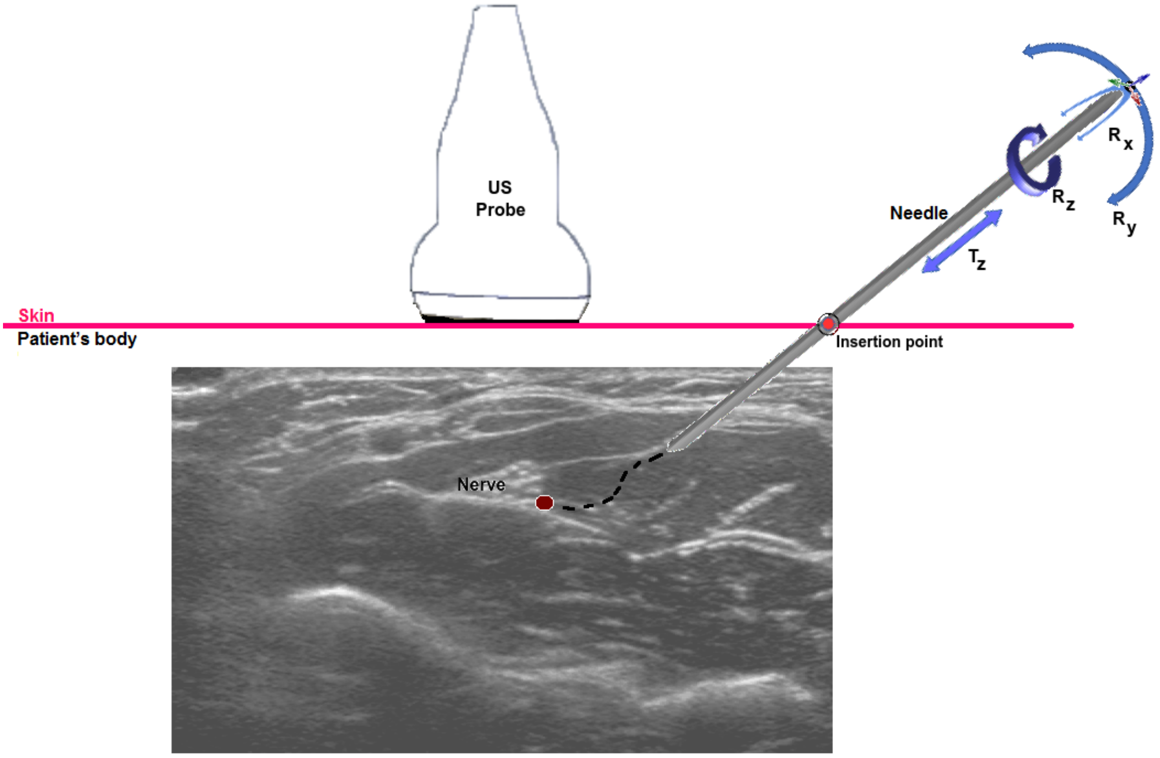


Figure 4.3: UGRA motion around the insertion point (Pinsertion - RCM), where the tool frame is the end-effector position and rotation. As a result, the needle can only move within 4-DoF: R_x, R_y, and R_z are the rotations around x, y and z-axis respectively, and T_z is the translation along the z-axis.

To respond to this need, this chapter proposes a control framework for robot-assisted UGRA for physical human-robot collaboration using a 7-DoF robot manipulator (Franka Emika [62]).

Herein, we discuss a geometric method for computing the intended robot's end-effector position with respect to the RCM constraints, it is intended to assist the anesthetist to execute a more sophisticated motion of the needle within the patient's body with high accuracy. This accuracy is set as the distance error between the RCM point position and the fixed insertion point position. After that, we provide two robot control strategies with respect to the RCM constraints; haptic and CoBotic needle insertion controls. The basic action of these needle insertion control strategies is to move the needle through the insertion point and to generate a remote center of motion (RCM) remotely or CoBotically, while avoiding risky regions.

Task Ⓕ is presented by proposing the needle insertion kinematics in Section 4.3 (RCM for needle insertion) and introducing various control strategies in Section 4.4 (haptic force feedback and CoBotic control). This followed by presenting the process CoBotic control under RCM constraints with regions avoidance in Section 4.5 where tasks Ⓖ, Ⓗ, and Ⓘ are proposed. Experimental setup, results, and discussion in Section 4.6. Finally, conclusions are given in Section 4.7.

4.3 Needle insertion kinematic

In theory, a high number of DoF serial robot provides a large operation workspace and dexterity. However, keeping the end-effector working through an RCM point and limiting its motions is a very challenging task which demands an accurate control strategy [181, 182]. In the literature, various methods have been proposed to address RCM constraints such as Jacobian with constrained quadratic optimization [183], isotropy-based kinematic optimization [184], constrained Jacobian with Lie algebra [185], gradient projection approach [186], dual quaternion-based kinematic [187], etc. The previous researches discussed the modeling of trocar kinematics only or combined with trajectory following. Very few control strategies reported in the literature are concerned with the issue of RCM constraints and their geometry in the workspace. Mayer et al. [188] proposed a trocar modeling with Euler angles representation for heart surgery. In [189], the authors proposed using geometric constraints with stereo visual servoing for controlling the robot position from point-to-point and extended Jacobian solution for manipulating serial end-effector.

The main contribution of this section is formulating a geometric control method in the operational space to maintain RCM constraints while offering a more accurate end-effector position. The method controls the motion of the rigid needle and it describes the end-effector position with respect to the RCM constraints for a robotic UGRA control using a 7-DoF serial robot. It provides a safe and flexible solution for the robot and the medical staff in a shared workspace where it explicitly models the rotational and translational motions at the RCM point.

In UGRA, the needle insertion process relies on only 3-DoF for its orientations (i.e., pan, tilt, and spin rotations) corresponding to the roll, pitch, and yaw angles around the *RCM* point; 1-DoF is needed for the axial translation corresponding to the depth of insertion as shown in Figure 4.3. Hence, to simplify the control strategy for the robot and to help the kinematic design, we choose to define the movements of the needle with respect to these four displacements. Various strategies exist to control the needle movements inside the patient, where the most comfortable way would be the operational space using the Cartesian representation.

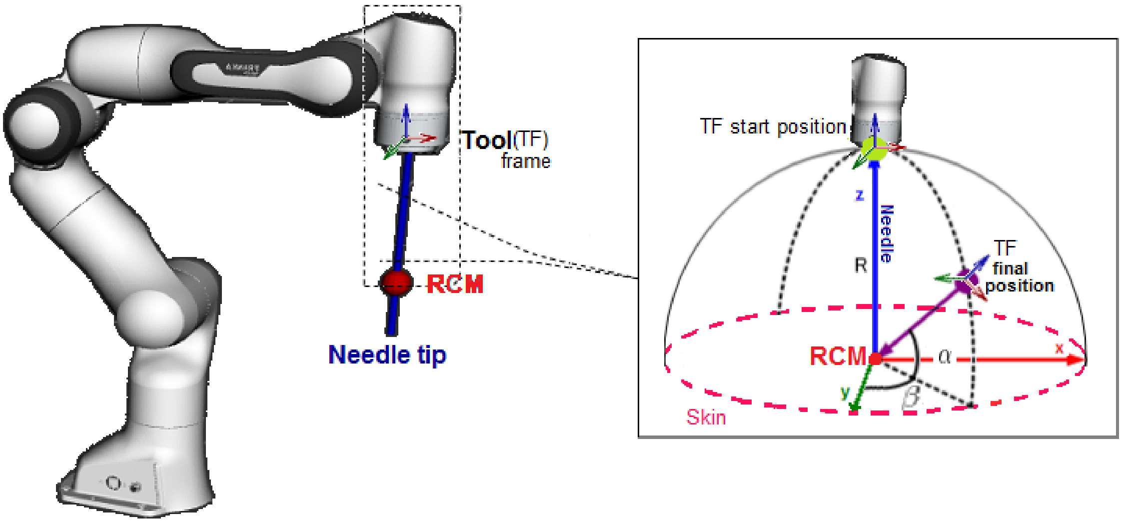


Figure 4.4: A Spherical representation of end-effector position. Where α and β are the sphere representation angles and R is the radius of the sphere.

The principal objective of this section is to propose an explicit and yet generic expression of the intended tool frame position as shown in Figure 4.3. Hence, the robot command and movements are controlled using the tool frame position. The tool frame control requires guaranteeing the RCM constraints. Therefore, we choose to perform a tool frame motion along a sphere centered at RCM point as shown in Figure 4.4. Thus, the tool frame is moved from a "start" position on the sphere to a "final" position on the same sphere with $2D$ tangential constraints. The start position of the tool frame is found using the direct kinematics model computed from the robot built-in library. This position is composed of a 3D vector of translations and a 3×3 rotation matrix with respect to the base frame.

Let us define the starting position of the end-effector as $3D$ translations ($P_{ST} = [P_{STx} \; P_{STy} \; P_{STz}]^T$) and 3×3 rotation matrix (P_{SR}). To find the RCM position (sphere center with a radius R as shown in Figure 4.4), first, we have to retrieve from the rotation matrix the three rotation of the end-effector (Yaw ψ, Pitch θ, and Roll ϕ):

$$\psi = atan2(P_{SR32}, P_{S33}) \tag{4.1}$$

$$\theta = atan2(P_{S31}, \sqrt{P_{SR32}^2 + P_{SR33}^2}) \tag{4.2}$$

$$\phi = atan2(P_{SR21}, P_{SR11}) \tag{4.3}$$

Using these rotations, the RCM position ($[RCM_x \; RCM_y \; RCM_z]^T$) is found by:

$$RCM_x = P_{STx} + (cos(\phi)sin(\theta)cos(\psi) + sin(\phi)sin(\psi))R \tag{4.4}$$

$$RCM_y = P_{STy} + (sin(\phi)sin(\theta)cos(\psi) - cos(\phi)sin(\psi))R \tag{4.5}$$

$$RCM_z = P_{STz} - cos(\theta)cos(\psi)R \tag{4.6}$$

As shown in Figure 4.3 and Figure 4.4, the tool frame (end-effector) movements from one position to another (such as start position to final position in Figure 4.4) depend on the applied rotations on the end-effector (R_x and R_y). Hence for finding the final position (translation), the two spherical rotations are:

$$\alpha = \frac{\pi}{2} - R_x \tag{4.7}$$

$$\beta = \frac{\pi}{2} - R_y \tag{4.8}$$

We now have to consider how to compute the desired final position. Thanks to the spherical representation and having the RCM position with the two rotation angles, the final position translation coordinates ($P_{FT} = [P_{FTx} \quad P_{FTy} \quad P_{FTz}]^T$) are:

$$P_{FTx} = RCM_x + cos(\alpha)R \tag{4.9}$$

$$P_{FTy} = RCM_y + cos(\beta)R \tag{4.10}$$

$$P_{FTz} = RCM_z + sin(\alpha)sin(\beta)R \tag{4.11}$$

To guarantee the RCM constraints, the tool frame position should always face the RCM point (the tool frame z_{axis} should cross the RCM point); to comply with this requirement, the final rotation should respect the RCM position. Thereby, the rotation matrix of the final end-effector position (P_{FR}) is represented by:

$$P_{FR} = [X_{axis} \quad Y_{axis} \quad Z_{axis}]_{3\times3} \tag{4.12}$$

where X_{axis}, Y_{axis}, and Z_{axis} are the axes that represent the final tool-frame orientation.

To find these axes, a $3D$ coordinate is initialized where one axis points toward the RCM position, it will simplify the problem to use axis perpendicularity to find the others. Using this property, these axes are found by:

$$Z_{axis} = normalize(P_{ST} - RCM) \tag{4.13}$$

$$X_{axis} = normalize((0,1,0) - Z_{axis}) \tag{4.14}$$

$$Y_{axis} = Z_{axis} \times X_{axis} \tag{4.15}$$

where × is the cross product and each axis represented as a $3D$ vector in space.

All the introduced steps above allow to control the 2 rotation constraints (R_x and R_y). For the $3rd$ rotation (rotation on the z-axis), the end-effector self-rotation around its axis is just applying the R_z rotation directly on the end-effector rotation matrix.

To control the translation on the z-axis, the needle insertion is equivalent to the decrease of the sphere radius. The position and rotation of the end-effector depend on the RCM position, where the translation on its z-axis will produce the end-effector position changes but keeping the RCM position with reference to the NHR base frame. The orientation of the end-effector will remain the same but the new translation position of the end-effector after applying a translation T_z on its z-axis will be:

$$P_{FTx} = P_{STx} + T_z(RCM_x - S_{Tx}) \tag{4.16}$$

$$P_{FTy} = P_{STy} + T_z(RCM_y - S_{Ty}) \tag{4.17}$$

$$P_{FTz} = P_{STz} + T_z(RCM_z - S_{Tz}) \tag{4.18}$$

And the radius (R) will be updated to:

$$R = R \pm \sqrt{(P_{FTx} - P_{STx})^2 + (P_{FTy} - P_{STy})^2 + (P_{FTz} - P_{STz})^2} \tag{4.19}$$

where $\pm$ is $+$ or $-$ operator which depends on the sign of T_z (insertion or extraction of the needle).

With the orientation and the position of the end-effector, and using the inverse kinematics model provided by the robot built-in library, the positions of the robot joint angles can be defined.

The redundancy that can provided by our 7-DoF robot compared to the 4-DoF necessary for the robotic-assisted UGRA is not an issue. On one hand, the alignment singularities are already well managed by the built-in Franka library; on the other hand, we have chosen an initial configuration of the robot that allows us to stay in the same configuration (as our workspace is reduced to $20 \times 20 \times 20\ cm^3$), and where the dexterity is maximum.

4.4 The control strategies

Robots were introduced in the operative room to provide important support in terms of accuracy and comfort for the surgeons during their daily routine. Different control strategies were introduced in the literature to control the RCM point. In UGRA procedure, a small error (up to 2-3 mm) at the RCM point is acceptable (transverse motion of the skin), but the needle tip error cannot reach the same error also to avoid any damages or trauma.

Robotic solutions are based on teleoperation or CoBotic control of the needle motion under RCM constraints. In general, these studies focus on reducing the errors at the RCM point and the instrument tip. In [190], the authors proposed to use a free-wrist anthropomorphic robot that guarantees the RCM constraints. Although the free-wrist control provides good control, it suffers from the fact that the control is only at the end-effector point using an open-loop strategy. Hence, the RCM is not controlled directly but it responds to the force/torque applied on the end-effector. This means that if there is an error at the RCM, an error will be also expected at the needle tip. Hence, in this study, we control the end-effector point in a closed-loop strategy while imposing RCM point. This results in knowing the exact poistion of the needle tip, which means that if there is an error at the RCM point, it will not affect the needle tip position (no error).

Here, we propose controlling the 7-DoF robot in a closed-loop using two control techniques: a haptic teleoperation control in Section 4.4.1; a CoBotic control in Section 4.4.2. While the haptic control is not a predefined task in the proposed robotic system, it is a preliminary task that was done to validate various tasks in the NHR system, as well as, it would be an option for improving the learning curve for junior anesthetist operators. Experiments of these control strategies are introduced in Section 4.6.2.

4.4.1 Haptic needle insertion teleoperation control: integration of the haptic force feedback and RCM constraints

Force feedback represents an important challenge for the community of medically robotic-assisted manipulators. Using force-feedback control provides several advantages such as: assisting the expert with structures identification by providing tissue palpation; avoiding nerve trauma or damage healthy tissue applied by large forces; reducing unintended injuries; identifying any interactions between the needle and the surrounding organs outside the viewing area [8].

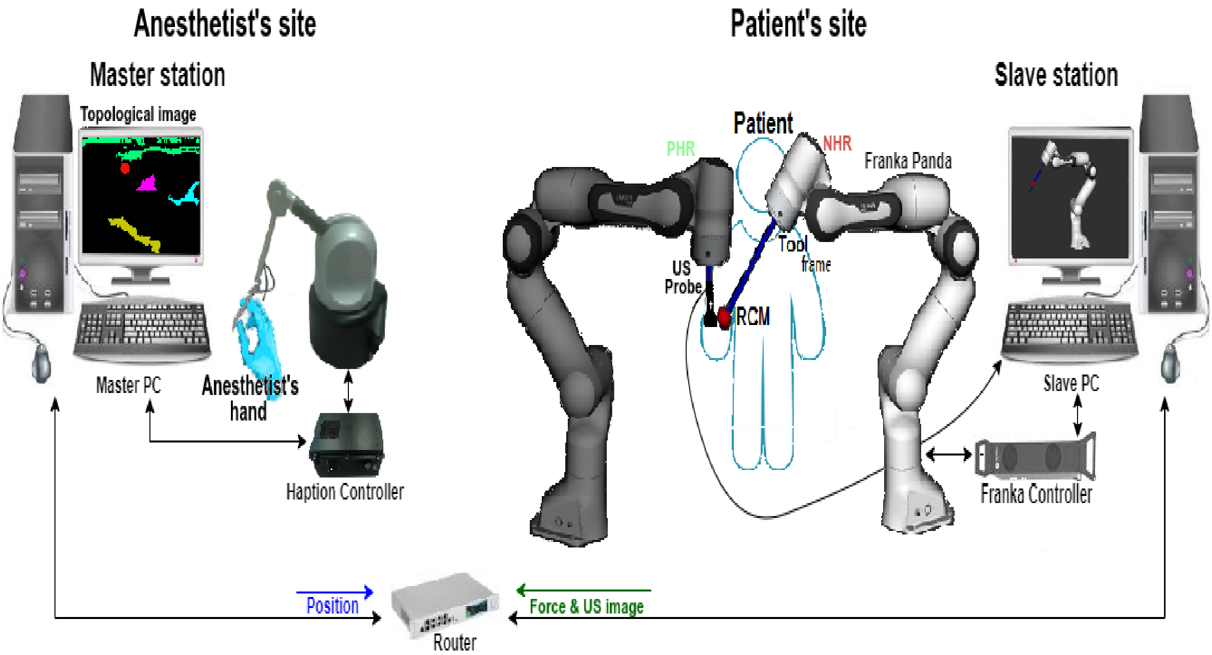


Figure 4.5: Schematic of the **teleoperated** UGRA system using a haptic device.

Teleoperated robotic-assisted UGRA is a solution that can provide the anesthetists with better control of the needle motion. This teleoperation approach allows the operator to control the needle insertion procedure from the anesthetist's site. There are different reported methods in the literature representing teleoperated robotic systems, such as in the DaVinci surgical system [191]. However, one of the teleoperated robotic-assisted system's major limitations is the lack of haptic force feedback. The haptic rendering experienced by the anesthetists, which is represented by the force feedback, provides crucial information for a more accurate and precise needle control. Therefore, this section proposes a haptic force feedback control framework for teleoperated robot-assisted UGRA for physical human-robot collaboration using 7 DoF robot manipulator (Franka Emika).

Feeling the forces, from the needle mounted on the slave robot while in contact with the environment, allows the anesthetist to get more information regarding the needle insertion process and to perform the UGRA act in a safer way with higher accuracy [192, 193]. There are different commercial force feedback devices, such as [194, 195], were used in various research works concerning teleoperated systems; the haptic phantom device from Haption(c) was used in the proposed platform.

To solve the force feedback issue during the needle insertion with RCM constraints, we present in this section a novel approach to assist anesthetists with comfort and confidence throughout the procedure. This approach can be summarized as moving the needle through the insertion point to generate a remote center of motion (RCM) and to provide force feedback to the expert through the haptic force feedback device as shown in Figure 4.5. To the best of our knowledge, there is no research that focuses on haptic force feedback for UGRA act under RCM constraints.

The proposed closed-loop control strategy based on force feedback consists of two parts: the haptic station (master station) and the robot station (slave station) as shown in Figure 4.5. The master system consists of a haptic interface and includes: (a) computer; (b) Haptic Virtuose device (Haption S.A. phantom) [196]; (c) haptic controller. The slave system consists of a needle mounted on a 7-DoF robot and includes: (a) computer; (b) Franka robot [197]; (c) Franka controller.

The master station is intended to provide the anesthetist a real sense of the force through the force feedback haptic device, alongside sending the intended end-effector position to the slave station. The slave station completes the control of the robot's position under RCM constraints and sends the force data back to the master station. In more detail, the anesthetist manipulates the haptic device (master station) as if it was the needle; the new position of the end-effector is sent to the robot station (slave station). On the patient's site, the slave robot applies the new position to the end-effector taking into account the RCM constraints and sends the forces back to the master station (the haptic device). To do so, the master device sends to the slave station the desired position which consists of the rotations (on x-axis and y-axis) and the translation on the z-axis. On the other side, the slave station sends the estimated forces back to the master station, which are the force on the x-axis, the force on the y-axis, and the force on the z-axis. The desired position is achieved in the master station by applying the RCM constraints explained in Section 4.3, and the returned forces are applied to the haptic device. While the anesthetist holds the haptic device, an augmented reality US image (introduced in Section 4.5) is provided based on US image segmentation. This image aimed to be used by the expert to facilitate the needle maneuver during the UGRA act.

The 6-DoF Virtuose 3D Desktop is used which allows the movements on both translations and rotations. However, the force feedback of the Virtuose 3D Desktop is only active on the first three axes (translations). For this reason and to enable the use of the force feedback in the proposed architecture, the phantom standard usage has been changed to use the first three axes as two rotations (R_x and R_y) and one translation (T_z) (the rotation on z-axis has been blocked because it is unnecessary in the proposed teleoperated system). Hence, the phantom base position has been

(a) The original haptic base position.

(b) The used haptic base position.

Figure 4.6: Description of the position and orientation of the haptic phantom.

changed (Figure 4.6) to be more suitable for our architecture and a teleoperated needle insertion.

To retrieve the changes in the position of the first three axes and to apply the received forces, the haptic built-in control library is used. The control is simple by retrieving the position and set the required forces.

4.4.2 CoBotized needle insertion control under RCM constraints - task Ⓕ

Collaborative Robots 'CoBots' are intended for direct physical interaction with humans in a shared workspace [198]. CoBots are distinct from teleoperators, in which a human operator controls a robot remotely. CoBotic control provides the accuracy of the robot, as well as, the flexibility and decision making of human participants. Therefore, the goal of this section is to assist the anesthetist with a CoBotic system that facilitates performing the UGRA procedure. It must be emphasized that CoBots are designed to be a complementary working tool for human workers. For that, the proposed system does not replace the anesthetist but assists the expert to perform UGRA act.

When the needle is inside the patient's body, the CoBotic system assists the anesthetist to respect the needle kinematics. This assistance guarantees RCM control around the insertion point. Furthermore, an augmented reality US image (introduced in Section 4.5) is provided based on US image segmentation. To the best of our knowledge, there is no research works that focus on CoBotic control for UGRA act and taking into account the RCM constraints.

In UGRA, the anesthetists apply forces on the needle handle which leads the robot movements by enforced software-defined guided constraints (RCM). As shown in Figure 4.7, in the proposed CoBot mechanism, the expert is the only operator who generates the movement. Thanks to the Franka Controller, the applied forces are retrieved and the desired position can be applied to the NHR end-effector. In order to create a more realistic operation process and make the force control as transparent as possible, a closed-loop control strategy is used based on the applied forces (Figure 4.7).

Franka Controller allows the robot to be controlled in joint or cartesian space and provides the direct and inverse kinematics models. At the NHR end-effector, the force sensor computes the external forces applied to each axis F_x, F_y, and F_z. To use the RCM constraints which have been presented in Section 4.3, the simplest way to do so is by converting the applied forces into two rotations and one translation. F_x and F_y are converted to the rotations R_x and R_y, while F_z is converted to the translation T_z. Furthermore, the rotation on the z-axis (R_z) has been blocked as it serves no purpose on the UGRA act.

Furthermore, to avoid nerve trauma or damage healthy tissue applied by large forces on the end-effector, the forces should have a maximum limit. Based on expert's recommendations, these limits consider as safety limits for the applied forces and sent to robot control to avoid any unwanted displacements. As well as, to avoid any robotics discontinuity errors, the applied rotations and translation should be reachable and do not exceed the predefined limits with predefined maximum velocity and acceleration. Therefore, while working at 1 kHz, the planning of smooth motion is essential.

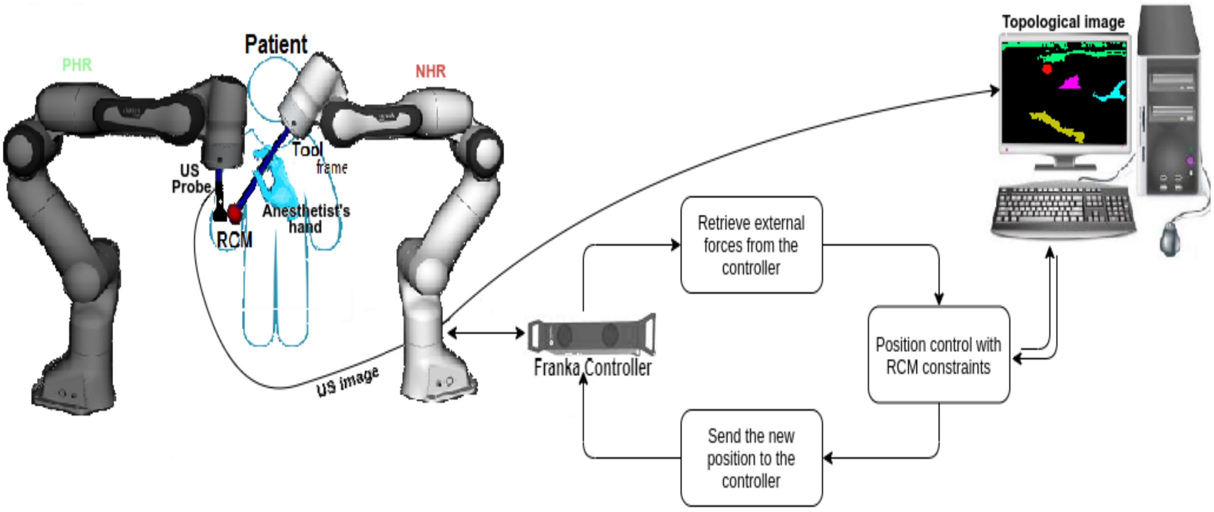


Figure 4.7: Schematic of the **CoBotic** UGRA system.

The considered smooth motion is an important key for controlling the robot, where the movement changes gradually until reaching the maximum predefined velocity. Therefore, the position should be controlled with respect to the velocities. Figure 4.8 shows the CoBotic control model. The model starts by retrieving the forces which are computed by the Franka controller. These forces ($F = [F_x \quad F_y \quad F_z]^T$) are converted into velocities ($V = [V_x \quad V_y \quad V_z]^T$) by taking into account the maximum predefined velocity and acceleration (are examined in the experiments). This followed by computing the desired rotations and translations (R_x, R_y, and T_z) which will be fed to Section 4.3.

The Franka controller allows us to work at 1 kHz which is sufficient to have good haptic feedback in the anesthetist's hand. Let's compute R_x after retrieving the applied F_x each 1 millisecond (ms) (1 kHz). First, the current velocity of the end-effector x-axis (V_x) is computed by

$$V_x = abs(F_x) * abs(V_{max}/Acc_{max}) \tag{4.20}$$

$$V_x = min(V_x, V_{max}) \tag{4.21}$$

where V_{max} and Acc_{max} are the predefined maximum velocity and acceleration.

And the computed rotation R_x each 1 ms is given by

$$R_x = R_x \pm (0.001 * V_x) \tag{4.22}$$

where 0.001 is the network running time of 1 kHz, and ± is + or − operator which depends on the sign of F_x

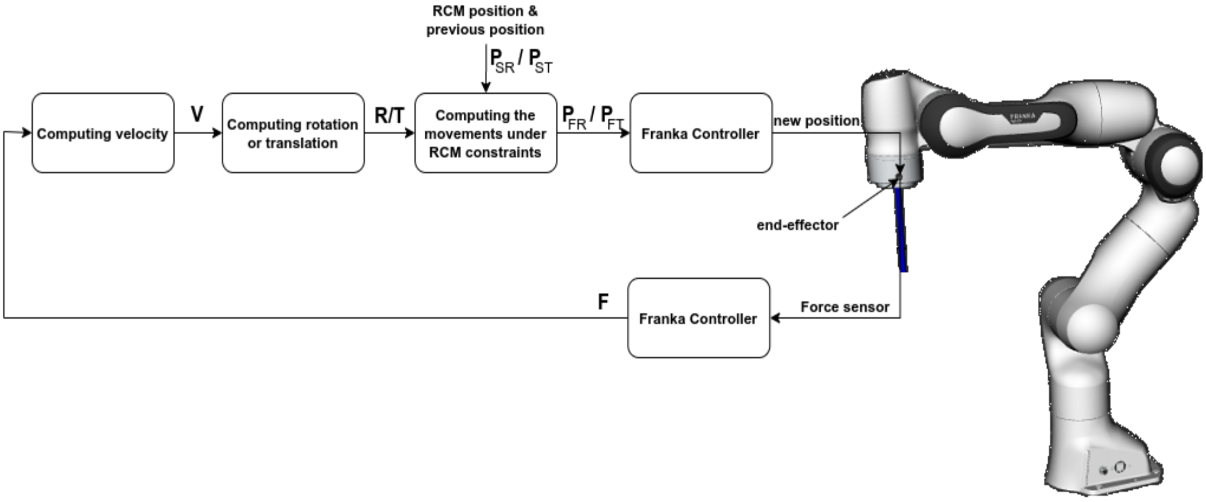


Figure 4.8: In the **CoBotic** control model, the robot retrieves the externally applied forces and converts it into position movements. F is the computed forces by the built-in Franka controller library, V is the velocity, R/T are the computed desired rotation (R_x and R_y) and translation (T_z), P_{SR} and P_{ST} are the previous end-effector position, and P_{FR} and P_{FT} are the new end-effector position.

This process is done also for F_y and F_z to compute R_y and T_z, respectively. Followed by computing the desired position using these rotations and translation parameters with RCM constraints that have been introduced in Section 4.3.

4.5 Risky regions avoidance

Risky region avoidance is an important step for a more complete robotic UGRA system. The objective is to gives the anesthetist freedom of movements while inserting the needle with the advantage of feeling an extra repulsive force while the needle's tip gets close to any risky region such as a nerve, an artery, a vein, or a tendon. Moreover, this procedure should not restrict the anesthetist to one path toward the target nerve. The process of risky regions avoidance starts by segmenting the US image to provide the anesthetist with an on-lined processed topological (augmented reality) image. However, segmenting an image frame is computationally expensive. Hence, a 2D mesh model is contructed from the topological image which will update the regions of interest location in the image. Then, small repulsive forces will be applied to the user's hand once the needle's tip gets close to a risky region.

In Section 4.5.1, we introduce a neural network model for segmenting the US image to produce the topological image. This is followed by presenting the 2D model in Section 4.5.2. While the force control strategy to avoid risky regions is proposed in Section 4.5.3.

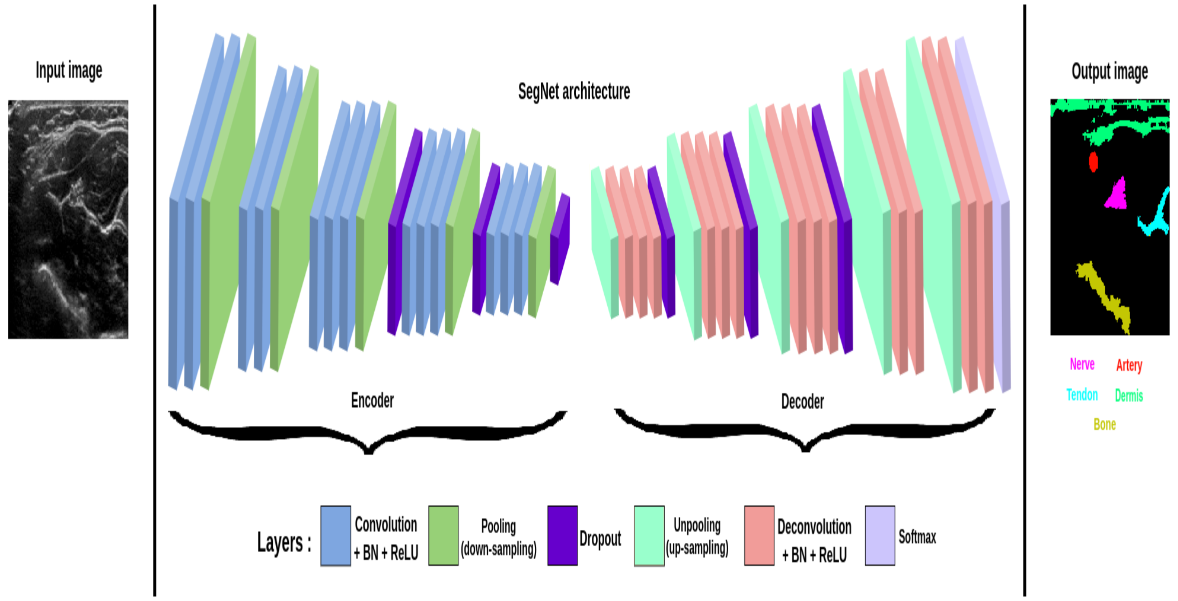


Figure 4.9: An illustration of the SegNet architecture for US image segmentation.

4.5.1 The topological image - task Ⓖ

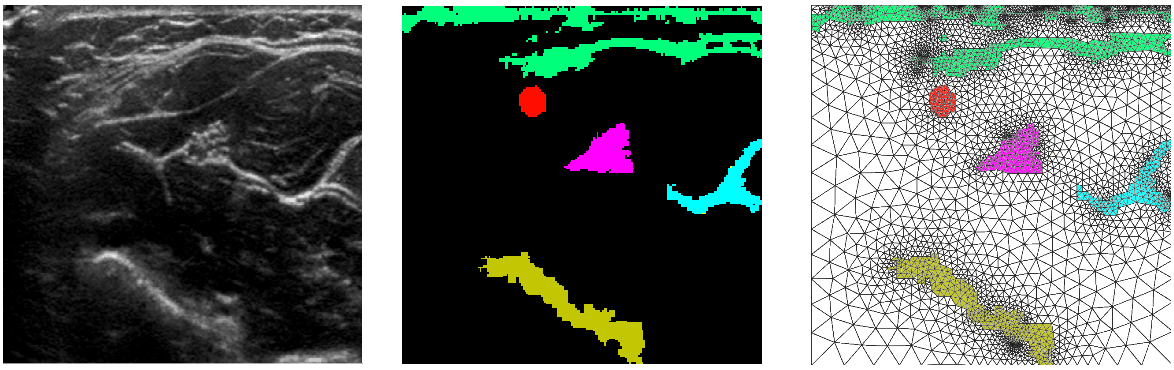
To date, research works have been developed to regions of interest in US images [131,138,199,200]. Nevertheless, no research focuses on segmenting all organs (regions) in the US image such as nerves, arteries, dermis, tendons, and bones. Hence, segmentation in US images topic requires further development and investigation.

Failure to locate the nerve and other regions could lead to nerve trauma or local anesthetic toxicity. The aim of this section is to provide the anesthetists with a tool based on the US image processing to handle this issue and improve the UGRA practice. Segmentation in US images is a challenging task since noise and other artifacts corrupt the visual properties of this kind of tissue.

In the last several years, various Convolutional Neural Networks (CNNs) architectures have been proposed in order to segment an image. Some of these architectures have been developed to directly segment the whole image, such as SegNet [201]. Hence, to segment the US image, we will use this tool. SegNet architecture is divided into Encoder network, Decoder network, and a final pixel-wise classification layer as shown in Figure 4.9. These networks are built by stacking multiple different layers where the essential layers are the convolutional layers and pooling layers.

The encoder network consists of 13 convolutional layers, 5 pooling layers, and 3 dropout layers. Each encoder layer has a corresponding decoder layer and hence the decoder network has 13 convolutional layers, 5 pooling layers, and 3 dropout layers. The final decoder output is fed to a multi-class softmax classifier to produce class probabilities for each pixel independently.

Each convolution or deconvolution layer has a set of various learned filters that produce different responses considered as the extracted feature maps. Each of these maps is normalized using Batch Normalization (BN) to enable end-to-end training. Then, the rectified linear unit acti-



(a) The original US image. (b) The topological image. (c) The mesh image.

Figure 4.10: An example of the 2D mesh creation.

vation function (ReLU) is applied to remove any negative value that results from the convolution.

Pooling has been used in the encoder and decoder network as down-sampling and up-sampling, respectively. Down-sampling divides each feature map into a set of non-overlapping regions and takes the maximum value for each sub-region as a representation of that sub-region. This leads to a reduction in the number of convolved features which make these features more robust to changes in the position. Up-sampling uses the memorized max-pooling indices from the corresponding encoder down-sampling and produces sparse feature maps.

The dropout layer is used as a regularizer in convolutional neural networks to prevent over-fitting and co-adaption of features. The high dimensional feature representation at the output of the final convolutional layer in the decoder is fed into to a trainable soft-max classifier. This soft-max classifies each pixel independently. The output of the soft-max classifier is an N channel image of probabilities where N is the number of classes (regions). The predicted segmentation corresponds to the class with maximum probability at each pixel.

4.5.2 2D model - task Ⓗ

In robotic control and as discussed before, all image regions should be segmented to be used in risky regions avoidance. Nevertheless, applying segmentation techniques in each image frame is computationally expensive. Studying the medium, and taking into account its dynamics, in which the needle will move is a good alternative. Considering that the US probe is fixed, the image will be fixed as well unless there is a tissue movement in the medium. This movement is generated by various forces due to the needle insertion in deformable tissues which means that certain points of this medium, in particular the nerve to be reached, will move under the effect of the needle movements within the tissues. In this study, we propose to update the topological image segmentation each 10 sec. Meanwhile and to reduce computation costs, meshes are obtained and analyzed to facilitate the further computation.

The mesh is a geometric data structure for representing surface subdivisions using a set of polygons. More simply, it is about a set of vertices, faces, and sides forming an object or a group of objects. These objects can be represented in 2D or 3D depending on the type of polygons used. In 2D as in the US image, the type of polygons mainly used are triangles and quadrilaterals.

The mesh is used to simplify calculations in a continuous space by discretization which is breaking it down into a set of finite elements in which the calculations are less complex to perform. The finite element method [202] (FEM) is a method of numerical calculation and used to perform calculations on an environment that is too complex for the direct calculations. FEM discrete the image into a known number of elements that have a different geometric shape. This results in creating the mesh model.

To conclude, using the mesh with the elastic model will provide all-region locations in every US frame until the new topological image is segmented. Figure 4.10 shows an example of creating the 2D mesh image.

For updating the mesh model while inserting the needle, a model of behavior should be associated to calculate the movements to each element in the mesh [203–205]. For US images, different models exist using deformation properties to update the mesh where it based on deformable environment fluidity. Three kind of elements are used to build these models, a spring, a dashpot, and a sliding frictional elements. The spring represents the elastic behavior (Hooke's law) of the element. The dashpot is a mechanical device that consists of a damper which resists motion via viscous friction. The damper represents the viscous behavior (Newton's viscous law) of the element. While the sliding frictional element represents the plastic behavior (irreversible deformation or hysteresis). Overall, the deformation models can be divided into three categories:

- **The plastic model**: is the most complete model(irreversible). This model takes into account the elasticity, the rate of deformation of the tissues, and their plastic behaviors during the application. This model consists of a spring, a dashpot, and a sliding frictional elements to find elastic, viscous, and plastic behaviors (Figure 4.11).

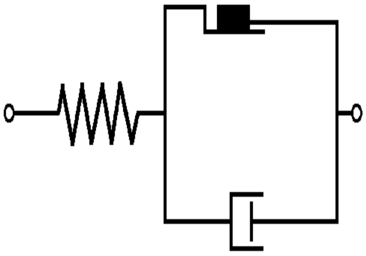


Figure 4.11: An example of the plastic deformation model (the Bingham–Norton model).

In the case of our system, the topological image and the mesh will be renewed every 10-15sec. Hence, taking into account this irreversible deformation model will then not be useful. Hence, it is better to take a less complex model the viscoelastic model.

- **The viscoelastic model**: models take into account the elasticity of materials as well as their rate of deformation. The basic operation is that the higher the speed of materials deformation the greater the viscosity of that material and therefore the greater the stress to be imposed on our materials for a given displacement. This model consists of a spring and a dashpot elements to relate elastic and viscous behaviors (Figure 4.12).

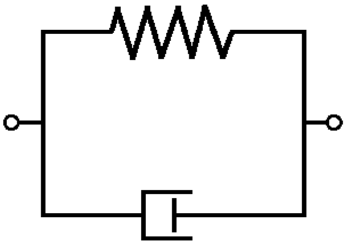


Figure 4.12: An example of the viscoelastic deformation model (the Kelvin–Voigt model).

For UGRA, the viscosity has no impact since the needle speed during the UGRA act is slow compared to the viscosity of the linear body. The deformation speed of tissues under the effect of the needle will not be significant and can even be considered negligible by comparing it with the term elasticity. For that, it is better to take a less complex model which is the elastic model.

- **The elastic model**: has the advantage of being a non-differential and linear model. This is the simplest model where it is composed of only one spring where the displacement of the spring is proportional to the stress applied to it. This model consists of just a spring to find the elastic behavior (Figure 4.13).

Figure 4.13: An example of the elastic deformation model (the Hooke model).

In UGRA, the tissues deformations will be represented having a purely elastic behavior under the stresses imposed by the movement of the needle during the insertion.

4.5.3 Force control for risky region avoidance - task Ⓘ

For risky regions avoidance, after creating the topological image and the 2D model, the repulsive forces should be analyzed. Overall, there are three different forces that exist during the robotic-assisted UGRA, two external forces and one applied internal force. The two external forces are the forces applied by the anesthetist on the end-effector and the retrieved forces from interacting the needle with the surrounding tissues. The internal force is a repulsive force applied on the end effector to avoid risky regions while inserting the needle.

For successful risky regions avoidance process, the connection between the two robotic systems the PHR and NHR in a shared workspace is essential. This connection will provide the needle's tip location in the US plane. In other words, the PHR will provide the US image and the NHR will compute the needle's tip location each ms. Hence, using the topological image with the needle's tip location will give the distance between the needle's tip and each region in the US image. These regions include the nerve due to the fact that the needle must inject the drug around the nerve and not inside it. Therefore, the nerve is considered as a risky region but with different applied repulsive forces. While moving under RCM constraints as discussed in Section 4.3, the applied force for region avoidance is under the same constraints too.

Let us take the 2D topological image with the needle's tip location in it, all distances between the needle's tip and image regions are calculated. For a more convenient procedure and to avoid sudden forces, the forces should be inversely proportional to the distance between the needle's tip and a region with predefined maximum forces. These predefined maximum forces are set as safety thresholds to avoid reaching high forces that could be harmful to the patient.

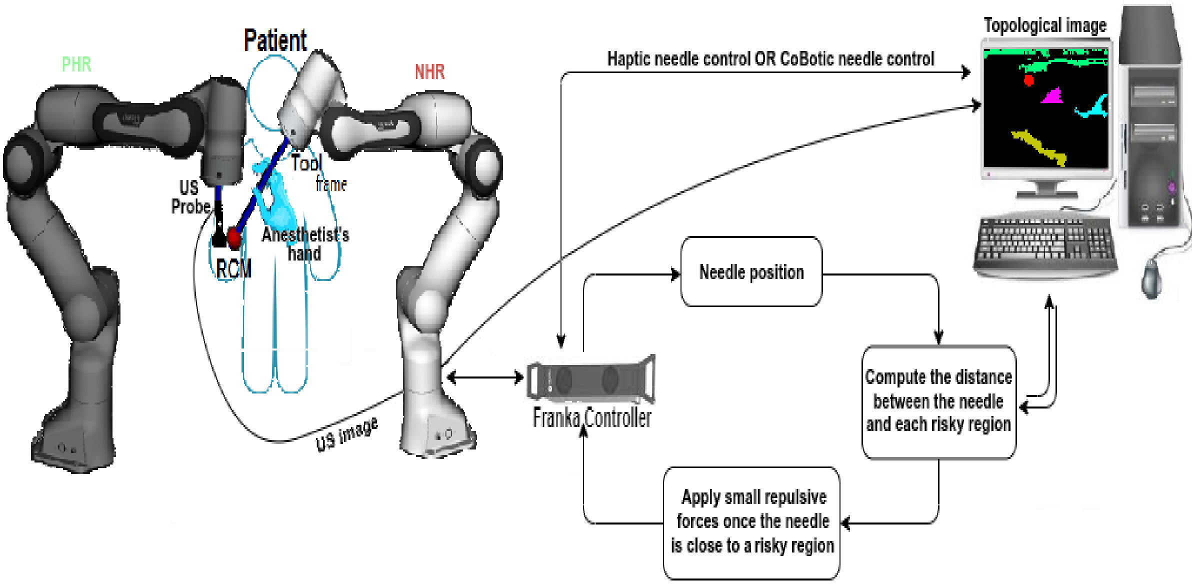


Figure 4.14: **CoBotic** needle insertion under RCM with risky regions avoidance.

Figure 4.14 presents the CoBotic needle insertion settings under RCM constraints with risky regions avoidance, where PHR provides the US image and NHR the needle's tip position. Merging the needle's tip location with the topological image will provide a 2D US image with all regions segmented in it. It should be noticed that two repulsive forces (one parallel to the US x plane and one to the depth z) will be applied since the distance calculations are in 2D.

Figure 4.15 shows the CoBotic control model under RCM constraints with risky region avoidance model. The model is similar to the CoBotic model in Figure 4.8 but with the addition of the repulsive forces. By taking the Euclidean distance, the distance D is calculated between each point in the needle's tip, the risky region, and the nerve. By taking these distances, small repulsive forces F' will be applied to the end-effector once the needle's tip is getting close to the risky region. Notice that these repulsive forces are in 2D (in x-axis and z-axis) due to the retrieved 2D image. These forces will not force the anesthetists to a specific path toward the nerve, but they will convey an alert once the needle's tip gets closer to a risky area.

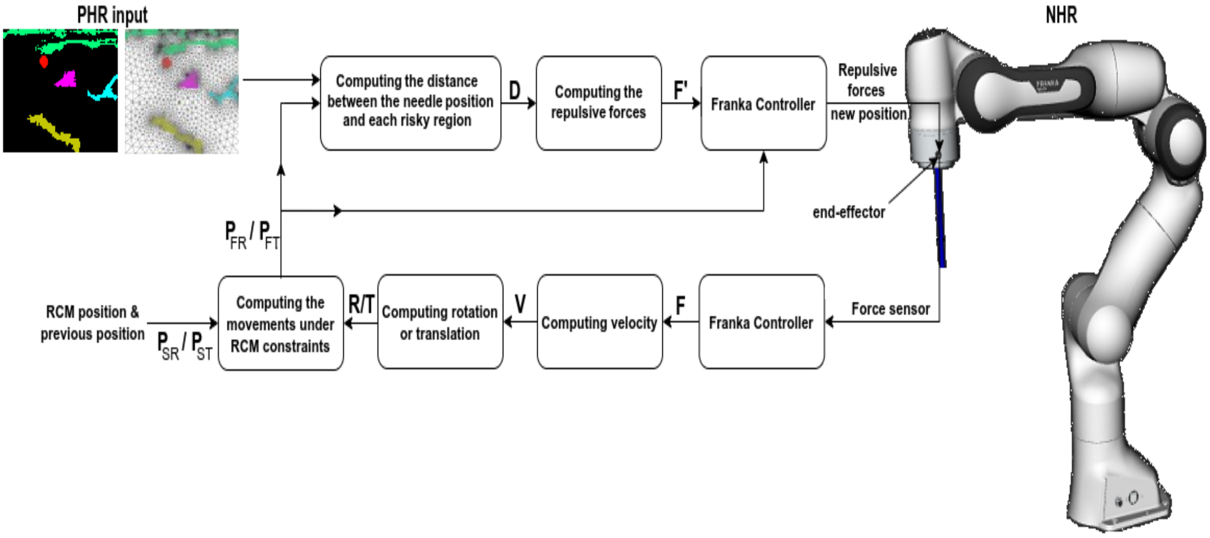


Figure 4.15: The **CoBotic** control model with risky regions avoidance. F is the computed forces by the built-in Franka controller library, V is the velocity, R/T are the computed desired rotation (R_x and R_y) and translation (T_z), P_{SR} and P_{ST} are the previous end-effector position, P_{FR} and P_{FT} are the new end-effector position, the PHR input is the 2D mesh retrieved from the PHR, D are the distances between the nerve and all regions in x-axis and z-axis, and F' is the applied forces on the end-effector which render to the anesthetist's hand.

4.6 Experimental setup, results and discussion

These experiments examine the efficiency of the proposed needle insertion in the robotic-assisted UGRA, and the performance evaluation of the proposed needle kinematics and control strategies. In this section, the used robot and setup are presented. Then, the needle insertion experiments are analyzed and discussed.

4.6.1 Setup and robots

The experiments were conducted using a 7-DoF Franka Emika Panda robotic arm [62]. Panda is a first generation collaborative robot system designed specifically to assist humans. Unique characteristics of the lightweight robot system designed to accomplish the most delicate tasks accurately. Architected, designed, and developed by Franka Emika GmbH in Munich and produced in Germany. The complete modularity, ultra-lightweight construction, highly integrated mechatronic design, sensitive torque sensors in all joints, and human-like kinematics, make the system unique. Based on the soft robotics control, inspired by human beings, Panda is able to recognize and process even the slightest touch by using its artificial reflex system to accomplish most delicate tasks accurately and safely within ms.

Figure 4.16 shows the Franka Emika Panda robot and its kinematic parameters according to the modified Denavit Hartenberg convention. This robot is equipped with 7 revolute joints, each mounting a torque sensor, and it has a total weight of approximately 18 kg, having the possibility to handle payloads up to 3 kg. It is possible to control the robot through the Franka Control Interface (FCI).

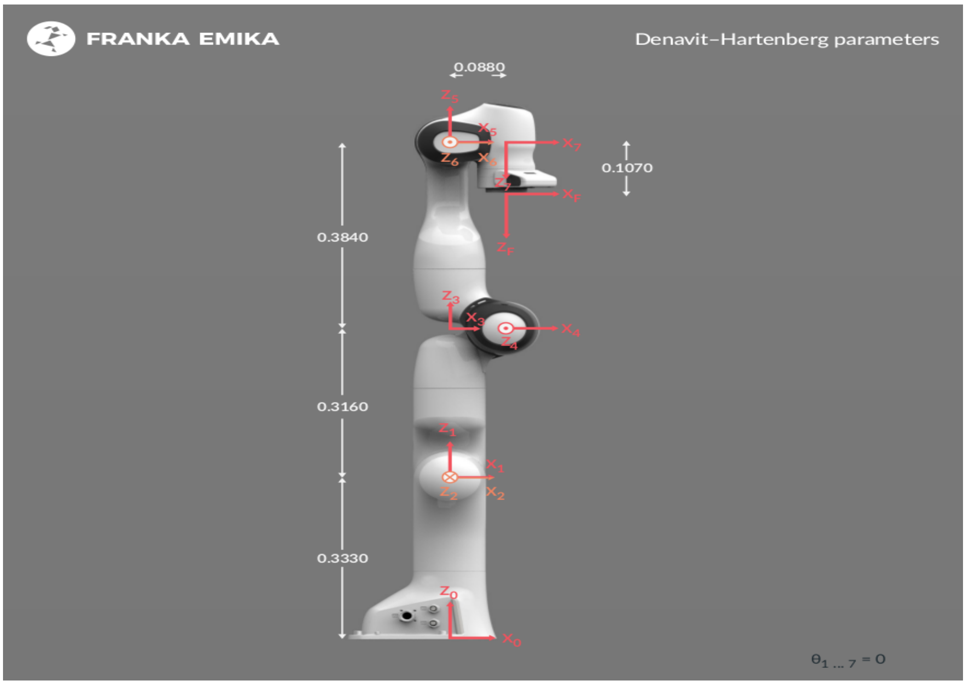


Figure 4.16: Panda's kinematic chain (Franka Emika).

The Franka Control Interface (FCI) allows a fast and direct low-level bidirectional connection to the robot. It provides the current status of the robot and enables its direct control with an external workstation PC connected via Ethernet. By using libfranka, the open-source C++ interface, one can send real-time control values at 1 kHz.

libfranka is the C++ implementation of the client-side of the FCI. It handles the network communication with Control and provides interfaces to easily: execute non-realtime commands to control and configure Arm parameters; execute realtime commands to run at 1 kHz control loops; read the robot state to get sensor data at 1 kHz; access the model library to compute the desired kinematic and dynamic parameters.

Realtime commands are UDP based and require a 1 kHz connection to Control. There are two types of realtime interfaces: Motion generators, which define a robot motion in joint or Cartesian space; Controllers, which define the torques to be sent to the robot joints.

In this study and as discussed in Section 4.3, to control the needle movements inside the patient's body, the most comfortable way would be to use the Cartesian space. Libfranka allows the robot to be controlled in Cartesian space and provides the direct and inverse kinematics models. This allows using the robot's built-in inverse kinematics instead of having to solve it.

The desired end-effector position is calculated where this position will be inputted to the

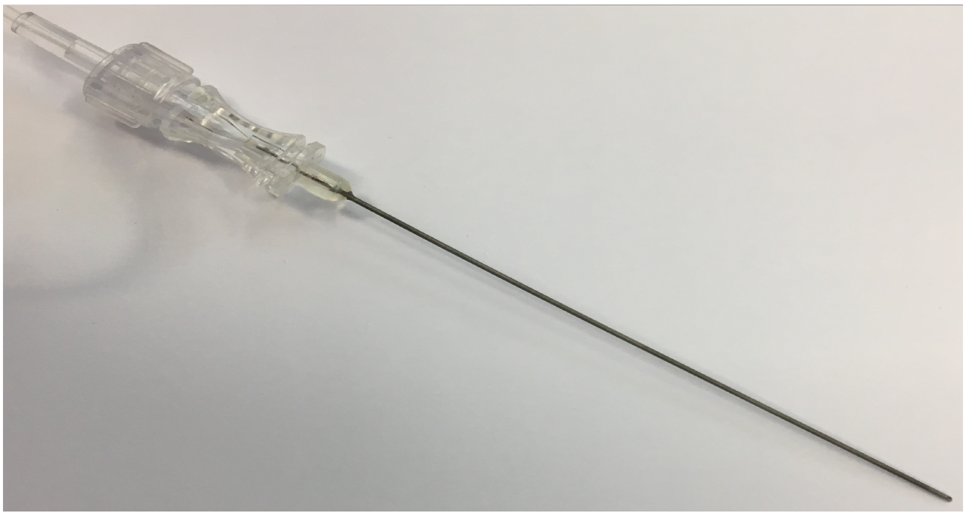


Figure 4.17: The anesthesia needle.

built-in inverse kinematic model. However, if the commanded values sent by the user do not comply with the interface requirements, an error will occur. This error indicates a discrepancy between the current robot values and the desired values sent by the user, in other words, the inverse kinematics solver of control yields to a joint configuration out of the limits. To counter this error, the desired values must take into account these limits to guarantee that the robot can reach the desired position using the inverse kinematics solver. For Cartesian space control, necessary conditions, such as the robot velocities, accelerations, or jerks, should be met to assure the control within the configuration limits.

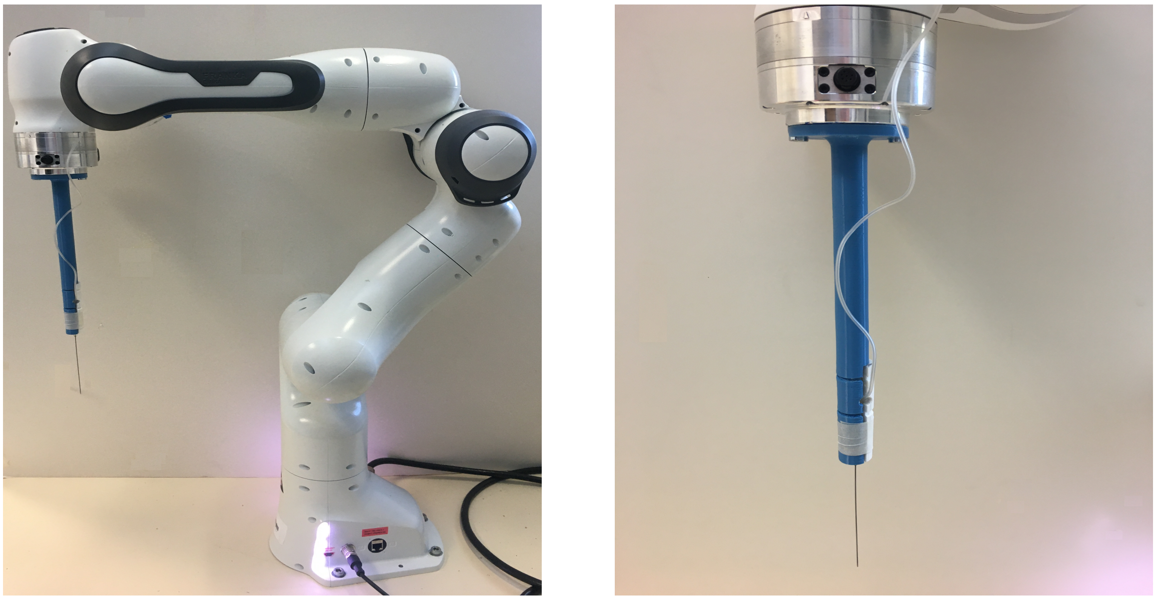


Figure 4.18: 3D printed needle holder.

Franka-ROS connects the robot with the entire ROS ecosystem. It is a metapackage that integrates libfranka into ROS Control. Originally developed in 2007 by the Stanford Artificial Intelligence Laboratory in support of the Stanford AI Robot project [206], ROS is an open-source operating system [207] for robots. ROS is steadily gaining popularity among robotics researchers as an open-source framework with a Linux platform for robot control. It offers hardware abstraction, low-level device control, implementation of commonly-used functionality, message-passing between processes, and package management. ROS uses the concept of nodes, messages, topics, services, stacks, and packages [207].

The proposed architecture was implemented as ROS nodes where it publishes and subscribes data with other nodes via ROS messages on the ROS topics. The communication between this interface and the robot is made possible thanks to the Franka Control Interface (FCI), which provides the current robot status and enables its direct control with an external workstation PC connected via local Ethernet at a communication rate of 1 kHz. In our work, the robot was assisted with a tool that we designed using a 3D printer to hold the needle (Figure 4.17) as shown in Figure 4.18.

In this work, where the needle in the UGRA procedure moves slowly and precisely, the position of the tool frame was controlled in the operational space with a maximum velocity of 0.03 m/s and acceleration of 2 m/s^2. This maximum acceleration is necessary to keep the fluent motion without inertial sensation at the anesthetist's hand.

For the haptic control, the connection between the interface and the phantom is achieved by the phantom controller using local Ethernet at a communication rate of 13 kHz which makes it very sensitive. To solve the sensitivity of the haptic device and to make the control between a 13 kHz haptic phantom and a 1 kHz Franka robot, an average strategy was applied to the haptic device. This strategy takes the average of the array of the 13 received position values in each ms, and updates it by adding the new position value and neglecting the first one.

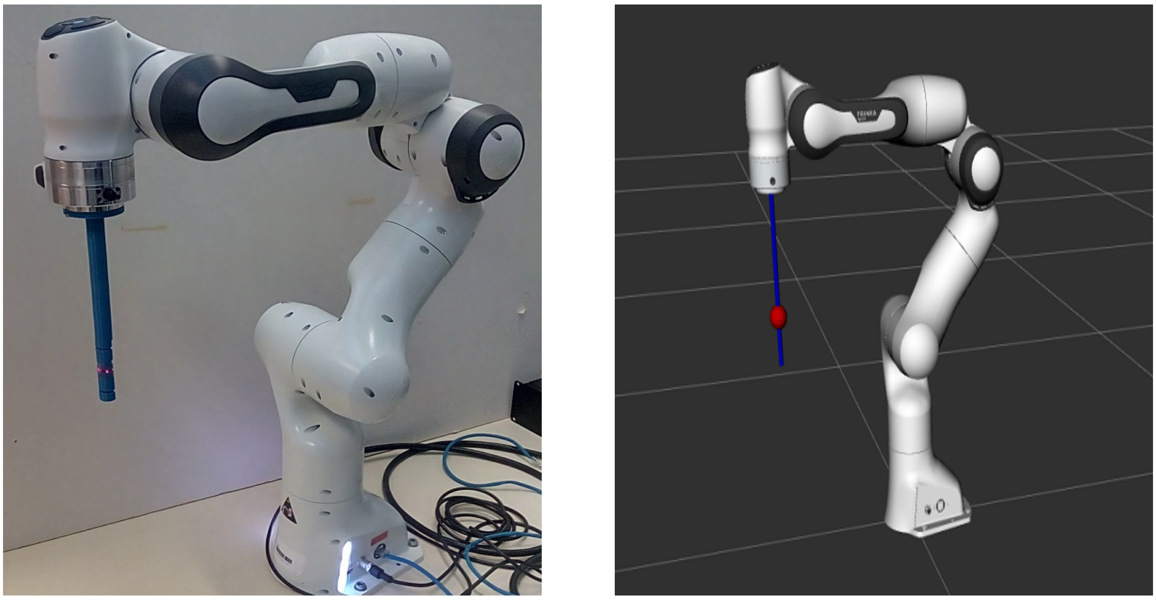
In order to allow communication between nodes in different computers, the environment variable ROS_MASTER_URI on every client PC must be set to the IP address of the PC where the master node is launched, namely the master PC. Additionally, the ROS_IP and ROS_PC_NAME environment variables must be set. In this experiment, the haptic station was set as the master PC and the robot station as the slave PC.

4.6.2 Experimental results and discussion

The objective of this experiment is to evaluate the proposed techniques in order to study their role and impact on needle control. Section 4.6.2.1 describes the experiments of the needle insertion under RCM constraints, and the results of US image segmentation for risky regions avoidance are presented in Section 4.6.2.2.

4.6.2.1 Needle insertion under RCM constraints

In this study, we used position accuracy as it is the major metric commonly used to evaluate RCM methods in medical robotics. Basically, position accuracy is the Euclidean distance between a given **insertion point** and the **RCM point** are computed by the proposed method each ms. The proposed method is validated with this metric and it is validated using 2 orthogonal laser beams that point out at the insertion point and at the sphere center in Rviz visualization as shown in Figure 4.19.



(a) The real robot.

(b) The Rviz display.

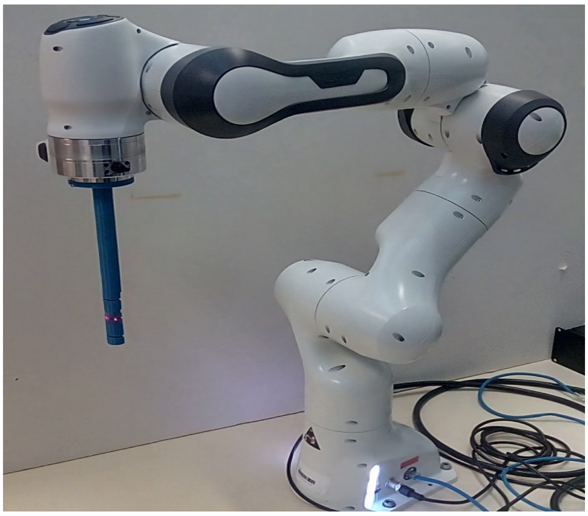
Figure 4.19: Robot setup with RCM representation as 2 laser beams on the real robot and at the sphere center in Rviz visualization.

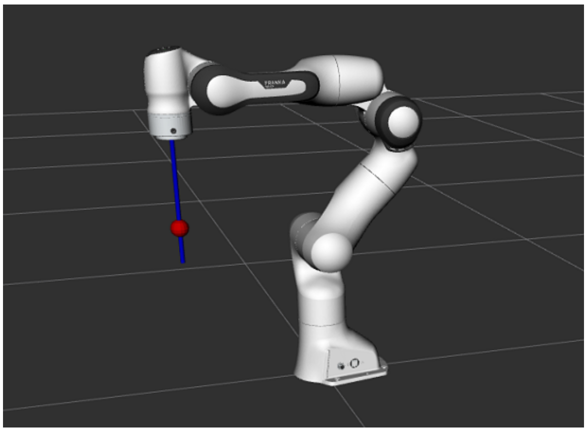
4.6.2.1.a RCM preliminary control test for needle insertion:

Figure 4.20 shows example when applying different rotations and translations on the Franka robot end-effector using a local control by joystick. Various rotations and translations were performed in each case. The position accuracy of RCM constraints has been computed and compared with the insertion point fixed position. These experiments have been repeated by taking the Euclidean distance between the RCM position and the insertion point position, the Euclidean maximum error is found less than 1 mm. The results suggest that with a given configuration control (the proposed RCM constraints control), the manipulator has a highly accurate performance when it is operating with respect to RCM constraints.

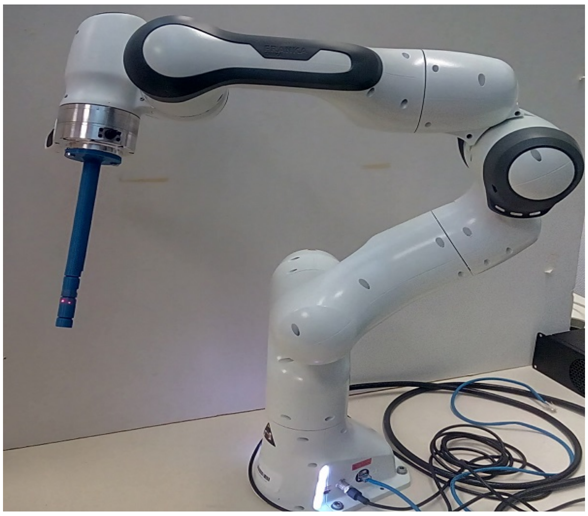
4.6.2.1.b Teleoperated haptic force feedback control under RCM constraints:

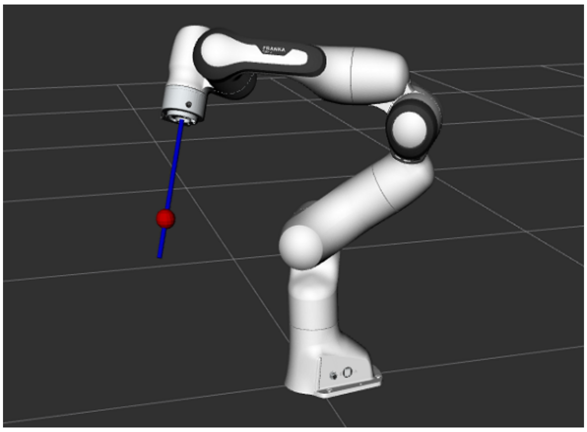
In the previous section, we validated the accuracy of the proposed RCM control method. Now, the teleoperated haptic force feedback control is added to it. To examine the RCM constraints error using the haptic device, the Euclidean distance error between the RCM position and the insertion point position is computed which is less than 1 mm. Next, the teleoperated force feedback experiment was carried out; Figure 4.21 illustrates the desired force (the insertion force applied on the needle at NHR), the actual force (feedback force rendered by



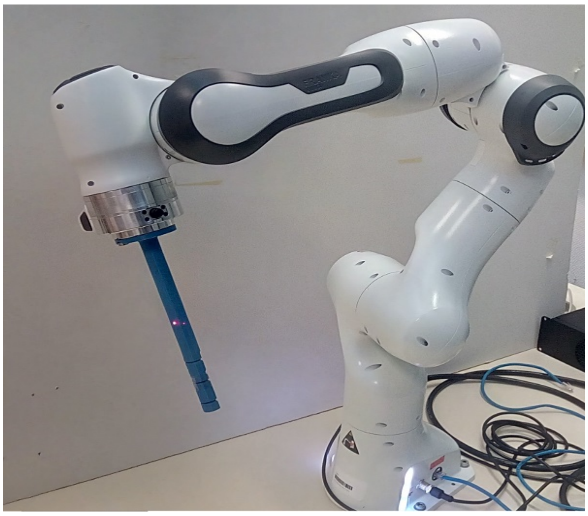


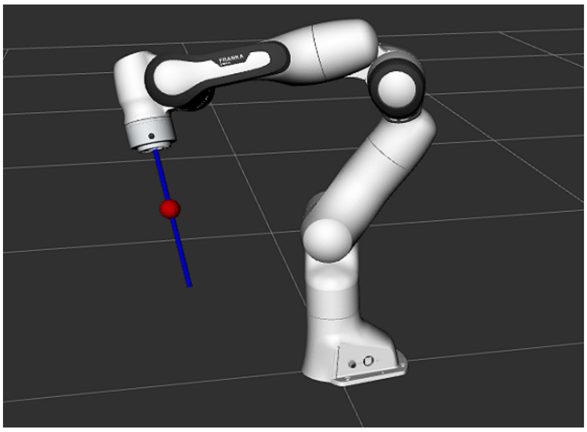
(a) The robot starting position.





(b) The robot position after applying only rotations ($R_x = 15°$ and $R_y = -15°$) at time (t) = 10 sec.

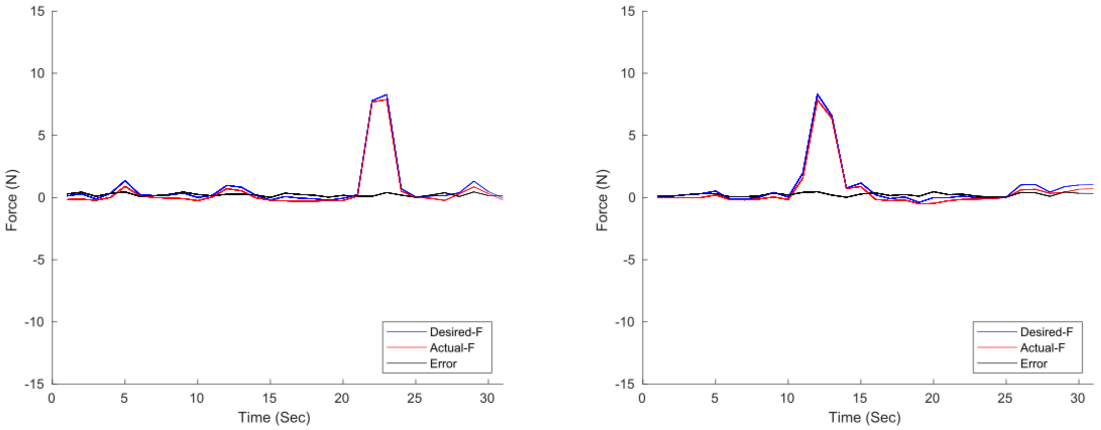




(c) The robot position after applying different rotations AND z-translations ($R_x = -15°$, $R_y = -15°$, and $T_z = 10cm$) at time (t) = 20 sec.

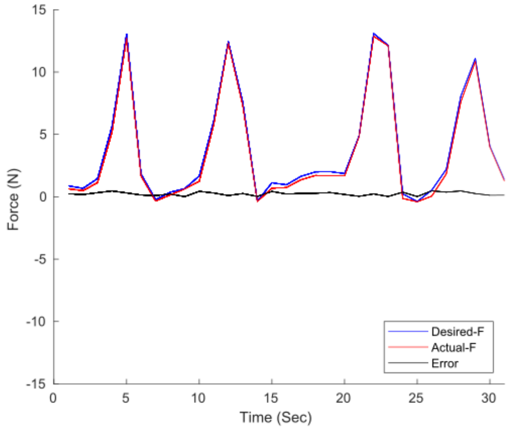
Figure 4.20: RCM results, where real robot images on the left column and the Rviz display on the right one.

the haptic device to the anesthetist's hand), and the force error between the two forces. It can be seen from Figure 4.21 that the error between the actual feedback force and the insertion force approaches zero (<0.1 N). As a result, the proposed control method offers good transparency in the force feedback using a haptic phantom.



(a) Force-x

(b) Force-y



(c) Force-z

Figure 4.21: Force error characterization: force along the x-axis (a), force along the y-axis (b) and force along the z-axis (c).

4.6.2.1.c CoBotic control under RCM constraints:

The CoBotic experiments were implemented by applying forces on the NHR end-effector and computing the RCM constraints error. Figure 4.22 depicts Euclidean distance error between the RCM position and the fixed insertion point position after applying different CoBotic control forces (on x, y, and z) for more than 60 sec. This error is computed by taking the 3D Euclidean distance between the desired RCM position and the calculated new position. It can be observed from Figure 4.22 that the error between the desired RCM position and the fixed insertion point position is less than 1 mm. Therefore, the results show that

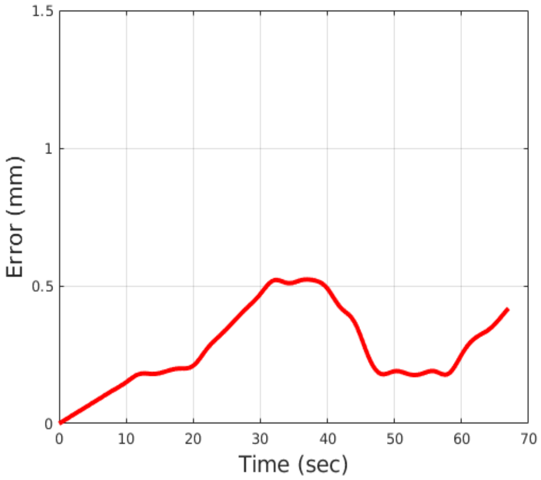


Figure 4.22: The Euclidean distance error between the RCM position and the fixed insertion point position using CoBotic control under RCM point constraints (hand force is applied by the operator on the robot arm).

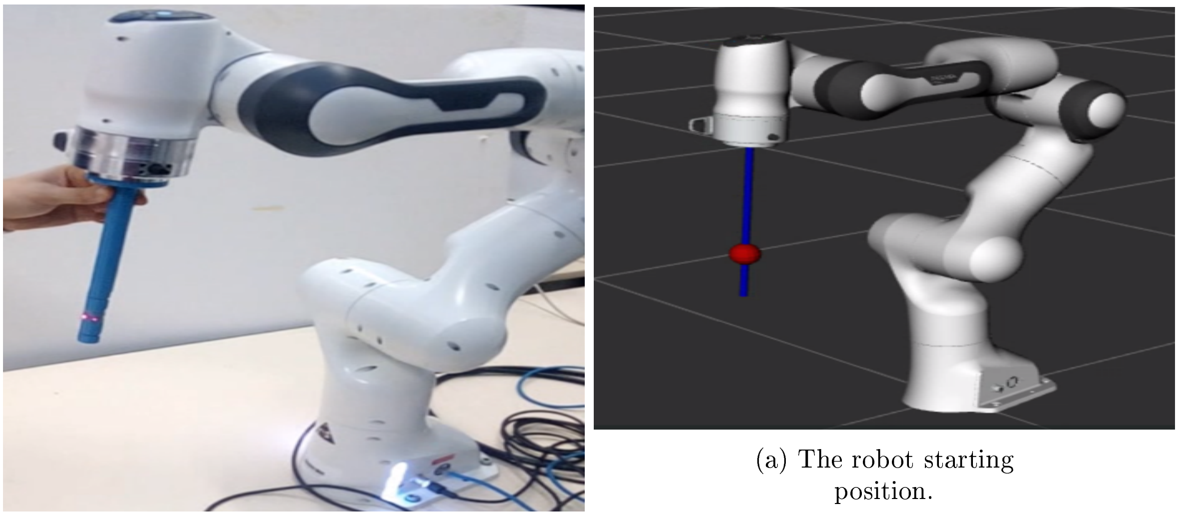
the proposed CoBotic control method has an acceptable result of performing the UGRA procedure with accurate and precise control. Figure 4.23 shows an example of applying different rotations and translations on the Franka robot using the CoBotic control under RCM constraints.

4.6.2.2 Risky region avoidance : topological image

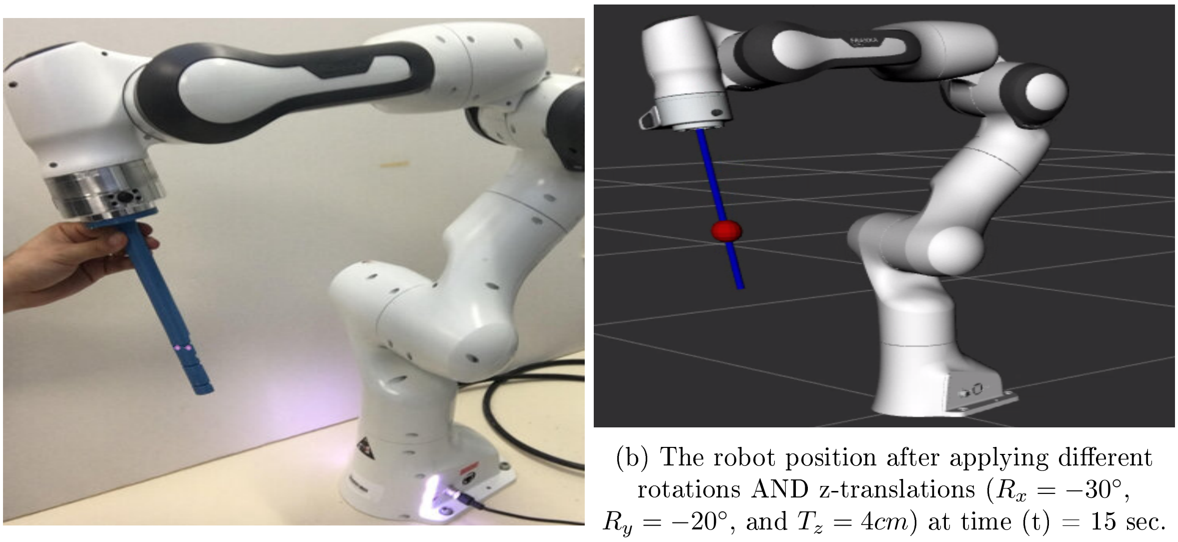
Risky region avoidance process gives the anesthetist the advantage of feeling an extra repulsive force once the needle's tip gets close to any risky region such as a nerve, an artery, a vein, or a tendon. This process starts by segmenting the US image to provide the anesthetist with an on-lined processed topological (augmented reality) image. US image segmentation experiments were carried out using PC with a Core 7 Duo 3.50GHz processor with 32GB RAM. Experiments were conducted on sonographic videos of the median nerve obtained from 20 anonymous adult patients (12 men and 8 women) using a US machine with a 5-12 MHZ transducer frequency. US videos of the median nerve were obtained from several volunteer patients under real conditions where the ground truth was provided by two regional anesthesia experts. The databases are in the form of a video for each patient. Each video consists of 650-750 frames and a total number of 14,000 US images of the median nerve were used. The used images in our experiments have a dimension of 600 × 350 pixels.

The Segnet model is designed under Python with the public deep learning libraries TensorFlow and Keras [173]. In these experiments, 10 videos were selected randomly as a training dataset and the remaining 10 videos as a testing dataset. For setting the dataset, each image was labeled manually and validated by the experts. During the training phase, the SegNet network uses the images with of 360 × 480 pixels, with their labels. The number of iterations was set to $200,000$ and the batch size is set to 3 which composed of 3 images selected randomly.

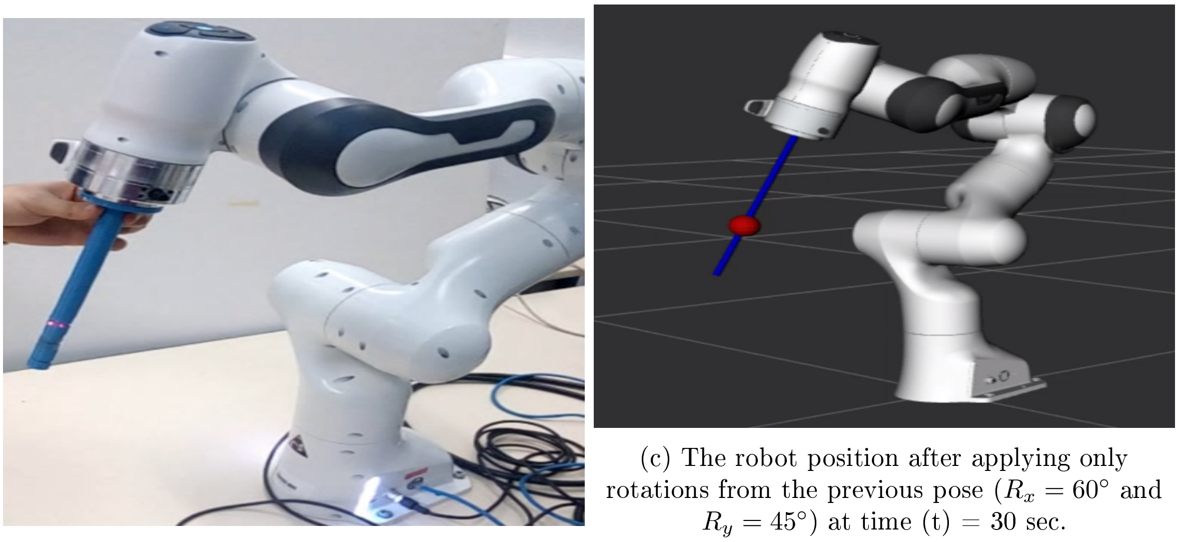
For testing an image, the testing phase returns the predicted labeled image. The output



(a) The robot starting position.



(b) The robot position after applying different rotations AND z-translations ($R_x = -30°$, $R_y = -20°$, and $T_z = 4cm$) at time (t) = 15 sec.



(c) The robot position after applying only rotations from the previous pose ($R_x = 60°$ and $R_y = 45°$) at time (t) = 30 sec.

Figure 4.23: CoBotic control under RCM constraints results, where real robot images on the left column and the Rviz display on the right one.

image is compared with the ground truth to estimate the segmentation accuracy. Accuracy is the most common performance measure, but the main drawback is that the accuracy hides some details that can help understanding better the segmentation model performance. Hence, Recall, Precision, and F1-score provide better performance understanding by taking both false positives and false negatives into account. Therefore, we evaluated the segmentation process by Precision, Recall, and F1-score values which are expressed by:

$$Precision = \frac{TP}{TP + FP} \tag{4.23}$$

$$Recall = \frac{TP}{TP + FN} \tag{4.24}$$

$$F1 - score = \frac{2 \times (Precision \times Recall)}{Precision + Recall} \tag{4.25}$$

where TP, TN, FP, and FN are the number of samples for ”True Positive”, ”True Negative”, ”False Positive” and ”False Negative”, respectively.

To show the effectiveness of the method, Table 4.1 depicts the segmentation average results. If there is a match between the ground truth and the SegNet estimation, it is considered as true positive, otherwise, it is a false positive. The results show the strength of the model which has an average segmentation Precision and Recall (>80%) for all classes. It can be observed from Table 4.1 that the adopted methodology can successfully segment the US images, which is important to achieve a successful CoBotic UGRA with regions avoidance.

Table 4.1: A comparison results for SegNet segmentation average.

Class	precision	recall	F1-score
Nerve	0.92	0.91	0.91
Artery	0.95	0.95	0.95
Tendon	0.94	0.87	0.90
Bone	0.95	0.85	0.89
Dermis	0.96	0.81	0.88

Time complexity is considered a crucial point, especially for US image segmentation in UGRA. Important aspects that affect the running time for any descriptor are feature extraction time and the feature dimensionality. In literature, it is generally the performance results of the segmentation method that are focused on, while time complexity has received less attention. In our experiments and to provide the topological image, the average reported time is 4.8 seconds per image. This average is calculated over 7000 images of size 360 × 480 without including the training time. The results show great performance and time since the topological image is updated each 6-8 seconds in the proposed robotic system. In addition to the possibility of using GPUs and parallel programming which ensure a faster process.

Qualitative results are shown in Figure 4.24, where it can be seen that the nerve and the other regions were successfully segmented compared to the expert's groundtruth. In the same figure, it can be observed the mesh is created for each image. Updating the 2D model and the force control for risky region avoidance are presented completely in Section 4.5. Nevertheless, these two tasks (Ⓗ and Ⓘ) are not included in our platform and will be a part of our future work as two robotic stations are needed (NHR and PHR).

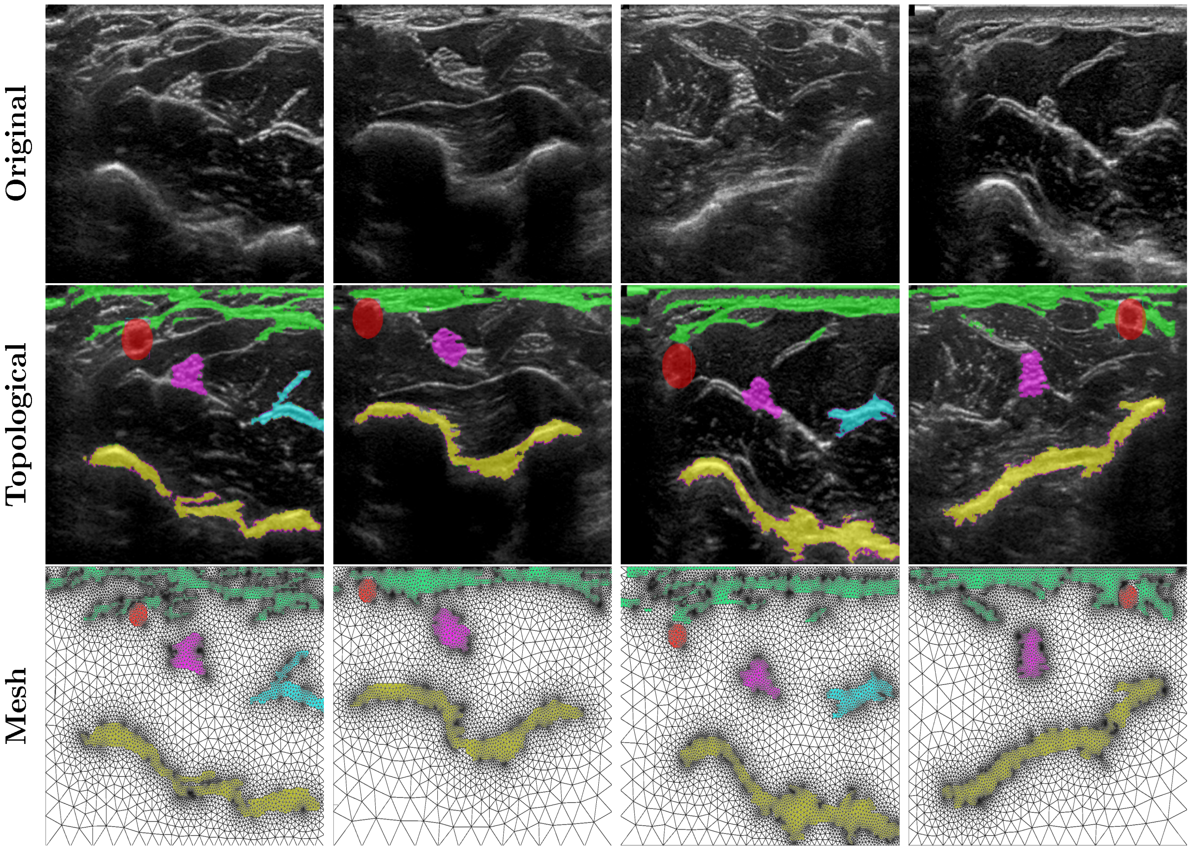


Figure 4.24: Qualitative results of US image segmentation using the SegNet model from 4 videos of 4 different patients.
🟪 Nerves in pink. 🟥 Arteries in red. 🟨 Bones in yellow. 🟦 Tendons in blue.
🟩 Dermis in green.

4.7 Conclusion

In this chapter, we addressed a challenging issue in robotic-assisted UGRA which is needle insertion control. We have presented an experimental system with the Remote Center of Motion (RCM) constraints. It enables full Cartesian control of the robot end-effector based on the needle insertion kinematics. The proposed method was developed using ROS and a 7-DoF Franka Emika robot. To evaluate the accuracy of the system, a performance metric is used based on the Euclidean distance between the RCM point and the fixed insertion point. The proposed method is accurate to find the exact position and rotation of the end-effector with an error of less than 1 mm while maintaining the needle insertion kinematics. Two control strategies were presented while using the RCM Cartesian control: a teleoperated UGRA needle insertion system with haptic force feedback; a CoBotic needle insertion system.

The teleoperated UGRA control enables needle insertion under RCM constraints with force feedback applied to the haptic device (6-DoF Virtuose 3D Desktop). The proposed method is accurate, as it finds the exact position and rotation of the end-effector with an error of less than 1 mm (The Euclidean distance error between the RCM point and the fixed insertion point). To evaluate the teleoperated force feedback system, the error between the desired force (the insertion force applied on the needle at NHR) and the actual force (feedback force rendered by the haptic device to the anesthetist's hand) is computed. The teleoperated force feedback system found to be accurate and sensitive with an error approaching zero ($<$0.1 N) between the desired and the actual forces.

On the other hand, the CoBotic UGRA control enables the needle movements control by applying hand force by the operator user on the robot arm while respecting the RCM constraints. To evaluate the accuracy of the system, the Euclidean distance error between the RCM point and the fixed insertion point is computed which is less than 1 mm while maintaining the needle insertion kinematics.

Finally, we discussed avoiding risky regions by applying small repulsive forces to the end-effector once the needle's tip gets close to a risky region. A topological image has been constructed by using a neural network segmentation technique (SegNet). In the experiments, the results show that the used technique obtained high accuracy ($>$80%) for segmenting the US image.

Conclusion and perspectives

Conclusion

Ultrasound-Guided Regional anesthesia (UGRA) has become a standard procedure in surgical operations and pain management; it offers the advantages of nerve localization, and provides regions of interest visualization. Despite a long training process and daily acts that the anesthetists perform, the UGRA procedure can still benefit from robotics and image processing researches. The main objective of this thesis is to facilitate the daily routine of the anesthetists by introducing the robotic-assisted system for UGRA. It must be emphasized that this system is designed to be a complementary working tool for the medical operators to assist them while performing the UGRA act. This robotic assistance is developed to improve the procedure accuracy and safety such as avoiding nerve trauma or reducing healthy tissues damage. Furthermore, it could increase the anesthetist's experience by more teaching and learning to avoid unintentional injuries.

This thesis is a part of DANIEAL2 project and proposes a complete robotic system for the UGRA procedure where we focused on 3 main issues: nerve detection, nerve tracking, and needle insertion. Figure I recalls the proposed robotic system where achieved tasks within each phase are highlighted in yellow. The major contributions are summarized as follows:

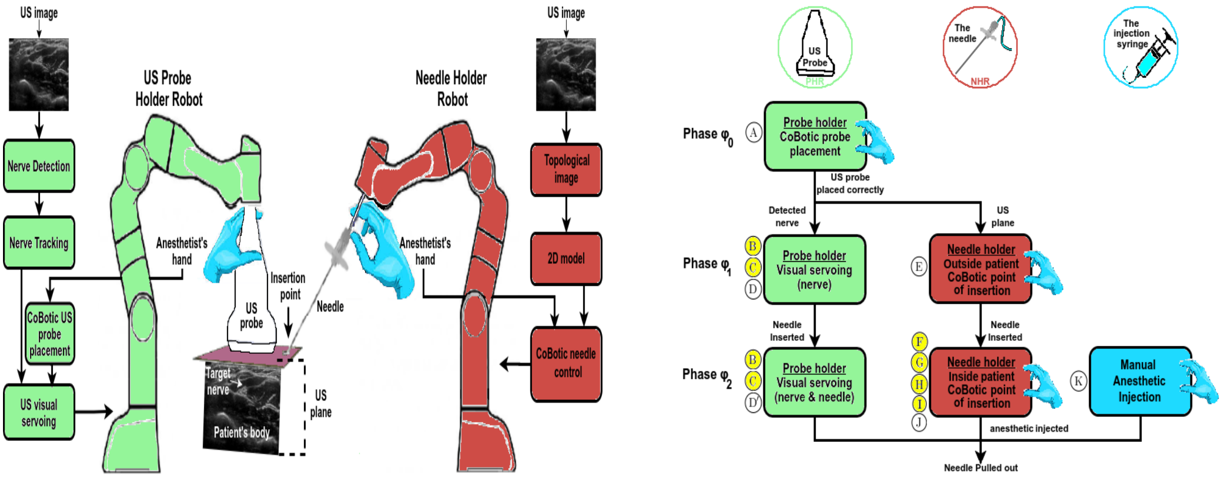


Figure I: The proposed robotic-assisted UGRA procedure.

- We proposed a complete robotic system for the UGRA procedure in Chapter 1. This robotic system can provide great assistance to the experts with techniques and tools that improve the procedure accuracy and safety such as avoiding nerve trauma or damage to healthy tissues. Moreover, it can be used as a learning tool for junior professionals.

- Nerve detection and tracking problems in the ultrasound (US) image require to take into account complex information embedded in the image which are the features. Any detection and tracking method depends mostly on choosing suitable image features. since the US image has the noisy texture properties, we proposed a new Robust Adaptive Median Binary Pattern (RAMBP) feature descriptor. RAMBP descriptor takes the advantage of pixel classification and the adaptive analysis to provide strong discriminativeness and noise robustness properties. RAMBP has been evaluated on noisy textures including different kinds of noise. Experimental results indicated that RAMBP outperforms other existing descriptors for handling high noisy textures classification. However, we believe Convolutional Neural Networks (CNNs) models are effective too for the US image textures. Thus, to learn more optimal features, we propose to merge RAMBP with CNN models to achieve better performance for nerve detection and tracking.

- Accurate and consistent nerve detection and tracking is essential for safe and efficient UGRA. In this thesis, the detection and tracking methods are based on RAMBP (texture descriptor), CNNs models, or the merged model of RAMBP and CNNs model. Although the texture descriptor or CNNs models based detectors and trackers perform good in US images, the results show that the merged model obtained higher accuracy by 10%. It is worth mentioning that the merged model combines the RAMBP and CNN architecture together which takes advantage of the strength of each side and improves the overall performance. Furthermore, different techniques were presented for detecting and tracking the nerve in the US image. For nerve detection, the obtained detection results show that the merged model based detector outperforms other techniques and achieved the best performance with 96% precision. For nerve tracking, we found that the merged model based trackers achieved the best results where we believe that the mix tracker (particle filter and the search technique) provides a good trade-off between tracking accuracy (with 93%) and time complexity (with 0.19 spf).

- Needle insertion is a very challenging task encountered by anesthetists in their daily routine and especially when they have to. Maneuvering the needle around the interaction pivot point between the needle and the skin of the patient is named the insertion point. This point restricts the anesthetist to control the needle to only four degrees of freedom (DoF) motion: three rotational DoF and one translational DoF. In robotic, this restriction is called the "remote center-of-motion (RCM) constraints". We have presented an experimental system by enabling full Cartesian control of the Franka emika robot end-effector under RCM constraints. Furthermore, we presented two different control strategies while using the proposed RCM control: a teleoperated UGRA needle insertion system with haptic force feedback using a 6-DoF Virtuose haptic device; a CoBotic needle insertion system by applying hand forced by the operator user on the Franka emika robot arm. To evaluate the accuracy of respecting the RCM constraints, the Euclidean distance error between the RCM point and the fixed insertion point is compted. The proposed two control strategies are accurate to find the exact position and rotation of the end-effector with an error of less than 1 mm while maintaining the RCM constraints. To evaluate the teleoperated force feedback system, the force error between the desired force (the insertion force applied on the needle at NHR) and the actual force (feedback force rendered by the haptic device to the anesthetist's hand) is computed. The proposed system found to be accurate and sensitive with a force error approaching zero ($<$0.1 N). Furthermore in this work, we discussed avoiding risky regions by applying small repulsive forces to the end-effector once the needle's tip is close to a risky region. For that, a topological image is constructed by using a neural network segmentation technique (SegNet). In the experiments, the results show that the using technique obtained high precision 90% for segmenting the US image.

Perspectives

Significant number of ideas remain untapped and will need to be analyzed and studied in detail. The future directions foreseen for a possible follow-up can be broken down into short term perspectives and long term perspectives.

- **Short term perspectives**

  1. We introduced the robotic-assisted UGRA system which is divided into several tasks as discussed in Chapter 4. In this thesis, we presented some tasks while 5 tasks have to be looked at:

     (a) CoBotic control of the probe holder robot for probe placement on the patient's body - task Ⓐ.
     (b) Visual servoing to control the probe holder robot for keeping the visualization of the nerve - task Ⓓ.
     (c) CoBotic control of the needle holder robot for needle placement in the US plane - task Ⓔ.
     (d) Visual servoing to control the probe holder robot to keep the visualization of the nerve and the needle in the US image - $\textcircled{D'}$.
     (e) Trajectories proposition of the needle to reach the nerve - task Ⓙ.

  2. We performed the detection and tracking tasks on real data videos. For future work, the detection and tracking tasks could be done directly in realtime using US system.

  3. For the proposed robotic needle insertion control, technical validation on anatomical phantom could be performed which will be one step for evaluating the proposed system toward clinical applications.

4. The possibility of using parallelization on GPUs platform or FPGAs for increasing the performance speed for the computer vision tasks.

- **Long term perspectives**

  1. In this thesis, we focused on the median nerve for the detection and tracking tasks. In future work, the detection and tracking techniques could be assessed on other types of nerves in order to improve the performance. The use of a larger database should significantly improve the detection and tracking tasks.

  2. Merging the computer vision methods (detection and tracking) with the robotic methods (needle control and visual servoing) for evaluating the more complete system.

  3. Although the robotic control of the needle holder robot to avoid risky regions is introduced, the method should be tested on a two-robot system with shared workspaces and frames. The communication between the probe and needle holder robots is a crucial step for obtaining the needle and the image region position in the 2D plane.

It is not easy to predict what UGRA procedure will be like ten or twenty years from now, but it is reasonable to believe that computer-assisted surgery alongside the robotic aids will keep on improving and providing smarter, more accurate, and surly a safer assistance tools to the anesthetists.

List of publications

- Mohammad Alkhatib, Adel Hafiane, Omar Tahri, Pierre Vieyres, Alain Delbos 2018. «Adaptive median binary patterns for fully automatic nerves tracking in ultrasound images» Computer Methods and Programs in Biomedicine , 160, pp.129-140

- Mohammad Alkhatib, Adel Hafiane 2019. «Robust Adaptive Median Binary Pattern for Noisy Texture Classification and Retrieval» IEEE Transactions on Image Processing , 28, pp.5407-5418

- Mohammad Alkhatib, Adel Hafiane, Pierre Vieyres, Alain Delbos 2019. «Deep visual nerve tracking in ultrasound images» Computerized Medical Imaging and Graphics , 76, pp.101639

- Mohammad Alkhatib, Cyril Novales, Laurence Nouaille, Adel Hafiane,and Pierre Vieyres. (2020, June). «Remote Center of Motion for Redundant Robotic-Assisted Ultrasound Guided Regional Anesthesia». In International Conference on Robotics in Alpe-Adria Danube Region (pp. 85-92). Springer, Cham.

- Mohammad Alkhatib, Adel Hafiane, and Pierre Vieyres. «Merged 1D-2D deep convolutional neural networks for nerve detection in ultrasound images» [accepted ICPR]

Chapter 5

Résumé

Outline:
Dans ce chapitre, un résumé complet de la thèse est présenté en langue française.

5.1 Introduction

L'anesthésie régionale (AR) est pratiquée pour bloquer la sensation de douleur dans une région spécifique du corps en arrêtant la connexion des impulsions nerveuses entre cette région et le système nerveux central [12]. De nos jours, l'AR est une procédure bien connue dans de nombreuses salles d'opération et est utilisée pour éviter de pratiquer une anesthésie générale. L'AR est pratiquée par l'anesthésiste afin de réduire la douleur, de faciliter une sortie plus précoce de l'hôpital et d'améliorer la mobilité postopératoire. Traditionnellement, cette technique était pratiquée en aveugle, ce qui augmentait le risque de défaillance du bloc, de traumatisme nerveux et de toxicité de l'anesthésie locale. Par conséquent, l'anesthésie régionale guidée par ultrasons (UGRA) devient une technique puissante dans la procédure chirurgicale et la gestion de la douleur [23].

L'UGRA a un impact considérable sur la pratique de l'anesthésie régionale [20]. Toutefois, lorsqu'ils pratiquent l'UGRA, les experts sont confrontés à plusieurs défis: trouver l'emplacement du nerf dans l'image échographique de mauvaise qualité; conserver la visualisation du nerf en permanence; localiser les différents organes dans l'image tels que les artères et les os; trouver le meilleur point d'insertion de l'aiguille; contrôler l'insertion de l'aiguille en tenant compte du point de pivot d'interaction entre l'aiguille et la peau du patient; aligner l'aiguille dans le plan échographique 2D; trouver la trajectoire appropriée de l'aiguille pour éviter les régions anatomiques ou cardiovasculaires sensibles (artères, os, etc.). Par conséquent, la procédure UGRA nécessite un long processus d'apprentissage et des années d'expérience [35, 36].

Les systèmes robotisés sont déjà utilisés dans le milieu médical et ont montré leur impact important pour assister les experts médicaux lors de la réalisation de leur acte médical [7]. L'UGRA assistée par robot pourrait être d'une grande utilité en aidant les experts avec des techniques et des outils qui améliorent la précision et la sécurité de la procédure, par exemple en évitant les traumatismes nerveux ou l'endommagement des tissus sains. De plus, cette approche permettrait d'améliorer la courbe d'apprentissage pour les anesthésistes juniors, en évaluant les re-jeux des gestes réalisés (trajectoires, point d'insertion). L'objectif principal du travail de thèse est de fournir aux anesthésistes des outils robotiques et de traitement d'images pour les aider et faciliter leur routine UGRA pour le bien de la santé des patients.

Le système UGRA assisté par robot présente deux aspects essentiels: la qualité visuelle de l'image UltraSonore (US) qui souffre de la présence de bruits; et le contrôle robotique et la précision de l'insertion de l'aiguille. La thèse vise à développer un système UGRA robotisé qui se concentre sur ces deux questions et propose des méthodes de détection et de suivi des nerfs et le contrôle robotisé de l'insertion de l'aiguille.

Cette thèse a été réalisée au sein du Laboratoire Pluridisciplinaire de Recherche en Ingénierie des Systèmes, Mécanique, Energétique (PRISME), un laboratoire de l'Université d'Orléans et de l'INSA-CVL. Cette thèse a été financée par le projet DANIEAL2 'Détection et Analyse des Nerfs dans les Images Echographiques pour l'Anesthésie Locorégionale'. L'Université d'Orléans, l'Université de Tours, l'hôpital Médipôle Garonne (Toulouse) et la société Adechotech ont développé une collaboration autour du projet DANIEAL2 avec l'aide financière de la région Centre Val-de-Loire.

Le projet DANIEAL phase 2 est issu de l'APR 2016 de la région Centre-Val de Loire. Il se situe dans la continuité du projet DANIEAL. Le contexte général du projet est celui du développement de dispositifs médicaux robotisés pour l'amélioration de l'accès aux soins et de l'assistance au geste médical. Le domaine applicatif visé est celui de l'anesthésie locorégionale réalisée sous échographie. Ce projet vise à fournir aux anesthésistes une plateforme collaborative expert-robot-environnement qui améliore de façon significative la pratique de l'anesthésie locorégionale.

L'objectif de DANIEAL2 est d'étudier et de développer un dispositif médical robotisé à sécurité intrinsèque, où l'expert est toujours dans la boucle de contrôle. Il s'agit d'assister le geste médical d'insertion d'aiguille vers le nerf sous imagerie US en utilisant la collaboration d'un bras robotisé. Cette plate-forme d'assistance sera conçue pour faciliter la prestation de soins médicaux de haute qualité et personnalisés ainsi que pour améliorer le processus de formation des anesthésistes et réduire les risques de traumatismes liés à l'anesthésie locorégionale.

Le consortium est composé, des laboratoires PRISME qui a une expertise reconnue en robotique médicale, en traitement d'images et en asservissement visuel, du Laboratoire LIFAT de Tours pour son expertise en analyse d'image video, de la société ADECHOTECH qui développe depuis plusieurs années des robots à finalités médicales et de la clinique Médipôle à Toulouse qui possède une notoriété nationale et internationale pour la pratique de l'anesthésie locorégionale guidée sous échographie.

le projet DANIEAL2 vise à fournir aux anesthésistes une plateforme collaborative expert-robot-environnement qui améliore considérablement la pratique de la pratique de l'anesthésie régionale guidée par ultrasons (Ultrasound-guided regional anesthesia UGRA); c'est-à-dire étudier et développer un dispositif médical robotisé à sécurité intrinsèque, où l'expert est toujours dans la boucle de contrôle. Il s'agit d'assister le geste médical lors de l'insertion de l'aiguille vers le nerf ciblé sous échographie (Ultrasound - US) en utilisant la collaboration d'un bras robotique.

Le système robotique proposé conçu pour la procédure UGRA est divisé en deux systèmes robotiques connectés, un robot porte-sonde (Probe Holder Robot (PHR)) et un robot porte-aiguille (Needle Holder Robot (NHR)), comme illustré à la Figure ci-dessous.

- **Partie PHR**: contrôle le PHR par asservissement visuel pour maintenir sans surveillance les régions d'intérêt souhaitées (par exemple le nerf) dans l'image ultrasonore 2D obtenue par l'appareil à ultrasons . Ensuite, une détection et un suivi automatiques du nerf, ainsi qu'une technique d'asservissement visuel sont effectués pour maintenir le nerf ciblé dans le plan de l'échographie 2D même en cas de mouvements physiologiques. De plus, le contrôle du PHR par asservissement visuel permet de maintenir le nerf ciblé et l'aiguille insérée dans

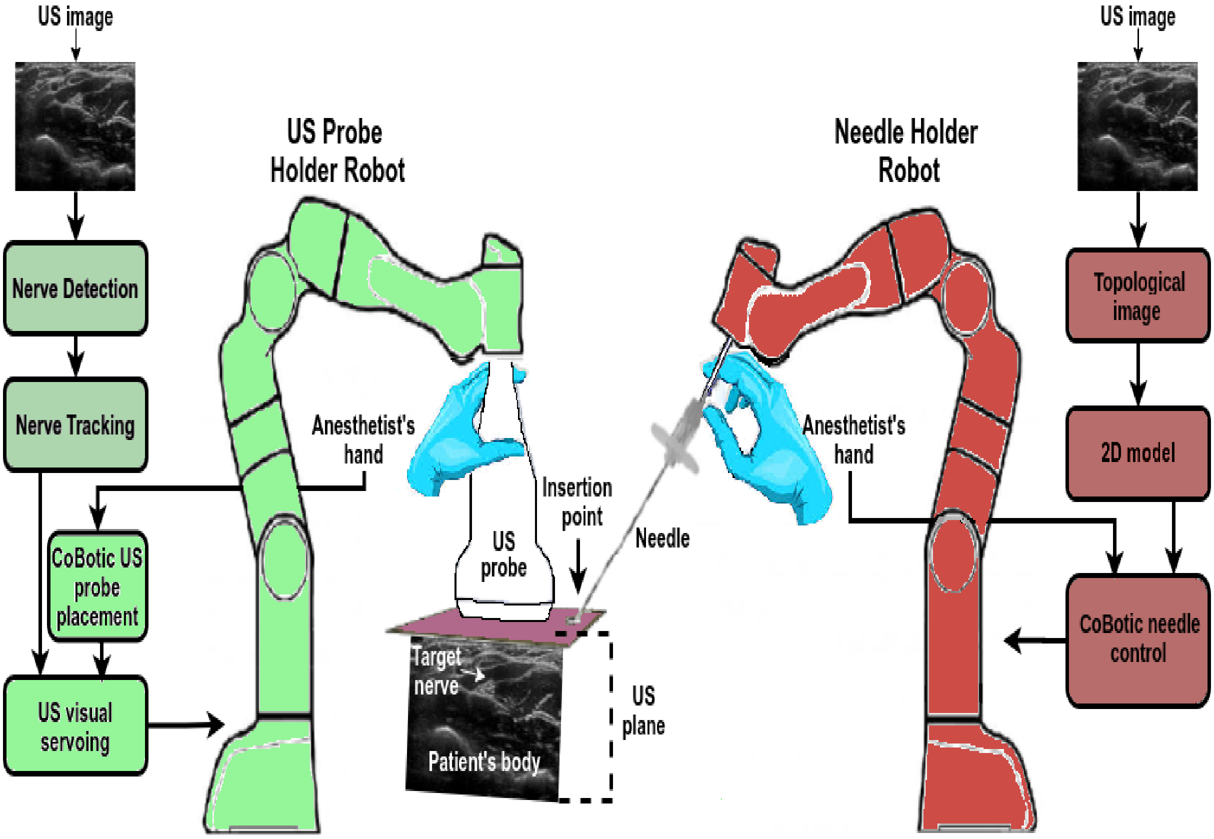


Figure 5.1: La procédure d'UGRA assistée par robot proposée.

le plan de l'image échographique 2D en utilisant une approche d'asservissement visuel par ultrasons. Pour mener à bien cette action, nous devons suivre le nerf dans les images US et récupérer la position/orientation de la pointe de l'aiguille en utilisant les transformations géométriques entre les deux robots. Ainsi, lorsque l'anesthésiste incline l'aiguille en dehors du plan ultrasonore, le PHR doit suivre la pointe de l'aiguille et la maintenir dans le plan ultrasonore.

- **Partie NHR**: lorsque l'aiguille est hors du corps du patient (processus dit de "contrôle externe du patient"), l'objectif de l'anesthésiste est de trouver un point d'insertion de l'aiguille sur la peau du patient qui pourrait fournir la meilleure trajectoire afin d'atteindre le nerf. Mais l'expert est confronté à deux contraintes : le point d'insertion doit se trouver dans le plan des ultrasons (à noter que l'aiguille ne peut apparaître sur l'image US tant qu'elle n'est pas à l'intérieur du corps du patient) ; la pose de l'aiguille est limitée par la nécessité d'être vu dans le plan des ultrasons. Connaissant les transformations géométriques du PHR et du NHR, nous pouvons offrir à l'anesthésiste un contrôle CoBotique du NHR qui respecte les deux contraintes avec un retour d'effort adapté sur les 6-Dimensions. Enfin, le NHR CoBotique assure la visualisation de l'aiguille dans le plan ultrasonore. Lorsque l'aiguille est insérée dans le corps du patient (processus appelé "contrôle interne du patient"), l'objectif de l'anesthésiste est de déplacer l'aiguille aussi près que possible du nerf en respectant les contraintes du centre de mouvement à distance (Remote Center of Motion - RCM). Le NHR est contrôlé CoBotiquement pour manipuler l'aiguille en réduisant le DDL autour du point RCM. Afin que l'anesthésiste puisse éviter les régions à risque telles que l'artère, la veine ou le tendon, une image topologique, basée sur l'image US, est traitée en ligne pour construire un modèle 2D (maillage) qui permet une interaction entre les différents tissus et la pointe et la tige de l'aiguille. Ce modèle permet au NHR d'appliquer des forces de répulsion une fois que l'aiguille se trouve à proximité des régions à risque. De plus, avec le modèle 2D, nous pouvons proposer à l'anesthésiste des trajectoires sûres pour atteindre la zone nerveuse.

En conclusion, le système robotique est divisé en plusieurs tâches :

- Partie PHR :
  - Détection des nerfs
  - Suivi des nerfs
  - Asservissement visuel pour maintenir la visualisation du nerf et de la pointe de l'aiguille
- Partie NHR :
  - Segmentation de l'image : image topologique
  - Modèle 2D : création et mise à jour du maillage
  - Contrôle cobotique sous contraintes RCM
  - NHR Contrôle CoBotique pour l'insertion d'aiguilles sous contraintes RCM avec évitement des régions à risque.

Dans cette thèse, nous nous sommes concentrés sur certains aspects (mis en évidence dans la figure 5.1) pour lesquels nous proposons de nouvelles techniques et méthodes. Pour les autres aspects, comme l'asservissement visuel, cela fera partie de nos travaux futurs. Globalement, les travaux menés au cours de cette thèse conduisent à trois contributions principales :

- La première contribution consiste à proposer un outil qui consiste à traiter les informations (caractéristiques) d'une image US afin de détecter et de suivre automatiquement les blocs nerveux. Ainsi, pour faciliter le processus de détection et de suivi, l'objet cible doit pouvoir être différencié des autres objets à l'aide des caractéristiques de l'image. Néanmoins, le choix et l'extraction de caractéristiques pertinentes est une tâche complexe en raison de l'apparence de la texture nerveuse et de la nature bruyante de ce type d'image. Dans la section 2, nous abordons ces critères en introduisant un nouveau descripteur de texture, le Robust Adaptive Median Binary Pattern (RAMBP). Ce descripteur est évalué pour des textures très bruyantes car le choix de la caractéristique robuste appropriée est une étape cruciale dans les applications de détection et de suivi. Cette contribution est présentée dans la section 2.

- La deuxième contribution consiste à proposer différentes techniques de détection et de suivi des nerfs dans l'image US. Ces techniques sont basées sur : des descripteurs de texture (RAMBP) ; des modèles CNNs ; et un nouveau modèle CNNs fusionné (RAMBP avec modèle CNNs). Les résultats Les résultats de la détection et du suivi des nerfs seront utilisés comme une entrée pour l'asservissement visuel. Ainsi, Cela faciliterait la routine quotidienne de l'anesthésiste et lui permettrait de se concentrer sur l'insertion de l'aiguille et l'administration de l'anesthésique. l'insertion de l'aiguille et l'administration de l'anesthésique. Cette contribution est présentée dans la section 3.

- La troisième contribution propose un système de contrôle robotique pour l'insertion des aiguilles. La stratégie de commande robotique utilise la "force" et de l'"endurance" des robots, ainsi que de la flexibilité et de la prise de décision de l'anesthésiste, pour faciliter la réalisation de la procédure UGRA. Ce système offre une solution sûre et flexible pour faciliter la réalisation de la procédure UGRA dans un espace partagé entre le personnel médical et le ssytème robotisé. De plus, ce système permet aux experts de contrôler le NHR pour une meilleure précision de l'insertion de l'aiguille. Cette contribution est présentée dans la section 4.

Les sections suivantes présentent des résumés de chaque contribution. Dans la section 5.2, RAMBP est présenté. Dans la section 5.3, les techniques de détection et de suivi proposées sont proposées. La section 5.4 présente le contrôle robotique. Ce chapitre se termine par des conclusions et perspectives finales.

5.2 Extraction de la caractéristique robuste

Les problèmes de détection et de suivi dans l'analyse des images et des signaux nécessitent la prise en compte d'informations complexes intégrées dans les données. Les images peuvent contenir plusieurs milliers de valeurs de pixels qui représentent différents objets. Même si les humains peuvent traiter à la fois des objets physiques et des notions abstraites dans leurs activités quotidiennes tout en prenant des décisions dans diverses situations, il n'est pas possible pour l'ordinateur de les traiter directement.

Une fonction doit contenir les informations nécessaires pour distinguer les classes, être insensible à la variabilité non pertinente de l'entrée [72]. De plus, les fonctions doivent être limitées en nombre, afin de permettre un calcul efficace des fonctions discriminantes et de limiter la quantité de données de formation requises. Les caractéristiques de l'image, telles que les bords et les points d'intérêt, fournissent de riches informations sur le contenu de l'image. Elles correspondent à des régions locales de l'image et sont fondamentales dans de nombreuses applications de l'analyse d'images, telles que la reconnaissance, la correspondance, la reconstruction, etc [72]. Un descripteur de caractéristiques est un algorithme qui prend une image et produit des descripteurs et des vecteurs de caractéristiques. Les descripteurs de caractéristiques encodent des informations intéressantes en une série de nombres et agissent comme une sorte d'"empreinte" numérique qui peut être utilisée pour différencier une caractéristique d'une autre. Les descripteurs de caractéristiques contribuent à améliorer l'efficacité de nombreuses tâches telles que la détection et le suivi. Pour cela, le choix d'une caractéristique appropriée est une étape cruciale dans les applications de détection et de suivi.

La texture est un indice important pour la détection dans de nombreuses applications d'images US [73], car certains tissus présentent des propriétés de texture bruyante spécifiques, comme certains nerfs. Pour cette contribution, nous proposons un nouveau et robuste descripteur de texture "Robust Adaptive Median Binary Patterns" (RAMBP). L'image échographique étant considérée comme une image bruyante avec des régions de texture, nous évaluons le descripteur proposé sur des textures bruyantes bien connues pour les problèmes de classification des textures.

RAMBP utilise une classification des pixels bruyants, une fenêtre adaptative pour le seuil et les modules binaires, et des valeurs régionales au lieu d'utiliser les intensités des pixels. RAMBP est divisé en trois étapes, le processus de classification de la détection des pixels bruyants, le processus de seuil et la génération du modèle binaire. Exemple de classification des textures, l'utilisation de RAMBP donne les meilleurs résultats et surpasse les autres méthodes de l'état de l'art.

Le descripteur RAMBP tire parti de la classification des pixels et de l'analyse adaptative pour offrir de fortes propriétés de discrimination et de résistance au bruit. Le descripteur proposé a été évalué sur des textures bruyantes, notamment le sel et le poivre, le bruit gaussien, le flou gaussien et le bruit mixte. Les résultats expérimentaux ont indiqué que RAMBP surpasse les autres descripteurs existants pour le traitement de la classification des textures très bruyantes et qu'il est l'un des meilleurs dans la classification des textures sans bruit. Dans ce travail, la recherche de textures bruyantes est présentée de la cohérence et de la stabilité de la politique de RAMBP, et il montre la haute performance et la robustesse de RAMBP. Dans ce document, la complexité de calcul avait été testé où le RAMBP offre une bonne durée de fonctionnement et un la dimensionnalité des caractéristiques.

Le descripteur RAMBP sera utilisé et évalué dans la prochaine contribution pour la détection et le suivi dans les images US.

5.3 Détection et suivi des nerfs

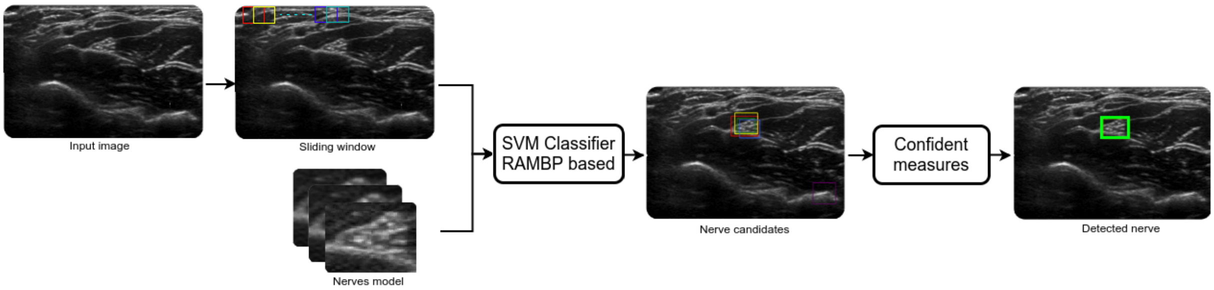
La procédure UGRA nécessite des connaissances anatomiques et des compétences dans l'utilisation de l'imagerie US, ainsi qu'un haut degré de concentration de la part de l'anesthésiste pour localiser le nerf [5, 6]. En outre, il n'est pas toujours facile de localiser certains nerfs dans la pratique de l'UGRA. Dans ce travail, un système UGRA assisté par robot est proposé. Ce chapitre vise à développer des outils pour aider les anesthésistes à détecter et à suivre les nerfs avec précision.

Bien que les algorithmes de détection et de suivi des images échographiques aient été largement développés [122–125], le problème reste ouvert, en particulier pour l'anesthésie régionale. Jusqu'à présent, très peu d'attention a été accordée à la détection et au suivi des nerfs. Dans le chapitre précédent, nous avons proposé des modèles binaires médians adaptatifs robustes (RAMBP) qui présentent les avantages de l'efficacité et de la robustesse aux textures avec différents bruits élevés. L'utilisation de détecteurs ou de trackers basés sur RAMBP offrirait une bonne performance, puisque les régions nerveuses sont considérées comme une texture bruyante. Cependant, nous pensons que les modèles de réseaux neuronaux convolutionnels (CNN) sont également efficaces pour les textures d'images US. Ainsi, pour apprendre des caractéristiques plus optimales, il est plus intéressant de fusionner les modèles RAMBP et CNN afin d'obtenir de meilleures performances pour la détection et le suivi des nerfs.

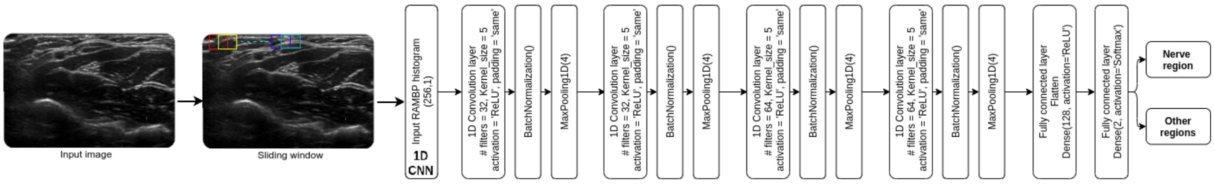
Pour cette contribution, nous proposons différentes méthodes pour détecter et suivre le nerf dans les images US. Ces méthodes sont basées sur le descripteur de texture (RAMBP), les modèles CNN et Support Vector Machine (SVM) et le modèle fusionné proposé (descripteur de texture avec CNN).

5.3.0.0.a Détection des nerfs basé sur le descripteur RAMBP

Nous présentons la détection des nerfs à l'aide du descripteur RAMBP en utilisant deux techniques, la machine à vecteur de support (SVM) [140] et le CNN 1D [132]. Bien que le CNN 1D soit considéré comme un modèle CNN, dans cette section, nous utilisons le descripteur RAMBP comme entrée pour le CNN 1D. La figure 5.2 illustre la chaîne de traitement complète des deux techniques.



(a) SVM



(b) 1D-CNN

Figure 5.2: Schéma de la détection des nerfs à l'aide du descripteur RAMBP. (a) Classificateur SVM. (b) Classificateur CNN 1D.

Comme l'illustre la Figure 5.2, plusieurs images US ont été utilisées comme ensemble d'apprentissage afin de représenter différents aspects du nerf. Pour détecter le nerf, les modèles SVM et 1D CNN sont utilisés pour comparer les fenêtres coulissantes de l'image d'entrée (test) et les modèles. Pour le SVM, la procédure de classification génère une liste de positions de régions candidates pour chaque nerf. A partir de ces positions, la position finale du nerf est assignée à la région où le niveau de confiance est le plus élevé comme le montre la Figure 5.2a. De l'autre côté, pour le CNN 1D, la sortie de la procédure de classification est le score de prédiction de chaque fenêtre glissante dans l'image. À partir de ces scores, la position finale du nerf est attribuée à la région ayant le score le plus élevé.

5.3.0.0.b Détection des nerfs basé sur le modèle CNN

Nous présentons la détection des nerfs à l'aide du l'architecture CNN 2D est construite pour apprendre des caractéristiques visuelles profondes à partir d'images US. Comme l'illustre la Figure 5.3, plusieurs images US ont été utilisées comme ensemble d'apprentissage afin de représenter différents aspects du nerf. Pour détecter le nerf, les modèles CNN 2D sont utilisés pour comparer les fenêtres coulissantes de l'image d'entrée (test) et les modèles. Le résultat de la procédure de classification est le score de prédiction de chaque fenêtre coulissante dans l'image. À partir de ces scores, la position finale du nerf est attribuée à la région ayant le score le plus élevé.

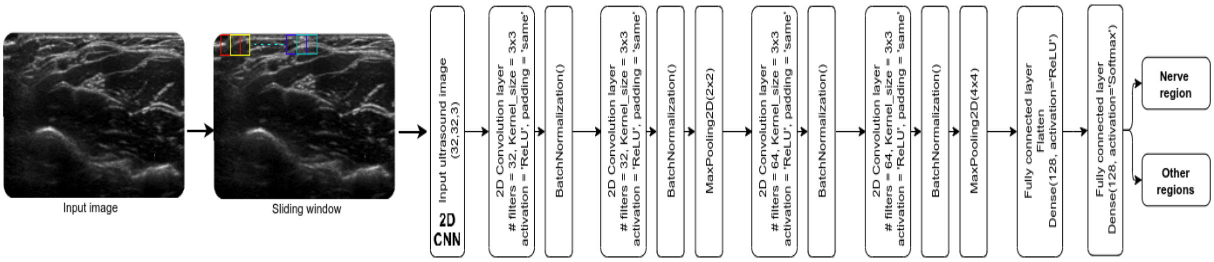


Figure 5.3: Schéma de l'architecture CNN 2D.

5.3.0.0.c Détection des nerfs basé sur le modèle fusionné

Malgré les fortes caractéristiques des descripteurs de texture ou des détecteurs basés sur le CNN, le sujet de la détection des nerfs nécessite un développement et un examen plus poussés. Ainsi, pour apprendre caractéristiques optimales, nous proposons d'apprendre des caractéristiques de haut niveau à partir de données de dimensions différentes en utilisant des CNN de différentes dimensions. L'architecture CNN profonde fusionnée comporte deux branches, une branche CNN unidimensionnelle (1D) (histogrammes RAMBP 1D) et une branche CNN 2D (images US 2D), comme le montre la figure 5.4 images US), comme le montre la figure 5.4. Ces deux architectures ont pour but d'apprendre des caractéristiques profondes et de les concaténer pour former le réseau profond fusionné.

5.3.0.0.d Suivi des nerfs basé sur le descripteur RAMBP

Comme nous traitons des régions de texture dans les images US, il serait plus intéressant d'utiliser des descripteurs robustes. Plusieurs études visent à augmenter la robustesse de la classification des textures bruyantes en utilisant une approche robuste de motifs binaires locaux. La figure5.5 décrit le processus de suivi où les algorithmes de suivi déterministes utilisés sont combinés avec différents types de méthodes d'extraction de caractéristiques. Le processus commence par la prise de la position précédente du nerf et son entrée dans le descripteur de caractéristiques (RAMBP). Il est suivi par l'application du traqueur de caractéristiques. Ces trackers comparent les caractéristiques de l'emplacement précédent du nerf avec les caractéristiques qui ont été prises à partir de l'emplacement du nerf candidat.

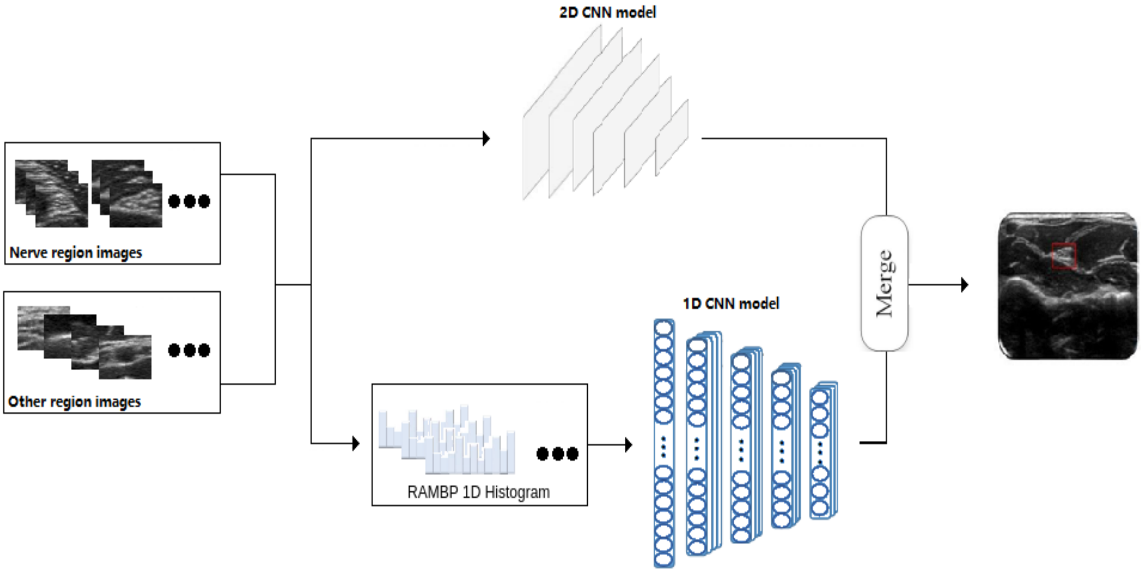


Figure 5.4: Schéma de l'architecture CNN 1D-2D fusionnée.

5.3.0.0.e Suivi des nerfs basé sur les modèles CNNs

Il est plus avantageux d'exploiter les trackers récents basés sur des processus d'apprentissage profond, car ils ont montré d'excellentes performances dans de nombreuses applications de vision par ordinateur [158–169].. Motivés par les percées des CNN, plusieurs trackers basés sur l'apprentissage profond ont été développés afin d'améliorer de manière significative les performances de suivi. Ces travaux de recherche ont donné des résultats prometteurs pour différentes applications de suivi.

Dans ce travail, nous avons mené une étude comparative de treize trackers profonds pour le suivi du nerf dans les images US. Ces trackers sont construits en empilant différentes couches CNN pour suivre le nerf dans les images US. Nous avons mené une enquête sur le suivi des nerfs en utilisant différents trackers CNN tels que traqueur à opérateurs de convolution continus (C-COT) [158], opérateurs de convolution efficaces (ECO) [159], etc.

5.3.0.0.f Suivi des nerfs basé sur le modèle fusionné 1D-2D

Comme la section 3.2.3 a discuté de la possibilité de fusionner les descripteurs de texture avec le modèle CNNs pour la détection des nerfs, il est intéressant, également, d'utiliser le modèle fusionné pour la tâche de suivi. Ici, nous proposons d'utiliser le modèle CNNs 1D-2D fusionné (Section 3.2.3) avec trois techniques de suivi pour suivre le nerf dans les images US. La première technique est basée sur la détection en utilisant le modèle CNNs fusionné dans la zone de recherche de l'image suivante plus grande que l'emplacement précédent du nerf. La deuxième technique est basée sur le suivi par filtre de particules (PF) [150] où le modèle CNNs fusionné est responsable de fournir les poids des particules. La troisième technique combine les deux premiers trackers.

Comme le montre la figure, le tracker mixte basé sur les CNN fusionnés est une combinaison des deux trackers précédents. Il commence par utiliser le tracker PF basé sur les CNNs fusionnés comme expliqué précédemment. Si le score le plus élevé (la similarité maximale) du poids des

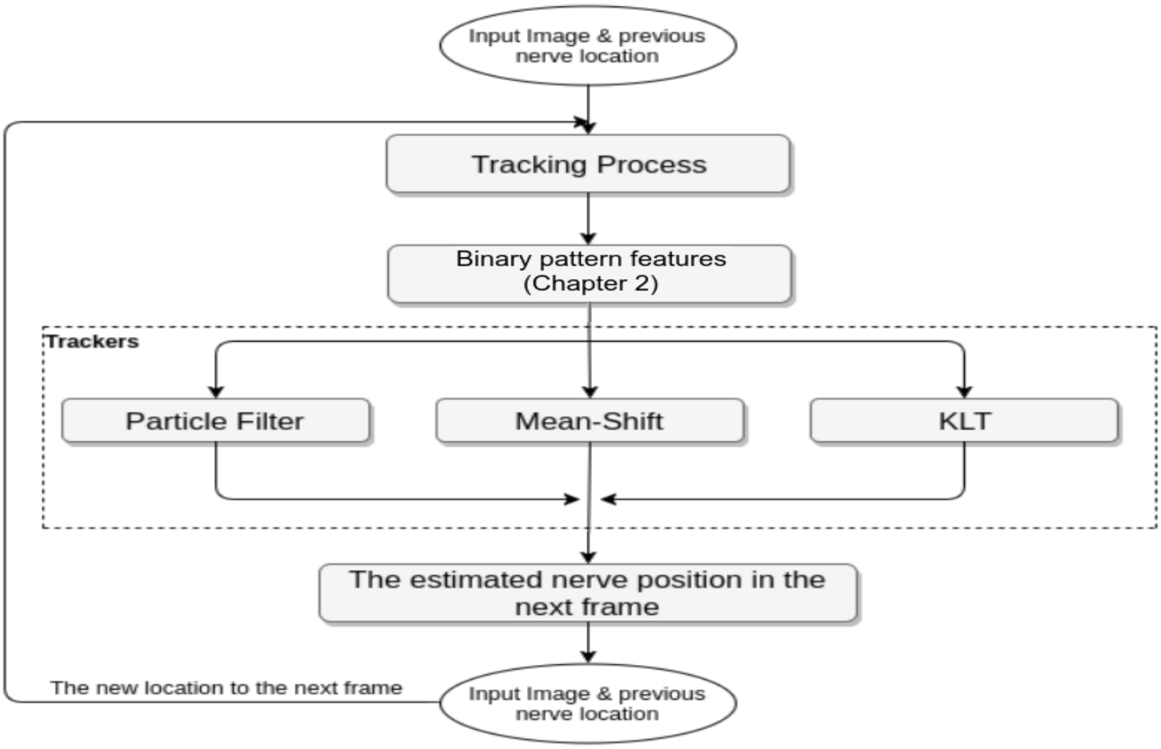


Figure 5.5: Le schéma du processus de suivi basé sur le descripteur RAMBP.

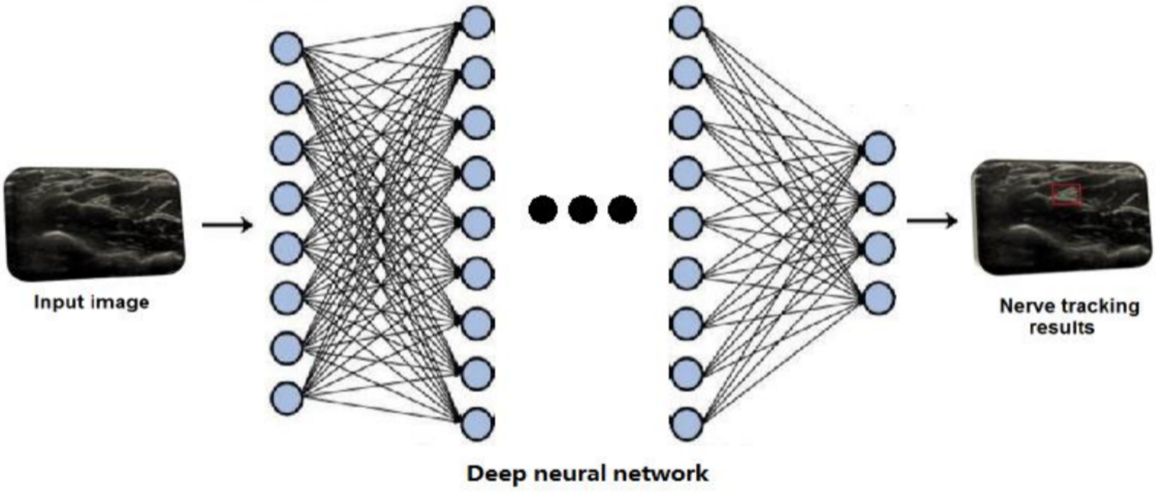


Figure 5.6: Le schéma du processus de suivi basé sur les modèles CNNs.

particules, récupéré à partir du modèle CNN fusionné, est inférieur à un certain seuil, le tracker de recherche basé sur les CNN fusionnés sera déclenché pendant une image pour suivre le nerf plus précisément.

Pour les trois techniques, nous adoptons une stratégie incrémentale pour mettre à jour le modèle CNN fusionné avec les nouvelles apparences du nerf. La stratégie incrémentale n'utilise que les nouveaux échantillons de l'image actuelle pour mettre à jour le modèle. Ainsi, ce modèle mis à jour est capable de s'adapter aux variations d'apparence de la cible tout en atténuant le problème de dérive.

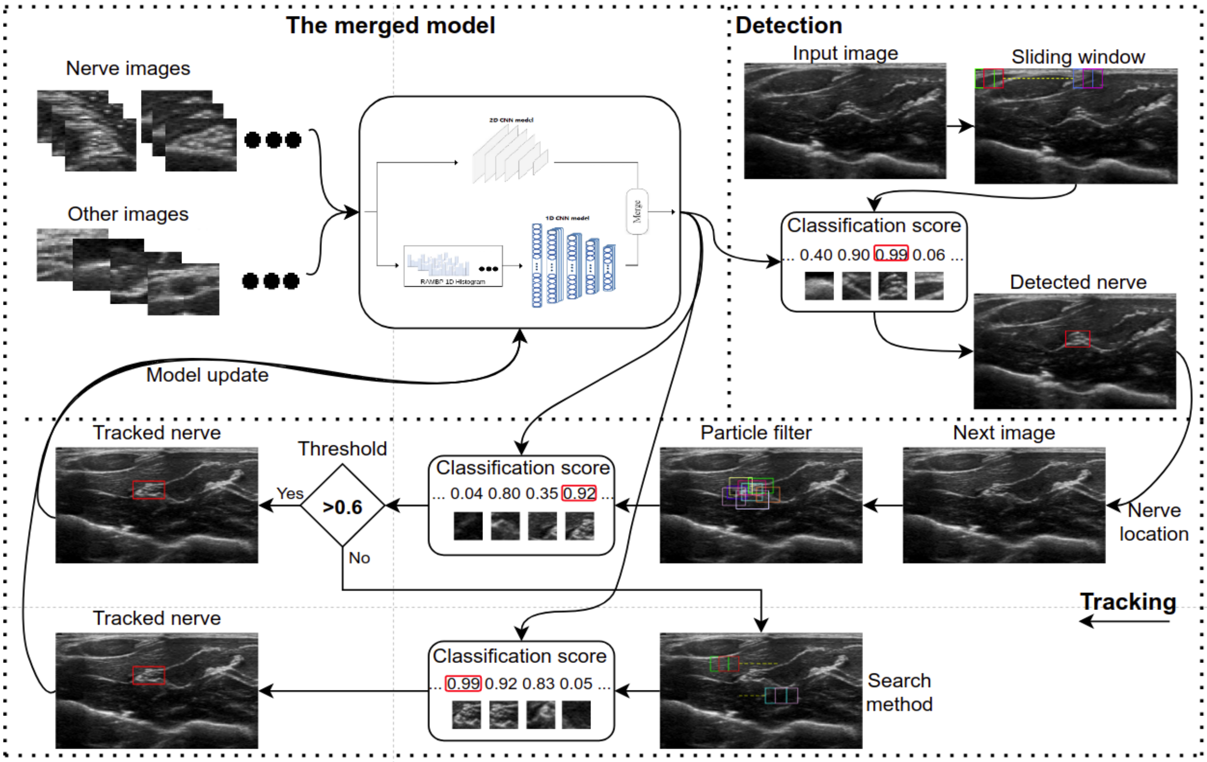


Figure 5.7: Le schéma du processus de suivi basé sur le modèle fusionné 1D-2D.

5.4 Contrôle robotique

L'exécution de la routine UGRA nécessite un long processus d'apprentissage, principalement en raison du risque de traumatisme nerveux et de la mauvaise qualité des images échographiques [5, 6]. C'est pourquoi le projet DANIEAL 2 vise à fournir aux anesthésistes une plateforme collaborative expert-robot-environnement qui améliore considérablement la pratique de l'UGRA. L'objectif de DANIEAL2 est d'étudier et de développer un dispositif médical robotisé à sécurité intrinsèque, où l'expert est toujours dans la boucle de contrôle. Il s'agit d'assister le geste médical d'insertion de l'aiguille vers le nerf sous imagerie US grâce à la collaboration d'un bras robotique.

Cette contribution se concentre sur le contrôle de la position de l'aiguille à l'intérieur du corps du patient. Ce contrôle vise à fixer le point d'interaction entre la peau du patient et l'aiguille à l'endroit où le mouvement de l'aiguille se fera autour de ce point. Le point d'interaction entre l'aiguille et la peau du patient est appelé le point d'insertion. Dans les opérations chirurgicales, il existe une énorme différence entre le terme de point d'insertion et celui de point de trocart dans le domaine médical. Le point de trocart limite les mouvements à seulement quatre degrés de liberté (DDL) : trois DDL rotationnels et un DDL translational. Alors que pour le point d'insertion, la rotation sur l'axe z est bloquée en raison de l'aspect symétrique de l'aiguille. Pour cela, le point d'insertion limite également l'anesthésiste à seulement quatre mouvements de liberté mais avec blocage de la rotation sur l'axe z.

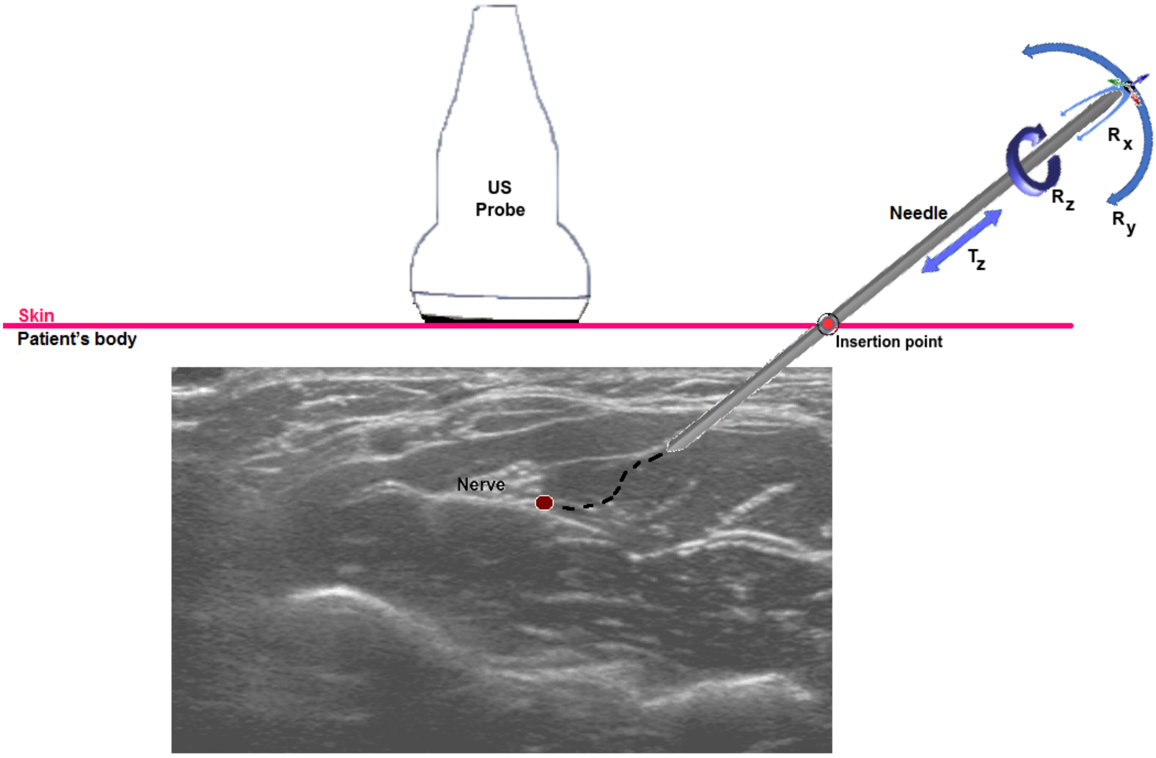


Figure 5.8: Mouvement UGRA autour du point d'insertion (Pinsertion - *RCM*), où le cadre de l'outil est la position et la rotation de l'effecteur final. Par conséquent, l'aiguille ne peut se déplacer que dans les limites de 4-DDL : R_x, R_y et R_z sont les rotations autour des axes x, y et z respectivement, et T_z est la translation le long de l'axe z.

Dans l'UGRA, le point d'insertion constitue une question essentielle car l'anesthésiste doit manipuler une aiguille à l'intérieur du corps humain. Cependant, la manipulation de l'aiguille impose une tâche très difficile, où il est important de s'assurer que l'aiguille se déplace dans les limites du point d'insertion afin d'éviter de blesser le patient. Lors d'une procédure assistée par robot, les mouvements de l'aiguille insérée sont limités à une translation le long de son axe et à des rotations autour du point d'insertion [179]. Le mouvement du manipulateur est ensuite contraint par rapport à ce point sur l'effecteur final appelé centre de mouvement à distance (RCM) [52, 180], comme indiqué ci-dessous.

La section 5.4.1 propose la cinématique d'insertion de l'aiguille (RCM pour l'insertion de l'aiguille) et présente diverses stratégies de contrôle (retour de force haptique et contrôle CoBotique). Nous présentons ensuite le processus de commande CoBotique sous les contraintes RCM avec évitement des régions dans la section 5.4.2.

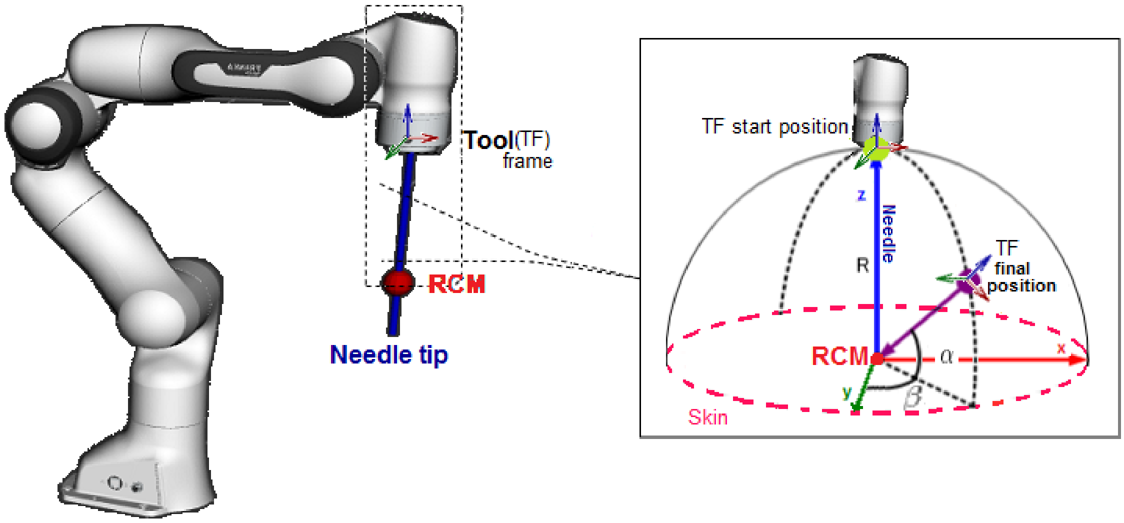


Figure 5.9: Une représentation sphérique de la position de l'effecteur. Où α et β sont les angles de représentation de la sphère et R est le rayon de la sphère.

5.4.1 Commande CoBotique sous les contraintes RCM

La principale contribution de cette section est la formulation d'une méthode de contrôle géométrique dans l'espace opérationnel pour maintenir les contraintes RCM tout en offrant une position plus précise de l'effecteur final. La méthode contrôle le mouvement de l'aiguille rigide et décrit la position de l'effecteur final par rapport aux contraintes du RCM pour un contrôle robotique de l'UGRA à l'aide d'un robot série de 7-DDL. Elle fournit une solution sûre et flexible pour le robot et le personnel médical dans un espace de travail partagé où elle modélise explicitement les mouvements de rotation et de translation au point RCM.

Mouvement UGRA autour du point d'insertion ($P_{insertion}$ - RCM), où le cadre de l'outil est la position et la rotation de l'effecteur final. Par conséquent, l'aiguille ne peut se déplacer que dans un rayon de 4 degrés : R_x, R_y et R_z sont les rotations autour des axes x, y et z respectivement, et T_z est la translation le long de l'axe z.

Pour répondre à ce besoin, nous avons proposé un cadre de contrôle pour l'UGRA assistée par robot pour la collaboration physique homme-robot à l'aide d'un robot manipulateur à 7 degrés de liberté (Franka Emika). Nous avons proposé une méthode géométrique pour calculer la position de l'effecteur final du robot prévu par rapport aux contraintes de la RCM, elle est destinée à aider l'anesthésiste à exécuter un mouvement plus sophistiqué de l'aiguille dans le corps du patient avec une grande précision. Cette précision est définie comme l'erreur de distance entre la position du point RCM et la position du point d'insertion fixe.

Nous avons fourni deux stratégies de contrôle du robot en ce qui concerne les contraintes RCM, les contrôles haptiques et CoBotique d'insertion de l'aiguille. L'action de base de ces stratégies de contrôle d'insertion d'aiguille consiste à déplacer l'aiguille à travers le point d'insertion et à générer un centre de mouvement à distance (RCM) à distance ou CoBotique.

Deux stratégies de contrôle ont été présentées en utilisant le contrôle cartésien du RCM : un système téléopéré d'insertion d'aiguille UGRA avec retour de force haptique ; un système d'insertion d'aiguille CoBotique. La commande téléopérée UGRA permet l'insertion d'une aiguille sous des contraintes RCM avec retour de force appliqué au dispositif haptique (6-DDL Vir-

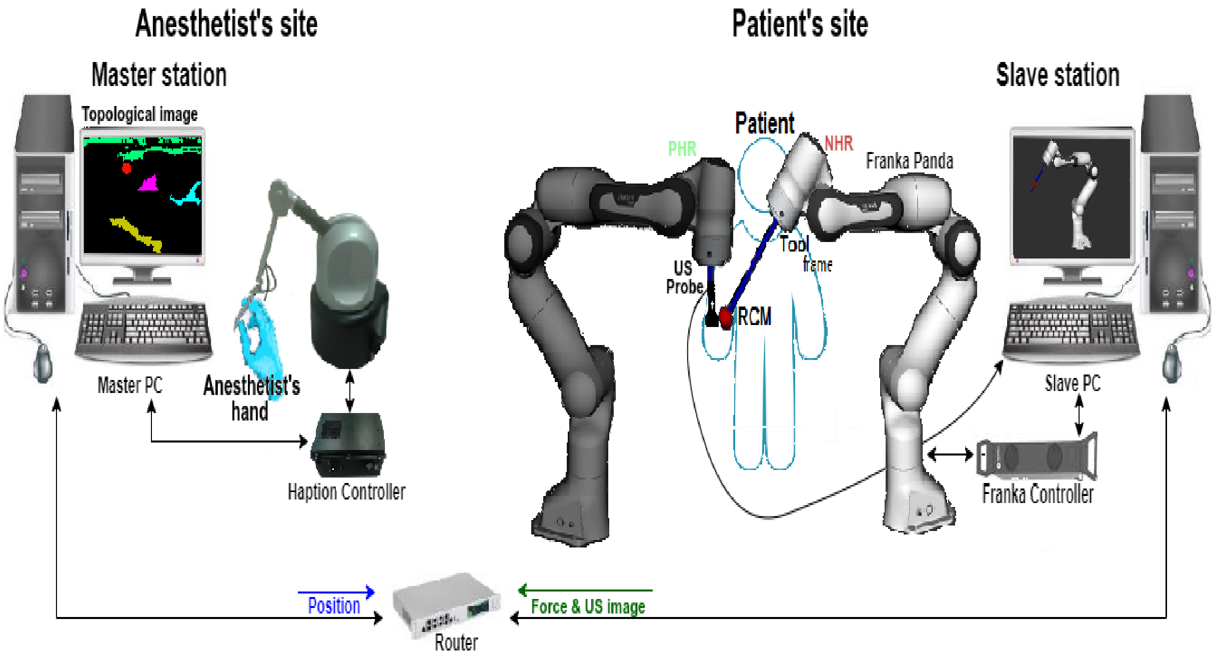


Figure 5.10: Schéma du système UGRA **téléopéré** utilisant un dispositif haptique.

tuose 3D Desktop). La méthode proposée est précise, car elle trouve la position et la rotation exactes de l'effecteur final avec une erreur de moins de 1 mm (erreur de distance euclidienne entre le point RCM et le point d'insertion fixe). Pour évaluer le système de retour de force téléopéré, l'erreur entre les forces réelles appliquées sur l'aiguille et les forces rendues souhaitées appliquées sur la main de l'expert est calculée. Le système de retour de force téléopéré s'est avéré précis et sensible, avec une erreur proche de zéro entre les forces réelles et les forces souhaitées. D'autre part, la commande CoBotique UGRA permet de contrôler les mouvements de l'aiguille en appliquant une force manuelle par l'utilisateur sur le bras du robot tout en respectant le RCM contraintes. Pour évaluer la précision du système, l'erreur de distance euclidienne entre les On calcule le point RCM et le point d'insertion fixe qui est inférieur à 1 mm tout en maintenant la cinématique d'insertion de l'aiguille.

5.4.2 Éviter les régions à risque

L'évitement des régions à risque est une étape importante pour un système robotique UGRA plus complet. L'objectif est de donner à l'anesthésiste une liberté de mouvement lors de l'insertion de l'aiguille avec l'avantage de ressentir une force répulsive supplémentaire pendant que la pointe de l'aiguille se rapproche de toute région à risque comme un nerf, une artère, une veine ou un tendon. En outre, cette procédure ne doit pas limiter l'anesthésiste à un seul trajet vers le nerf cible. Le processus d'évitement des régions à risque commence par la segmentation de l'image échographique pour fournir à l'anesthésiste une image topologique traitée en ligne (réalité augmentée). Cependant, la segmentation d'une image est coûteuse en termes de calcul. Pour cela, un modèle de maillage 2D est construit à partir de l'image topologique, ce qui permet de mettre à jour la localisation des régions dans l'image. Ensuite, de petites forces de répulsion seront appliquées à la main de l'utilisateur lorsque la pointe de l'aiguille s'approchera d'une

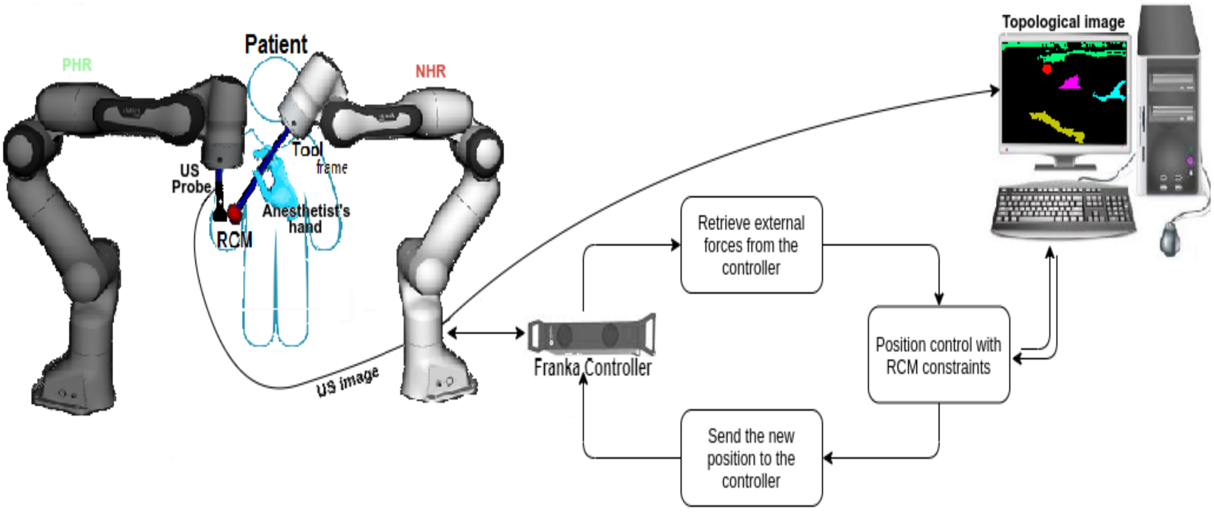


Figure 5.11: Schéma du système **CoBotique** UGRA.

région à risque.

5.4.2.0.a L'image topologique

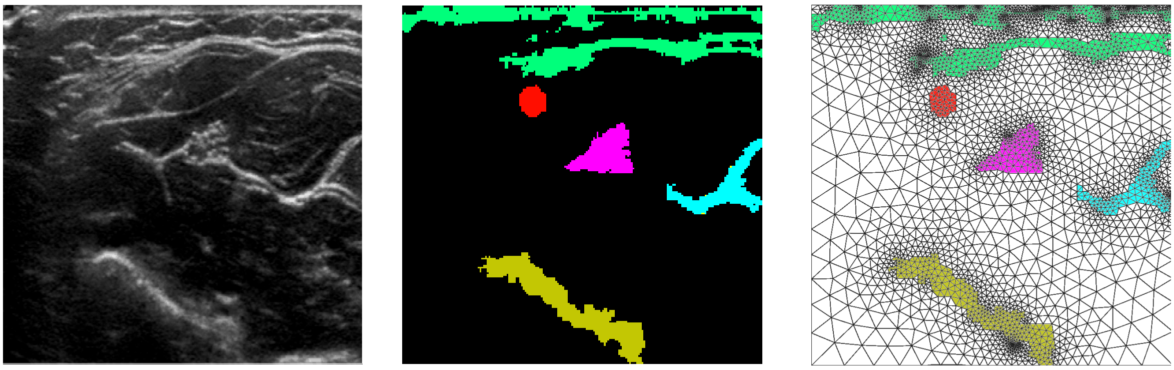
À ce jour, des travaux de recherche ont été développés sur les régions d'intérêt en matière d'images échographiques [131, 138, 199, 200]. Néanmoins, aucune recherche ne se concentre sur la segmentation de tous les organes (régions) dans l'image US tels que les nerfs, les artères, les muscles, les tendons et les os. Par conséquent, le sujet de la segmentation dans les images échographiques nécessite un développement et des recherches plus poussés. L'impossibilité de localiser le nerf et d'autres régions pourrait entraîner un traumatisme nerveux ou une toxicité anesthésique locale. L'objectif de cette section est de fournir aux anesthésistes un outil basé sur le traitement des images échographiques pour traiter cette question et améliorer la pratique de l'UGRA. La segmentation des images US est une tâche difficile car le bruit et d'autres artefacts corrompent les propriétés visuelles de ce type de tissu.

Au cours des dernières années, diverses architectures de réseaux neuronaux convolutifs (CNN) ont été proposées pour segmenter une image. Certaines de ces architectures ont été développées pour segmenter directement l'image entière, comme le SegNet [201], et c'est donc cet outil que nous utiliserons pour segmenter l'image US. L'architecture SegNet est divisée en un réseau d'encodeurs, un réseau de décodeurs et une couche de classification finale au niveau du pixel, comme indiqué ci-dessous. Ces réseaux sont construits en empilant plusieurs couches différentes, les couches essentielles étant les couches convolutives et les couches de regroupement.

La sortie du classificateur soft-max est une image à N canaux de probabilités où N est le nombre de classes (régions). La segmentation prédite correspond à la classe avec la probabilité maximale à chaque pixel.

5.4.2.0.b Le modèle 2D

En contrôle robotique et comme nous l'avons déjà mentionné, toutes les régions de l'image doivent être segmentées afin d'être utilisées dans les régions à risque à éviter. Néanmoins, l'application de techniques de segmentation dans chaque image est coûteuse sur le plan des



(a) The original US image. (b) The topological image. (c) The mesh image.

Figure 5.12: Un exemple de création de maillage 2D.

calculs. L'étude du milieu, et la prise en compte de sa dynamique, dans lequel l'aiguille se déplacera, est une bonne alternative. Étant donné que la sonde à ultrasons est fixe, l'image le sera également, à moins qu'il n'y ait un mouvement des tissus dans le milieu. Ce mouvement est généré par diverses forces dues à l'insertion de l'aiguille dans des tissus déformables, ce qui signifie que certains points de ce milieu, en particulier le nerf à atteindre, se déplaceront sous l'effet des mouvements de l'aiguille dans les tissus. Dans cette étude, nous proposons de mettre à jour la segmentation de l'image topologique toutes les 10 secondes. En attendant et pour réduire les coûts de calcul, des maillages sont obtenus et analysés pour faciliter la suite du calcul.

Le maillage est utilisé pour simplifier les calculs dans un espace continu par discrétisation qui consiste à le décomposer en un ensemble d'éléments finis dans lesquels les calculs sont moins complexes à effectuer. La méthode des éléments finis [202] (FEM) est une méthode de calcul numérique et est utilisée pour effectuer des calculs dans un environnement trop complexe pour les calculs directs. La méthode des éléments finis discrète l'image en un nombre connu d'éléments qui ont une forme géométrique différente. Cela permet de créer le modèle de maillage. Pour conclure, l'utilisation du maillage avec le modèle élastique fournira des emplacements de toutes les régions dans chaque trame US jusqu'à ce que la nouvelle image topologique soit segmentée.

Pour mettre à jour le modèle de maillage lors de l'insertion de l'aiguille, un modèle de comportement doit être associé pour calculer les mouvements de chaque élément du maillage. Pour les images US, il existe différents modèles utilisant les propriétés de déformation pour mettre à jour le maillage lorsqu'il est basé sur la fluidité de l'environnement déformable. Trois types de matériaux sont utilisés pour construire ces modèles, un ressort, un point d'appui et des éléments de frottement coulissants.

- L'élément de frottement glissant représente le comportement plastique (déformation irréversible ou hystérésis). Cependant, dans le cas de notre système, l'image topologique et le maillage seront renouvelés toutes les 10-15 sec. Par conséquent, la prise en compte de ce modèle de déformation irréversible ne sera alors pas utile. Il est donc préférable de prendre un modèle moins complexe, le modèle viscoélastique.

- Le dashpot est un dispositif mécanique qui consiste en un amortisseur qui résiste au mouvement par frottement visqueux. Pour l'UGRA, la viscosité n'a aucun impact puisque la vitesse de l'aiguille pendant l'acte UGRA est lente par rapport à la viscosité du corps linéaire. La vitesse de déformation des tissus sous l'effet de l'aiguille ne sera pas significative et peut même être considérée comme négligeable en la comparant avec le terme d'élasticité. Pour cela, il est préférable de prendre un modèle moins complexe qui est le modèle élastique.

- Le ressort représente le comportement élastique (Hooke's law) du matériau. Dans l'UGRA, les déformations des tissus seront représentées en ayant un comportement purement élastique sous les contraintes imposées par le mouvement de l'aiguille lors de l'insertion.

5.4.2.0.c Contrôle des forces pour éviter les régions à risque

Pour l'évitement des régions à risque, après avoir créé l'image topologique et le modèle 2D, il faut analyser les forces de répulsion. Globalement, trois forces différentes existent pendant l'UGRA assisté par le robot, deux forces externes et une force interne appliquée. Les deux forces externes sont les forces appliquées par l'anesthésiste sur l'effecteur final et les forces récupérées lors de l'interaction de l'aiguille avec les tissus environnants. La force interne est une force de répulsion appliquée sur l'effecteur final pour éviter les régions à risque lors de l'insertion de l'aiguille.

Pour réussir le processus d'évitement des régions à risque, la connexion entre les deux systèmes robotiques, le PHR et le NHR, dans un espace de travail partagé, est essentielle. Cette connexion permettra de localiser la pointe de l'aiguille dans le plan US. En d'autres termes, le PHR fournira l'image US et le NHR calcule l'emplacement de la pointe de l'aiguille toutes les millisecondes. Ainsi, l'utilisation de l'image topologique avec l'emplacement de la pointe de l'aiguille donnera la distance entre la pointe de l'aiguille et chaque région dans l'image US. Ces régions comprennent le nerf, car l'aiguille doit injecter la drogue autour du nerf et non à l'intérieur de celui-ci. Le nerf est donc considéré comme une région à risque, mais avec des forces de répulsion différentes. Tout en se déplaçant sous les contraintes de la RCM comme présenté dans ma thèse de doctorat, la force appliquée pour éviter la région est également soumise aux mêmes contraintes.

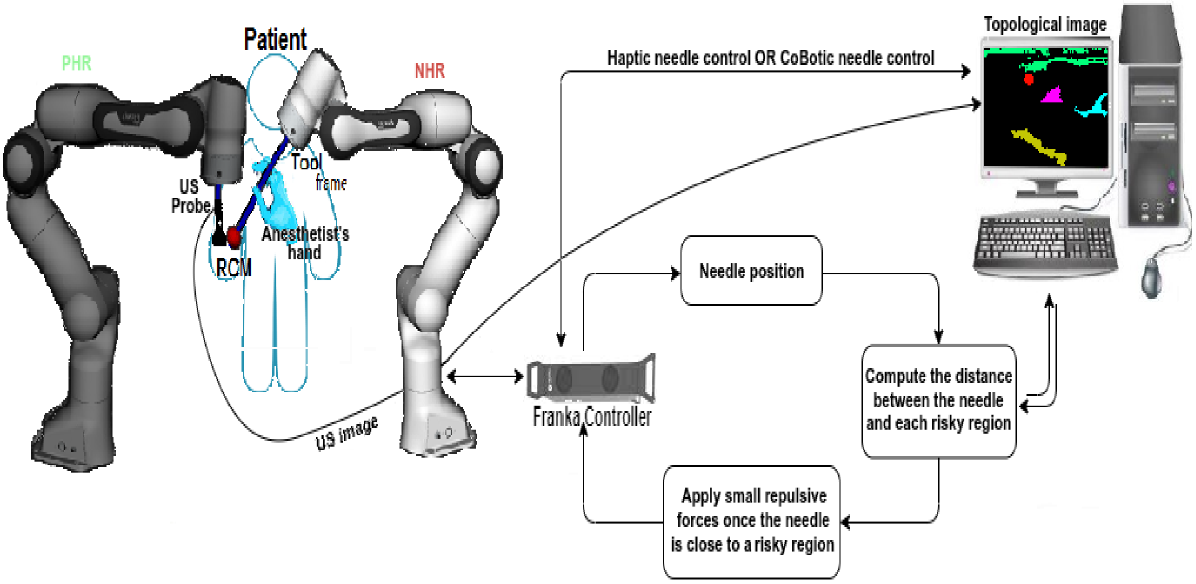


Figure 5.13: Insertion d'une aiguille **CoBotique** sous RCM avec évitement des régions à risque.

Prenons l'image topologique en 2D avec l'emplacement de la pointe de l'aiguille, toutes distances confondues entre la pointe de l'aiguille et les régions de l'image sont calculées. Pour une procédure plus pratique et pour éviter des forces soudaines, les forces doivent être inversement proportionnelles à la distance entre le la pointe de l'aiguille et une zone avec des forces maximales prédéfinies. Ces forces maximales prédéfinies sont fixées comme seuils de sécurité pour éviter d'atteindre des forces extrêmement élevées.

5.5 Conclusions et perspectives

Cette thèse fait partie du projet DANIEAL2 et propose un système robotique complet pour la procédure UGRA où nous nous sommes concentrés sur 3 points principaux : la détection des nerfs, le suivi des nerfs et l'insertion de l'aiguille. Les principales contributions sont résumées comme suit :

- Nous avons proposé un système robotique complet pour la procédure UGRA dans le chapitre 1. Ce système robotisé peut apporter une aide précieuse aux experts grâce à des techniques et des outils qui améliorent la précision et la sécurité de la procédure, notamment en évitant les traumatismes nerveux ou les lésions des tissus sains. En outre, il peut être utilisé comme outil d'apprentissage pour les jeunes professionnels.

- Les problèmes de détection et de suivi des nerfs dans les images échographiques (US) nécessitent de prendre en compte des informations complexes intégrées dans l'image, à savoir les caractéristiques. Toute méthode de détection et de suivi dépend principalement du choix de caractéristiques d'image appropriées. Comme l'image US présente des propriétés de texture bruyantes, nous avons proposé un nouveau descripteur de caractéristiques RAMBP (Robust Adaptive Median Binary Pattern). Le descripteur RAMBP tire parti de la classification des pixels et de l'analyse adaptative pour offrir une forte capacité de discrimination et des propriétés de résistance au bruit. RAMBP a été évalué sur des textures bruyantes comprenant différents types de bruit. Les résultats expérimentaux indiquent que RAMBP surpasse les autres descripteurs existants pour la classification de textures très bruyantes. Cependant, nous pensons que les modèles de réseaux neuronaux convolutionnels (CNN) sont également efficaces pour les textures d'images US. Ainsi, pour apprendre des caractéristiques plus optimales, nous proposons de fusionner la RAMBP avec les modèles CNN afin d'obtenir de meilleures performances pour la détection et le suivi des nerfs.

- Une détection et un suivi précis et cohérents des nerfs sont essentiels pour une UGRA sûre et efficace. Dans cette thèse, les méthodes de détection et de suivi sont basées sur le RAMBP (descripteur de texture), les modèles CNNs, ou le modèle fusionné du RAMBP et du modèle CNNs. Bien que les détecteurs et suiveurs basés sur le descripteur de texture ou les modèles CNNs donnent de bons résultats dans les images US, les résultats montrent que le modèle fusionné a obtenu une précision supérieure de 10%. Il convient de mentionner que le modèle fusionné combine les architectures RAMBP et CNN, ce qui permet de tirer parti de la force de chaque côté et d'améliorer les performances globales. En outre, différentes techniques ont été présentées pour détecter et suivre le nerf dans l'image US. Pour la détection des nerfs, les résultats obtenus montrent que le détecteur basé sur le modèle fusionné surpasse les autres techniques et obtient la meilleure performance avec une précision de 96%. Pour le suivi des nerfs, nous avons constaté que les trackers basés sur le modèle fusionné ont obtenu les meilleurs résultats. Nous pensons que le tracker mixte (filtre de particules et technique de recherche) offre un bon compromis entre la précision du suivi (avec 93%) et la complexité temporelle (avec $0,19$ spf).

- L'insertion d'une aiguille est une tâche très difficile que les anesthésistes rencontrent dans leur routine quotidienne. La manœuvre de l'aiguille autour du point de pivot d'interaction entre l'aiguille et la peau du patient est appelé le point d'insertion. Ce point limite le contrôle de l'aiguille par l'anesthésiste à seulement quatre degrés de liberté (DDL) : trois DDL de rotation et un DDL de translation. En robotique, cette restriction est appelée "contraintes du centre de mouvement distant (RCM)". Nous avons présenté un système expérimental permettant le contrôle cartésien complet de l'effecteur du robot Franka emika sous contraintes RCM. De plus, nous avons présenté deux stratégies de contrôle différentes en utilisant le contrôle RCM proposé : un système d'insertion d'aiguille UGRA téléopéré avec retour de force haptique utilisant un dispositif haptique Virtuose 6-DDL ; un système d'insertion d'aiguille CoBotique en appliquant la main forcée par l'utilisateur opérateur sur le bras du robot Franka emika. Pour évaluer la précision du respect des contraintes RCM, l'erreur de distance euclidienne entre le point RCM et le point d'insertion fixe est calculée. Les deux stratégies de contrôle proposées sont précises pour trouver la position et la rotation exactes de l'effecteur final avec une erreur inférieure à 1 mm tout en respectant les contraintes RCM. Pour évaluer le système de retour de force téléopéré, l'erreur de force entre la force désirée (la force d'insertion appliquée sur l'aiguille au NHR) et la force réelle (force de retour rendue par le dispositif haptique à la main de l'anesthésiste) est calculée. Le système proposé s'est avéré précis et sensible avec une erreur de force proche de zéro ($< 0,1$ N). De plus, dans ce travail, nous avons discuté de l'évitement des régions à risque en appliquant de petites forces répulsives à l'effecteur final lorsque la pointe de l'aiguille est proche d'une région à risque. Pour cela, une image topologique est construite en utilisant une technique de segmentation par réseau neuronal (SegNet). Dans les expériences, les résultats montrent que la technique utilisée a obtenu une précision élevée de 90% pour la segmentation de l'image US.

Un nombre important d'idées restent inexploitées et devront être analysées et étudiées en détail. Les orientations futures prévues pour un éventuel suivi peuvent être décomposées en perspectives à court terme et perspectives à long terme.

- **Perspectives à court terme**
  1. Nous avons présenté le système UGRA assisté par robot, qui est divisé en plusieurs tâches, comme indiqué au chapitre 4. Dans cette thèse, nous avons présenté certaines tâches, tandis que 5 tâches doivent être examinées :
     (a) Contrôle co-botique du robot porte-sonde pour le placement de la sonde sur le corps du patient.
     (b) Asservissement visuel pour contrôler le robot porte-sonde afin de maintenir la visualisation du nerf.
     (c) Contrôle co-botique du robot porte-aiguille pour le placement de l'aiguille dans le plan US.
     (d) Asservissement visuel du robot porte-sonde pour conserver la visualisation du nerf et de l'aiguille dans l'image US.

   (e) Proposition de trajectoires de l'aiguille pour atteindre le nerf.

2. Nous avons effectué les tâches de détection et de suivi sur des vidéos de données réelles. Dans le cadre de travaux futurs, les tâches de détection et de suivi pourraient être effectuées directement en temps réel à l'aide du système US.

3. Pour la commande d'insertion d'aiguille robotisée proposée, une validation technique sur un fantôme anatomique pourrait être effectuée, ce qui serait une étape pour évaluer le système proposé en vue d'applications cliniques.

4. La possibilité d'utiliser la parallélisation sur la plate-forme GPU ou FPGA pour augmenter la vitesse de performance pour les tâches de vision par ordinateur.

- **Perspectives à long terme**

  1. Dans cette thèse, nous nous sommes concentrés sur le nerf médian pour les tâches de détection et de suivi. Dans des travaux futurs, les techniques de détection et de suivi pourraient être évaluées sur d'autres types de nerfs afin d'améliorer les performances. L'utilisation d'une base de données plus importante devrait améliorer de manière significative les tâches de détection et de suivi.
  2. Fusionner les méthodes de vision par ordinateur (détection et suivi) avec les méthodes robotiques (contrôle de l'aiguille et asservissement visuel) pour évaluer un système plus complet.
  3. Bien que le contrôle robotique du robot porte-aiguille pour éviter les zones à risque soit introduit, la méthode doit être testée sur un système à deux robots avec des espaces de travail et des cadres partagés. La communication entre le robot sonde et le robot porte-aiguille est une étape cruciale pour obtenir la position de l'aiguille et de la zone d'image dans le plan 2D.

Bibliography

[1] Ban CH Tsui and Santhanam Suresh. Ultrasound imaging for regional anesthesia in infants, children, and adolescentsa review of current literature and its application in the practice of extremity and trunk blocks. *Anesthesiology: The Journal of the American Society of Anesthesiologists*, 112(2):473–492, 2010.

[2] Peter Marhofer and Vincent WS Chan. Ultrasound-guided regional anesthesia: current concepts and future trends. *Anesthesia & Analgesia*, 104(5):1265–1269, 2007.

[3] Sébastien Bloc, Olivier Rontes, Luc Mercadal, and Alain Delbos. Block success rate: a question of target and definition. *Regional anesthesia and pain medicine*, 38(6):553, 2013.

[4] Andrew T Gray. Ultrasound-guided regional anesthesiacurrent state of the art. *Anesthesiology: The Journal of the American Society of Anesthesiologists*, 104(2):368–373, 2006.

[5] Peter Marhofer, Harald Willschke, and Stephan Kettner. Current concepts and future trends in ultrasound-guided regional anesthesia. *Current Opinion in Anesthesiology*, 23(5):632–636, 2010.

[6] Glenn E Woodworth, Elliza M Chen, Jean-Louis E Horn, and Michael F Aziz. Efficacy of computer-based video and simulation in ultrasound-guided regional anesthesia training. *Journal of Clinical Anesthesia*, 26(3):212–221, 2014.

[7] Anthony R Lanfranco, Andres E Castellanos, Jaydev P Desai, and William C Meyers. Robotic surgery: a current perspective. *Annals of surgery*, 239(1):14, 2004.

[8] Tobias Ortmaier, Barbara Deml, Bernhard Kübler, Georg Passig, Detlef Reintsema, and Ulrich Seibold. Robot assisted force feedback surgery. In *Advances in Telerobotics*, pages 361–379. Springer, 2007.

[9] Fernande Lois and Marc De Kock. Does regional anesthesia improve long-term patient outcome? *Techniques in Regional Anesthesia and Pain Management*, 12(4):203–208, 2008.

[10] Jesús Calatayud and Ángel González. History of the development and evolution of local anesthesia since the coca leaf. *Anesthesiology: The Journal of the American Society of Anesthesiologists*, 98(6):1503–1508, 2003.

[11] J Nathan, L Asadourian, and MA Erlich. A brief history of local anesthesia. *Int J Head Neck Surg*, 7(1):29–32, 2016.

[12] Sukhminder Jit Singh Bajwa and Ashish Kulshrestha. Anaesthesia for laparoscopic surgery: General vs regional anaesthesia. *Journal of minimal access surgery*, 12(1):4, 2016.

[13] Paolo Busoni. Central or peripheral blocks? *Techniques in Regional Anesthesia and Pain Management*, 6(3):95–98, 2002.

[14] Jürgen K Mai and George Paxinos. *The human nervous system.* Academic press, 2011.

[15] Jorge Hernando Sáez, Carlos Tornero Tornero, Vicente Roqués Escolar, Francisco Hernández Méndez, and Luis Aliaga Font. Development of complications in ultrasound-guided regional anesthesia vs neurostimulation. *Techniques in Regional Anesthesia and Pain Management*, 16(3):152–157, 2012.

[16] Sanjay K Sinha, Jonathan H Abrams, and Robert S Weller. Ultrasound-guided interscalene needle placement produces successful anesthesia regardless of motor stimulation above or below 0.5 ma. *Anesthesia & Analgesia*, 105(3):848–852, 2007.

[17] Ban CH Tsui. *Atlas of ultrasound-and nerve stimulation-guided regional anesthesia.* Springer Science & Business Media, 2007.

[18] Stephen M Klein, M Steve Melton, Warren M Grill, and Karen C Nielsen. Peripheral nerve stimulation in regional anesthesia. *Regional Anesthesia & Pain Medicine*, 37(4):383–392, 2012.

[19] F Bonnet. Principles & practice of regional anaesthesia. *British Journal of Anaesthesia*, 91(3):453–454, 2003.

[20] Brian D Sites, Vincent W Chan, Joseph M Neal, Robert Weller, Thomas Grau, Zbigniew J Koscielniak-Nielsen, and Giorgio Ivani. The american society of regional anesthesia and pain medicine and the european society of regional anaesthesia and pain therapy joint committee recommendations for education and training in ultrasound-guided regional anesthesia. *Regional Anesthesia & Pain Medicine*, 34(1):40–46, 2009.

[21] Steven L Orebaugh, Michael L Kentor, and Brian A Williams. Adverse outcomes associated with nerve stimulator–guided and ultrasound-guided peripheral nerve blocks by supervised trainees: update of a single-site database. *Regional Anesthesia & Pain Medicine*, 37(6):577–582, 2012.

[22] Brian D Sites and John G Antonakakis. Ultrasound guidance in regional anesthesia: state of the art review through challenging clinical scenarios. *Local and regional anesthesia*, 2:1, 2009.

[23] Vincent Chan and Anna Dabu. *Guide pratique des blocs nerveux échoguidés.* Sauramps médical, 2006.

[24] BA Pollard. Ultrasound guidance for vascular access and regional anesthesia. *UltrasonixMedical Corporation*, 2012.

[25] Andrew T Gray. *Atlas of Ultrasound-Guided Regional Anesthesia E-Book: Expert Consult-Online.* Elsevier Health Sciences, 2012.

[26] Samer N Narouze. *Atlas of ultrasound-guided procedures in interventional pain management*. Springer, 2018.

[27] Harsha Shanthanna. Review of essential understanding of ultrasound physics and equipment operation. *World Journal of Anesthesiology*, 3(1):12–17, 2014.

[28] Carlos Tornero Tornero and Luis Aliaga Font. Training on regional anesthesia—from neurostimulation to ultrasound. *Techniques in regional anesthesia and pain management*, 16(3):131–135, 2012.

[29] R Gebhard, A Hadzic, and W Urmey. Dual guidance: a multimodal approach to nerve location. *Philadelphia: B Braun Medical Inc*, pages 2–50, 2008.

[30] Diogo Brüggemann da Conceição, Pablo Escovedo Helayel, Francisco Amaral Egydio de Carvalho, Jaderson Wollmeister, and Getúlio Rodrigues de Oliveira Filho. Ultrasound images of the brachial plexus in the axillary region. *Revista brasileira de anestesiologia*, 57(6):684–689, 2007.

[31] Thorsten Steinfeldt, U Schwemmer, Thomas Volk, Michael Neuburger, Thomas Wiesmann, Axel R Heller, Oliver Vicent, A Stanek, M Franz, Hinnerk Wulf, et al. Nerve localization for peripheral regional anesthesia. *Der Anaesthesist*, 63(7):597–602, 2014.

[32] Francesco Giovagnorio and Carlo Martinoli. Sonography of the cervical vagus nerve: normal appearance and abnormal findings. *American Journal of Roentgenology*, 176(3):745–749, 2001.

[33] Brian D Sites, Richard Brull, Vincent WS Chan, Brian C Spence, John Gallagher, Michael L Beach, Vincent R Sites, Sherif Abbas, and Gregg S Hartman. Artifacts and pitfall errors associated with ultrasound-guided regional anesthesia: Part ii: a pictorial approach to understanding and avoidance. *Regional Anesthesia & Pain Medicine*, 32(5):419–433, 2007.

[34] Gerald Retzl, Stephan Kapral, Manfred Greher, and Walter Mauritz. Ultrasonographic findings of the axillary part of the brachial plexus. *Anesthesia & Analgesia*, 92(5):1271–1275, 2001.

[35] Joris Emanuel Nicolaas Jaspers. Simple tools for surgeons: Design and evaluation of mechanical alternatives for robotic instruments for minimally invasive surgery. 2006.

[36] Nastaran Aghakhani, Milad Geravand, Navid Shahriari, Marilena Vendittelli, and Giuseppe Oriolo. Task control with remote center of motion constraint for minimally invasive robotic surgery. In *2013 IEEE International Conference on Robotics and Automation*, pages 5807–5812. IEEE, 2013.

[37] Anahi Perlas, Vincent WS Chan, and Martin Simons. Brachial plexus examination and localization using ultrasound and electrical stimulationa volunteer study. *Anesthesiology: The Journal of the American Society of Anesthesiologists*, 99(2):429–435, 2003.

[38] Pavan Kumar BC Raju and Calum RK Grant. Practical aspects of ultrasound-guided regional anaesthesia. *Anaesthesia & Intensive Care Medicine*, 14(4):137–141, 2013.

[39] P Marhofer, K Schrögendorfer, H Andel, H Koinig, W Girsch, S Kapral, and N Mayer. Combined sciatic nerve-3 in 1 block in high risk patient. *Anasthesiologie, Intensivmedizin, Notfallmedizin, Schmerztherapie: AINS*, 33(6):399–401, 1998.

[40] Hillenn Cruz Eng, Shayanti Meela Ghosh, and Ki Jinn Chin. Practical use of local anesthetics in regional anesthesia. *Current Opinion in Anesthesiology*, 27(4):382–387, 2014.

[41] Ronald S Adler. The use of compact ultrasound in anesthesia: friend or foe, 2007.

[42] NM Denny and W Harrop-Griffiths. Editorial i: Location, location, location! ultrasound imaging in regional anaesthesia, 2005.

[43] Brian D Sites, Brian C Spence, John D Gallagher, Christopher W Wiley, Marc L Bertrand, and George T Blike. Characterizing novice behavior associated with learning ultrasound-guided peripheral regional anesthesia. *Regional Anesthesia & Pain Medicine*, 32(2):107–115, 2007.

[44] Wilfried Kunde, Jochen Müsseler, and Herbert Heuer. Spatial compatibility effects with tool use. *Human Factors*, 49(4):661–670, 2007.

[45] John Graham Holden, John M Flach, and Y Donchin. Perceptual-motor coordination in an endoscopic surgery simulation. *Surgical endoscopy*, 13(2):127–132, 1999.

[46] Ignacio Avellino, Gilles Bailly, Geoffroy Canlorbe, Jérémie Belgihti, Guillaume Morel, and Marie-Aude Vitrani. Impacts of telemanipulation in robotic assisted surgery. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–15, 2019.

[47] Tobias Ortmaier. *Motion compensation in minimally invasive robotic surgery.* PhD thesis, Technische Universität München, 2003.

[48] Brian Davies. A review of robotics in surgery. *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine*, 214(1):129–140, 2000.

[49] Yik San Kwoh, Joahin Hou, Edmond A Jonckheere, and Samad Hayati. A robot with improved absolute positioning accuracy for ct guided stereotactic brain surgery. *IEEE Transactions on Biomedical Engineering*, 35(2):153–160, 1988.

[50] Ilana Nisky, Felix Huang, Amit Milstein, Carla M Pugh, Ferdinando A Mussa-Ivaldi, and Amir Karniel. Perception of stiffness in laparoscopy–the fulcrum effect. *Studies in health technology and informatics*, 173:313, 2012.

[51] Christine Sutter, Sandra Sülzenbrück, Martina Rieger, and Jochen Müsseler. Limitations of distal effect anticipation when using tools. *New Ideas in Psychology*, 31(3):247–257, 2013.

[52] RH Taylor and D Stoiariovici. gmedical robotics in computerintegrated surgery,• h ieee trans. *Robot. Automat*, 19(5):922–926, 2003.

[53] Jaydeep H Palep. Robotic assisted minimally invasive surgery. *Journal of minimal access surgery*, 5(1):1, 2009.

[54] Allison M Okamura. Haptic feedback in robot-assisted minimally invasive surgery. *Current opinion in urology*, 19(1):102, 2009.

[55] Mustapha Daouadi, Amer H Zureikat, Mazen S Zenati, Haroon Choudry, Alan Tsung, David L Bartlett, Steven J Hughes, Ken K Lee, A James Moser, and Herbert J Zeh. Robot-assisted minimally invasive distal pancreatectomy is superior to the laparoscopic technique. *Annals of surgery*, 257(1):128–132, 2013.

[56] BL Davies, RD Hibberd, AG Timoney, and JEA Wickham. A clinically applied robot for prostatectomies. *Computer Integrated Surgery: Technology and Clinical Applications*, pages 593–601, 1996.

[57] Lama Al Bassit. *Structures mécaniques à modules sphériques optimisées pour un robot médical de télé-échographie mobile.* PhD thesis, 2005.

[58] Nabil Zemiti, Guillaume Morel, Tobias Ortmaier, and Nicolas Bonnet. Mechatronic design of a new robot for force control in minimally invasive surgery. *IEEE/ASME Transactions On Mechatronics*, 12(2):143–153, 2007.

[59] Jonathan M Sackier and Yulun Wang. Robotically assisted laparoscopic surgery. *Surgical endoscopy*, 8(1):63–66, 1994.

[60] Moji Ghodoussi, Steven E Butner, and Yulun Wang. Robotic surgery-the transatlantic case. In *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*, volume 2, pages 1882–1888. IEEE, 2002.

[61] Gary S Guthart and J Kenneth Salisbury. The intuitive/sup tm/telesurgery system: overview and application. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, volume 1, pages 618–621. IEEE, 2000.

[62] S Haddadin and S Parusel. Franka emika panda, 2018.

[63] Rainer Bischoff, Johannes Kurth, Günter Schreiber, Ralf Koeppe, Alin Albu-Schäffer, Alexander Beyer, Oliver Eiberger, Sami Haddadin, Andreas Stemmer, Gerhard Grunwald, et al. The kuka-dlr lightweight robot arm-a new reference platform for robotics research and manufacturing. In *ISR 2010 (41st international symposium on robotics) and ROBOTIK 2010 (6th German conference on robotics)*, pages 1–8. VDE, 2010.

[64] Universal Robots. Technical specifications ur5, 2014.

[65] J Kessler, P Marhofer, PM Hopkins, and MW Hollmann. Peripheral regional anaesthesia and outcome: lessons learned from the last 10 years. *British Journal of Anaesthesia*, 114(5):728–745, 2015.

[66] Ana M Djuric, RJ Urbanic, and JL Rickli. A framework for collaborative robot (cobot) integration in advanced manufacturing systems. *SAE International Journal of Materials and Manufacturing*, 9(2):457–464, 2016.

[67] Frederick W Kremkau and KJ Taylor. Artifacts in ultrasound imaging. *Journal of ultrasound in medicine*, 5(4):227–237, 1986.

[68] Ju Zhang, Chen Wang, and Yun Cheng. Comparison of despeckle filters for breast ultrasound images. *Circuits, Systems, and Signal Processing*, 34(1):185–208, 2015.

[69] R Vanithamani and G Umamaheswari. Performance analysis of filters for speckle reduction in medical ultrasound images. *International Journal of Computer Applications*, 12(6):23–27, 2010.

[70] Sobika Ambardar and Manish Singhal. A review and comparative study of de-noising filters in ultrasound imaging. *International Journal of Emerging Technology and Advanced Engineering*, 4(8), 2014.

[71] Oussama Hadjerci, Adel Hafiane, Donatello Conte, Pascal Makris, Pierre Vieyres, and Alain Delbos. Computer-aided detection system for nerve identification using ultrasound images: A comparative study. *Informatics in Medicine Unlocked*, 3:29–43, 2016.

[72] Mark Nixon and Alberto Aguado. *Feature extraction and image processing for computer vision.* Academic press, 2019.

[73] Mihran Tuceryan and Anil K Jain. Texture analysis. In *Handbook of pattern recognition and computer vision*, pages 235–276. World Scientific, 1993.

[74] Qiang Ji, John Engel, and Eric Craine. Texture analysis for classification of cervix lesions. *IEEE Transactions on medical imaging*, 19(11):1144–1149, 2000.

[75] Anil K Jain and Yu Zhong. Page segmentation using texture analysis. *Pattern recognition*, 29(5):743–770, 1996.

[76] Marko Heikkila and Matti Pietikainen. A texture-based method for modeling the background and detecting moving objects. *IEEE transactions on pattern analysis and machine intelligence*, 28(4):657–662, 2006.

[77] Maria Petrou and Pedro Garcia Sevilla. *Image processing: dealing with texture*, volume 1. Wiley Online Library, 2006.

[78] Matti Pietikäinen, Abdenour Hadid, Guoying Zhao, and Timo Ahonen. *Computer vision using local binary patterns*, volume 40. Springer Science & Business Media, 2011.

[79] Umasankar Kandaswamy, Stephanie A Schuckers, and Donald Adjeroh. Comparison of texture analysis schemes under nonideal conditions. *IEEE Transactions on Image Processing*, 20(8):2260–2275, 2011.

[80] Timo Ojala, Matti Pietikainen, and Topi Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on pattern analysis and machine intelligence*, 24(7):971–987, 2002.

[81] Shu Liao, Max WK Law, and Albert CS Chung. Dominant local binary patterns for texture classification. *IEEE transactions on image processing*, 18(5):1107–1118, 2009.

[82] Amit Satpathy, Xudong Jiang, and How-Lung Eng. LBP-based edge-texture features for object recognition. *IEEE Transactions on Image Processing*, 23(5):1953–1964, 2014.

[83] Marko Heikkilä, Matti Pietikäinen, and Cordelia Schmid. Description of interest regions with local binary patterns. *Pattern recognition*, 42(3):425–436, 2009.

[84] Niraj P Doshi and Gerald Schaefer. A comprehensive benchmark of local binary pattern algorithms for texture retrieval. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 2760–2763. IEEE, 2012.

[85] Loris Nanni, Alessandra Lumini, and Sheryl Brahnam. Local binary patterns variants as texture descriptors for medical image analysis. *Artificial intelligence in medicine*, 49(2):117–125, 2010.

[86] Timo Ahonen, Abdenour Hadid, and Matti Pietikäinen. Face recognition with local binary patterns. In *European conference on computer vision*, pages 469–481. Springer, 2004.

[87] Antonio Fernández, Marcos X Álvarez, and Francesco Bianconi. Texture description through histograms of equivalent patterns. *Journal of mathematical imaging and vision*, 45(1):76–102, 2013.

[88] Jianguo Zhang, Marcin Marszałek, Svetlana Lazebnik, and Cordelia Schmid. Local features and kernels for classification of texture and object categories: A comprehensive study. *International journal of computer vision*, 73(2):213–238, 2007.

[89] Sheryl Brahnam, Lakhmi C Jain, Loris Nanni, Alessandra Lumini, et al. *Local binary patterns: new variants and applications.* Springer, 2014.

[90] Kai Wang, Charles-Edmond Bichot, Yan Li, and Bailin Li. Local binary circumferential and radial derivative pattern for texture classification. *Pattern Recognition*, 67:213–229, 2017.

[91] Li Liu, Jie Chen, Paul Fieguth, Guoying Zhao, Rama Chellappa, and Matti Pietikainen. A survey of recent advances in texture representation. *arXiv preprint arXiv:1801.10324*, 3, 2018.

[92] Hongliang Jin, Qingshan Liu, Hanqing Lu, and Xiaofeng Tong. Face detection using improved LBP under bayesian framework. In *Image and Graphics (ICIG'04), Third International Conference on*, pages 306–309. IEEE, 2004.

[93] Zhenhua Guo, Lei Zhang, and David Zhang. A completed modeling of local binary pattern operator for texture classification. *IEEE Transactions on Image Processing*, 19(6):1657–1663, 2010.

[94] Jie Chen, Vili Kellokumpu, Guoying Zhao, and Matti Pietikäinen. RLBP: Robust local binary pattern. In *BMVC*, 2013.

[95] Adel Hafiane, Guna Seetharaman, and Bertrand Zavidovique. Median binary pattern for textures classification. In *International Conference Image Analysis and Recognition*, pages 387–398. Springer, 2007.

[96] Li Liu, Yunli Long, Paul W Fieguth, Songyang Lao, and Guoying Zhao. BRINT: binary rotation invariant and noise tolerant texture classification. *IEEE Transactions on Image Processing*, 23(7):3071–3084, 2014.

[97] Gerald Schaefer and Niraj P Doshi. Multi-dimensional local binary pattern descriptors for improved texture analysis. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 2500–2503. IEEE, 2012.

[98] Adel Hafiane, Kannappan Palaniappan, and Guna Seetharaman. Joint adaptive median binary patterns for texture classification. *Pattern Recognition*, 48(8):2609–2620, 2015.

[99] Zhenhua Guo, Xingzheng Wang, Jie Zhou, and Jane You. Robust texture image representation by scale selective local binary patterns. *IEEE Transactions on Image Processing*, 25(2):687–699, 2016.

[100] Li Liu, Songyang Lao, Paul W Fieguth, Yulan Guo, Xiaogang Wang, and Matti Pietikäinen. Median robust extended local binary pattern for texture classification. *IEEE Transactions on Image Processing*, 25(3):1368–1381, 2016.

[101] Felix Juefei-Xu, Vishnu Naresh Boddeti, and Marios Savvides. Local binary convolutional neural networks. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, volume 1. IEEE, 2017.

[102] Tsung-Han Chan, Kui Jia, Shenghua Gao, Jiwen Lu, Zinan Zeng, and Yi Ma. Pcanet: A simple deep learning baseline for image classification? *IEEE Transactions on Image Processing*, 24(12):5017–5032, 2015.

[103] Mircea Cimpoi, Subhransu Maji, and Andrea Vedaldi. Deep filter banks for texture recognition and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3828–3836, 2015.

[104] Li Liu, Paul Fieguth, Yulan Guo, Xiaogang Wang, and Matti Pietikäinen. Local binary features for texture classification: Taxonomy and experimental study. *Pattern Recognition*, 62:135–160, 2017.

[105] Mário AT Figueiredo, José M Bioucas-Dias, and Robert D Nowak. Majorization–minimization algorithms for wavelet-based image restoration. *IEEE Transactions on Image processing*, 16(12):2980–2991, 2007.

[106] Hiroyuki Takeda, Sina Farsiu, and Peyman Milanfar. Kernel regression for image processing and reconstruction. *IEEE Transactions on image processing*, 16(2):349–366, 2007.

[107] Antoni Buades, Bartomeu Coll, and Jean-Michel Morel. A review of image denoising algorithms, with a new one. *Multiscale Modeling & Simulation*, 4(2):490–530, 2005.

[108] Humor Hwang and Richard A Haddad. Adaptive median filters: new algorithms and results. *IEEE Transactions on image processing*, 4(4):499–502, 1995.

[109] Pei-Eng Ng and Kai-Kuang Ma. A switching median filter with boundary discriminative noise detection for extremely corrupted images. *IEEE Transactions on image processing*, 15(6):1506–1516, 2006.

[110] How-Lung Eng and Kai-Kuang Ma. Noise adaptive soft-switching median filter. *IEEE Transactions on image processing*, 10(2):242–251, 2001.

[111] Xuming Zhang and Youlun Xiong. Impulse noise removal using directional difference based noise detector and adaptive weighted mean filter. *IEEE Signal processing letters*, 16(4):295–298, 2009.

[112] Adel Hafiane, Kannappan Palaniappan, and Guna Seetharaman. Adaptive median binary patterns for texture classification. In *2014 22nd International Conference on Pattern Recognition*, pages 1138–1143. IEEE, 2014.

[113] Timo Ojala, Topi Maenpaa, Matti Pietikainen, Jaakko Viertola, Juha Kyllonen, and Sami Huovinen. Outex-new framework for empirical evaluation of texture analysis algorithms. In *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, volume 1, pages 701–706. IEEE, 2002.

[114] Manik Varma and Andrew Zisserman. A statistical approach to texture classification from single images. *International journal of computer vision*, 62(1-2):61–81, 2005.

[115] Phil Brodatz. *Textures: a photographic album for artists and designers*. Dover Pubns, 1966.

[116] P Mallikarjuna, M Fritz, A Tavakoli Targhi, E Hayman, B Caputo, and JO Eklundh. The kth-tips and kth-tips2 databases, 2006.

[117] Gertjan J Burghouts and Jan-Mark Geusebroek. Material-specific adaptation of color invariant features. *Pattern Recognition Letters*, 30(3):306–313, 2009.

[118] Hui Zhou, Runsheng Wang, and Cheng Wang. A novel extended local-binary-pattern operator for texture analysis. *Information Sciences*, 178(22):4314–4325, 2008.

[119] Abdolhossein Fathi and Ahmad Reza Naghsh-Nilchi. Noise tolerant local binary pattern operator for efficient texture analysis. *Pattern Recognition Letters*, 33(9):1093–1100, 2012.

[120] Xiaopeng Hong, Guoying Zhao, Matti Pietikainen, and Xilin Chen. Combining LBP difference and feature correlation for texture description. *IEEE Transactions on Image Processing*, 23(6):2557–2568, 2014.

[121] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, and Andrea Vedaldi. Deep filter banks for texture recognition, description, and segmentation. *International Journal of Computer Vision*, 118(1):65–94, 2016.

[122] J Alison Noble and Djamal Boukerroui. Ultrasound image segmentation: a survey. *IEEE Transactions on medical imaging*, 25(8):987–1010, 2006.

[123] Heng-Da Cheng, Juan Shan, Wen Ju, Yanhui Guo, and Ling Zhang. Automated breast cancer detection and classification using ultrasound images: A survey. *Pattern recognition*, 43(1):299–317, 2010.

[124] Afsaneh Jalalian, Shamans BT Mashohor, Hajjah Rozi Mahmud, M Iqbal B Saripan, Abdul Rahman B Ramli, and Babak Karasfi. Computer-aided detection/diagnosis of breast cancer in mammography and ultrasound: a review. *Clinical imaging*, 37(3):420–426, 2013.

[125] Mohammad Alkhatib, Adel Hafiane, Omar Tahri, Pierre Vieyres, and Alain Delbos. Adaptive median binary patterns for fully automatic nerves tracking in ultrasound images. *Computer methods and programs in biomedicine*, 160:129–140, 2018.

[126] Ed-Edily Mohd Azhari, Muhd Mudzakkir Mohd Hatta, Zaw Zaw Htike, and Shoon Lei Win. Tumor detection in medical imaging: a survey. *International journal of Advanced Information Technology*, 4(1):21, 2014.

[127] Kinjal A Joshi and Darshak G Thakore. A survey on moving object detection and tracking in video surveillance system. *International Journal of Soft Computing and Engineering*, 2(3):44–48, 2012.

[128] Patricio Loncomilla, Javier Ruiz-del Solar, and Luz Martínez. Object recognition using local invariant features for robotic applications: A survey. *Pattern Recognition*, 60:499–514, 2016.

[129] Adel Hafiane, Pierre Vieyres, and Alain Delbos. Deep learning with spatiotemporal consistency for nerve segmentation in ultrasound images. *arXiv preprint arXiv:1706.05870*, 2017.

[130] Oussama Hadjerci, Adel Hafiane, Nicolas Morette, Cyril Novales, Pierre Vieyres, and Alain Delbos. Assistive system based on nerve detection and needle navigation in ultrasound images for regional anesthesia. *Expert Systems with Applications*, 61:64–77, 2016.

[131] Oussama Hadjerci, Adel Hafiane, Pascal Makris, Donatello Conte, Pierre Vieyres, and Alain Delbos. Nerve detection in ultrasound images using median gabor binary pattern. In *International Conference Image Analysis and Recognition*, pages 132–140. Springer, 2014.

[132] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[133] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[134] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[135] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[136] Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen Awm Van Der Laak, Bram Van Ginneken, and Clara I Sánchez. A survey on deep learning in medical image analysis. *Medical image analysis*, 42:60–88, 2017.

[137] Qinghua Huang, Fan Zhang, and Xuelong Li. Machine learning in ultrasound computer-aided diagnostic systems: a survey. *BioMed research international*, 2018, 2018.

[138] Oussama Hadjerci, Adel Hafiane, Pierre Vieyres, Donatello Conte, Pascal Makris, and Alain Delbos. On-line learning dynamic models for nerve detection in ultrasound videos. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 131–135. IEEE, 2016.

[139] Mohammad Alkhatib, Adel Hafiane, Pierre Vieyres, and Alain Delbos. Deep visual nerve tracking in ultrasound images. *Computerized Medical Imaging and Graphics*, 76:101639, 2019.

[140] Vladimir Vapnik. *The nature of statistical learning theory.* Springer science & business media, 2013.

[141] Naiyan Wang and Dit-Yan Yeung. Learning a deep compact image representation for visual tracking. In *Advances in neural information processing systems*, pages 809–817, 2013.

[142] Min Li, Chandra Kambhamettu, and Maureen Stone. Automatic contour tracking in ultrasound images. *Clinical linguistics & phonetics*, 19(6-7):545–554, 2005.

[143] Lisa Tang, Tim Bressmann, and Ghassan Hamarneh. Tongue contour tracking in dynamic ultrasound via higher-order mrfs and efficient fusion moves. *Medical image analysis*, 16(8):1503–1520, 2012.

[144] Jacinto C Nascimento and Jorge S Marques. Robust shape tracking with multiple models in ultrasound images. *IEEE Transactions on Image Processing*, 17(3):392–406, 2008.

[145] Anastasios Roussos, Athanassios Katsamanis, and Petros Maragos. Tongue tracking in ultrasound images with active appearance models. In *Image Processing (ICIP), 2009 16th IEEE International Conference on*, pages 1733–1736. IEEE, 2009.

[146] Gustavo Carneiro and Jacinto C Nascimento. Combining multiple dynamic models and deep learning architectures for tracking the left ventricle endocardium in ultrasound data. *IEEE transactions on pattern analysis and machine intelligence*, 99(1):1, 2013.

[147] Julian Guerrero, Septimiu E Salcudean, James A McEwen, Bassam A Masri, and Savvakis Nicolaou. Real-time vessel segmentation and tracking for ultrasound imaging applications. *IEEE transactions on medical imaging*, 26(8):1079–1090, 2007.

[148] Paul M Novotny, Jeff A Stoll, Nikolay V Vasilyev, J Pedro, Pierre E Dupont, Todd E Zickler, and Robert D Howe. Gpu based real-time instrument tracking with three-dimensional ultrasound. *Medical image analysis*, 11(5):458–464, 2007.

[149] Qi Duan, Elsa D Angelini, Susan L Herz, Christopher M Ingrassia, Kevin D Costa, Jeffrey W Holmes, Shunichi Homma, and Andrew F Laine. Region-based endocardium tracking on real-time three-dimensional ultrasound. *Ultrasound in Medicine and Biology*, 35(2):256–265, 2009.

[150] Jonathan Deutscher, Andrew Blake, and Ian Reid. Articulated body motion capture by annealed particle filtering. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No. PR00662)*, volume 2, pages 126–133. IEEE, 2000.

[151] Dorin Comaniciu, Visvanathan Ramesh, and Peter Meer. Kernel-based object tracking. *IEEE Transactions on pattern analysis and machine intelligence*, 25(5):564–577, 2003.

[152] Carlo Tomasi and Takeo Kanade. Detection and tracking of point features. 1991.

[153] Jifeng Ning, Lei Zhang, David Zhang, and Chengke Wu. Robust object tracking using joint color-texture histogram. *International Journal of Pattern Recognition and Artificial Intelligence*, 23(07):1245–1263, 2009.

[154] Dongsheng Ding, Zengru Jiang, and Chengyuan Liu. Object tracking algorithm based on particle filter with color and texture feature. In *Control Conference (CCC), 2016 35th Chinese*, pages 4031–4036. IEEE, 2016.

[155] Piotr Bilinski, François Bremond, and Mohamed Becha Kaaniche. Multiple object tracking with occlusions using hog descriptors and multi resolution images. 2009.

[156] Mohammad Alkhatib and Adel Hafiane. Robust adaptive median binary pattern for noisy texture classification and retrieval. *IEEE Transactions on Image Processing*, 28(11):5407–5418, 2019.

[157] Adel Hafiane, Guna Seetharaman, Kannappan Palaniappan, and Bertrand Zavidovique. Rotationally invariant hashing of median binary patterns for texture classification. *Image Analysis and Recognition*, pages 619–629, 2008.

[158] Martin Danelljan, Andreas Robinson, Fahad Shahbaz Khan, and Michael Felsberg. Beyond correlation filters: Learning continuous convolution operators for visual tracking. In *European Conference on Computer Vision*, pages 472–488. Springer, 2016.

[159] Martin Danelljan, Goutam Bhat, F Shahbaz Khan, and Michael Felsberg. Eco: Efficient convolution operators for tracking. In *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA*, pages 21–26, 2017.

[160] Kaihua Zhang, Qingshan Liu, Yi Wu, and Ming-Hsuan Yang. Robust visual tracking via convolutional networks without training. *IEEE Transactions on Image Processing*, 25(4):1779–1792, 2016.

[161] Hyeonseob Nam and Bohyung Han. Learning multi-domain convolutional neural networks for visual tracking. In *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on*, pages 4293–4302. IEEE, 2016.

[162] Heng Fan and Haibin Ling. Sanet: Structure-aware network for visual tracking. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2217–2224. IEEE, 2017.

[163] Luca Bertinetto, Jack Valmadre, Joao F Henriques, Andrea Vedaldi, and Philip HS Torr. Fully-convolutional siamese networks for object tracking. In *European conference on computer vision*, pages 850–865. Springer, 2016.

[164] Jack Valmadre, Luca Bertinetto, João Henriques, Andrea Vedaldi, and Philip HS Torr. End-to-end representation learning for correlation filter based tracking. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 5000–5008. IEEE, 2017.

[165] Qiang Wang, Jin Gao, Junliang Xing, Mengdan Zhang, and Weiming Hu. Dcfnet: Discriminant correlation filters network for visual tracking. *arXiv preprint arXiv:1704.04057*, 2017.

[166] Tianzhu Zhang, Changsheng Xu, and Ming-Hsuan Yang. Multi-task correlation particle filter for robust object tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, page 3, 2017.

[167] Yuankai Qi, Shengping Zhang, Lei Qin, Hongxun Yao, Qingming Huang, Jongwoo Lim, and Ming-Hsuan Yang. Hedged deep tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4303–4311, 2016.

[168] Chao Ma, Jia-Bin Huang, Xiaokang Yang, and Ming-Hsuan Yang. Hierarchical convolutional features for visual tracking. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3074–3082, 2015.

[169] Yibing Song, Chao Ma, Lijun Gong, Jiawei Zhang, Rynson WH Lau, and Ming-Hsuan Yang. Crest: Convolutional residual learning for visual tracking. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2574–2583. IEEE, 2017.

[170] Jacinto C Nascimento, Gustavo Carneiro, and António Freitas. Tracking and segmentation of the endocardium of the left ventricle in a 2d ultrasound using deep learning architectures and monte carlo sampling. *Biomedical Image Segmentation: Advances and Trends*, page 387, 2016.

[171] M Sanjeev Arulampalam, Simon Maskell, Neil Gordon, and Tim Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on signal processing*, 50(2):174–188, 2002.

[172] Oliver Heinemeyer and Carl D Reimers. Ultrasound of radial, ulnar, median, and sciatic nerves in healthy subjects and patients with hereditary motor and sensory neuropathies. *Ultrasound in medicine & biology*, 25(3):481–485, 1999.

[173] Keras Keras. Deep learning library for theano and tensorflow. 2015, 2019.

[174] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[175] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[176] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.

[177] John G Daugman. Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *JOSA A*, 2(7):1160–1169, 1985.

[178] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[179] Chad R Johnson, Roger C Barr, and Stephen M Klein. A computer model of electrical stimulation of peripheral nerves in regional anesthesia. *Anesthesiology: The Journal of the American Society of Anesthesiologists*, 106(2):323–330, 2007.

[180] Chin-Hsing Kuo and Jian S Dai. Kinematics of a fully-decoupled remote center-of-motion parallel manipulator for minimally invasive surgery. *Journal of Medical Devices*, 6(2):021008, 2012.

[181] Olivier Schneider and Jocelyne Troccaz. A six-degree-of-freedom passive arm with dynamic constraints (padyc) for cardiac surgery application: Preliminary experiments. *Computer aided surgery*, 6(6):340–351, 2001.

[182] Etienne Dombre, Micaël Michelin, François Pierrot, Philippe Poignet, Philippe Bidaud, Guillaume Morel, Tobias Ortmaier, Damien Sallé, Nabil Zemiti, Philippe Gravez, et al. Marge project: design, modeling and control of assistive devices for minimally invasive surgery. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 1–8. Springer, 2004.

[183] Janez Funda, Russell H Taylor, Ben Eldridge, Stephen Gomory, and Kreg G Gruben. Constrained cartesian motion control for teleoperated surgical robots. *IEEE Transactions on Robotics and Automation*, 12(3):453–465, 1996.

[184] Roderick CO Locke and Rajni V Patel. Optimal remote center-of-motion location for robotics-assisted minimally-invasive surgery. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 1900–1905. IEEE, 2007.

[185] Cong D Pham, Fernando Coutinho, Antonio C Leite, Fernando Lizarralde, Pål J From, and Rolf Johansson. Analysis of a moving remote center of motion for robotics-assisted minimally invasive surgery. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1440–1446. IEEE, 2015.

[186] Hamidreza Azimian, Rajni V Patel, and Michael D Naish. On constrained manipulation in robotics-assisted minimally invasive surgery. In *2010 3rd IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics*, pages 650–655. IEEE, 2010.

[187] Murilo M Marinho, Mariana C Bernardes, and Antônio PL Bó. A programmable remote center-of-motion controller for minimally invasive surgery using the dual quaternion framework. In *5th IEEE RAS/EMBS International Conference on Biomedical Robotics and Biomechatronics*, pages 339–344. IEEE, 2014.

[188] Hermann Mayer, István Nagy, and Alois Knoll. Kinematics and modelling of a system for robotic surgery. In *On Advances in Robot Kinematics*, pages 181–190. Springer, 2004.

[189] Bassem Dahroug, Brahim Tamadazte, and Nicolas Andreff. 3d path following with remote center of motion constraints. In *ICINCO (1)*, pages 84–91, 2016.

[190] Rémi Chalard. *Robustesse à la variabilité du comportement mécanique du point d'insertion en chirurgie mini-invasive robotisée. Application à la biopsie de prostate.* PhD thesis, Sorbonne Université, 2019.

[191] Shigeyuki Shimachi, Surakij Hirunyanitiwatna, Yasunori Fujiwara, Akira Hashimoto, and Yoshinori Hakozaki. Adapter for contact force sensing of the da vinci® robot. *The International Journal of Medical Robotics and Computer Assisted Surgery*, 4(2):121–130, 2008.

[192] Tiago Leal Ghezzi and Oly Campos Corleta. 30 years of robotic surgery. *World journal of surgery*, 40(10):2550–2557, 2016.

[193] Linfei Xiong, Chin Boon Chng, Chee Kong Chui, Peiwu Yu, and Yao Li. Shared control of a medical robot with haptic guidance. *International journal of computer assisted radiology and surgery*, 12(1):137–147, 2017.

[194] Rebecca Kokes, Kevin Lister, Rao Gullapalli, Bao Zhang, Alan MacMillan, Howard Richard, and Jaydev P Desai. Towards a teleoperated needle driver robot with haptic feedback for rfa of breast tumors under continuous mri. *Medical image analysis*, 13(3):445–455, 2009.

[195] Tianmiao Wang, Dapeng Zhang, and Liu Da. Remote-controlled vascular interventional surgery robot. *The International Journal of Medical Robotics and Computer Assisted Surgery*, 6(2):194–201, 2010.

[196] Virtuose 3D Desktop, year = 2010, url = https://www.haption.com/fr/products-fr/virtuose-3d-desktop-fr.html, urldate = 2020-02-29.

[197] FRANKA EMIKA. Franka control interface, 2018.

[198] Michael Peshkin and J Edward Colgate. Cobots. *Industrial Robot: An International Journal*, 1999.

[199] Adel Hafiane, Pierre Vieyres, and Alain Delbos. Phase-based probabilistic active contour for nerve detection in ultrasound images for regional anesthesia. *Computers in biology and medicine*, 52:88–95, 2014.

[200] Erik Smistad and Lasse Løvstakken. Vessel detection in ultrasound images using deep convolutional neural networks. In *Deep Learning and Data Labeling for Medical Applications*, pages 30–38. Springer, 2016.

[201] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017.

[202] Larry J Segerlind and H Saunders. Applied finite element analysis. 1987.

[203] Nicolas Morette. *Contribution à la Navigation de robots mobiles: approche par modèle direct et commande prédictive.* PhD thesis, 2009.

[204] Fenglin Liang. *Caractérisation et modélisation du comportement rhéologique des boues résiduaires urbaines concentrées.* PhD thesis, 2016.

[205] Quoc-Viet Dang. *Conception et commande d'une interface haptique à retour d'effort pour la CAO.* PhD thesis, 2013.

[206] Quigley Morgan, Berger Eric, and Andrew Y Ng. Stair: Hardware and software architecture. In *AAAI 2007 Robotics Workshop*, 2007.

[207] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y Ng. Ros: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, page 5. Kobe, Japan, 2009.

Mohammad ALKHATIB

Système d'assistance robotisé basé sur l'imagerie échographique pour l'anesthésie loco-régionale

Résumé :

L'anesthésie régionale guidée par ultrasons (UGRA) devient une technique puissante dans les procédures chirurgicales et la gestion de la douleur. Toutefois, la procédure UGRA nécessite un long processus d'apprentissage et des années d'expérience. Les aides robotisées sont déjà utilisées dans l'assistance médicale et montrent leur énorme impact. Ainsi, l'UGRA robotisée pourrait être d'une grande aide en aidant les anesthésistes à utiliser des techniques et des outils qui améliorent la précision et la sécurité de la procédure, par exemple en évitant les traumatismes nerveux ou les dommages aux tissus sains. Le système UGRA robotisé présente deux défis majeurs : la qualité visuelle des images échographiques, qui souffre de bruit, et le contrôle robotique de l'insertion de l'aiguille. Par conséquent, cette thèse se concentre sur le développement de méthodes de détection et de suivi des nerfs dans les images échographiques et sur le contrôle robotique de l'insertion de l'aiguille.

Dans cette thèse, l'image ultrasonore est analysée en soulignant l'importance de trouver des caractéristiques robustes et fortes dans l'image. Puisque le nerf partage certaines propriétés texturales bruyantes, nous proposons des modèles binaires médians adaptatifs robustes (RAMBP) qui ont les avantages de l'efficacité et de la robustesse aux textures avec différents bruits élevés. Ces caractéristiques sont utilisées comme base pour différentes techniques de détection et de suivi des nerfs et évaluées en termes de précision, de cohérence, de complexité temporelle et de traitement de différentes situations nerveuses. Cependant, nous pensons que les modèles de réseaux neuronaux convolutifs (CNN) sont également efficaces pour les textures d'images échographiques. Ainsi, pour apprendre des caractéristiques plus optimales, nous proposons de fusionner les modèles RAMBP et CNN afin d'obtenir de meilleures performances pour la détection et le suivi des nerfs.

Pour le contrôle robotique de l'UGRA, l'objectif de cette contribution est de guider l'anesthésiste là où le centre de mouvement à distance (RCM) impose une tâche très difficile, où il est important de s'assurer que l'aiguille se déplace dans les contraintes du point d'insertion et de l'évitement des obstacles afin d'éviter de nuire au patient et d'atteindre le nerf ciblé. Pour répondre à ce besoin, deux stratégies de contrôle ont été présentées tout en utilisant le contrôle cartésien RCM : un système d'insertion d'aiguille UGRA téléopéré avec retour de force haptique ; un système d'insertion d'aiguille CoBotic. La commande UGRA téléopérée permet l'insertion de l'aiguille sous les contraintes du RCM avec un retour de force appliqué au dispositif haptique. D'autre part, le contrôle CoBotic UGRA permet de contrôler le mouvement de l'aiguille en appliquant la force de la main de l'utilisateur opérateur sur le bras du robot tout en respectant les contraintes RCM. Enfin, nous avons discuté de l'évitement des régions à risque en appliquant de petites forces répulsives à l'effecteur final lorsque la pointe de l'aiguille s'approche d'une région à risque. Une image topologique a été construite en utilisant une technique de segmentation par réseau neuronal et une force répulsive a été calculée en fonction de la distance entre la pointe de l'aiguille et la région à risque.

Mots clés : Suivi de nerfs, suivi visuel, apprentissage profond, extraction de caractéristiques, suivi d'objets basé sur la multimodalité, vision par ordinateur, images échographiques, cinématique de l'insertion de l'aiguille, contraintes du centre de mouvement distant, télérobotique, robotique médicale, anesthésie régionale.

Mohammad ALKHATIB

Robotic assistance system based on ultrasound images analysis for regional anesthesia

Summary :

Ultrasound-guided regional anesthesia (UGRA) is becoming a powerful technique in surgical procedures and pain management. However, the UGRA procedure requires a long learning process and years of experience. Robotic aids are already used in medical assistance and show their enormous impact. For this, robotic UGRA could be of great help in helping anesthetists use techniques and tools that improve the accuracy and safety of the procedure, for example, by avoiding nerve trauma or damage to healthy tissue. The robot-assisted UGRA system presents two major challenges: the visual quality of ultrasound images, which suffers from noise, and the robotic control of needle insertion. Therefore, this thesis focuses on the development of methods for nerve detection and tracking in ultrasound images and robotic control of needle insertion.

In this thesis, the ultrasound image is analyzed by highlighting the importance of finding robust and strong features in the image. Since the nerve shares some noisy textural properties, we propose Robust Adaptive Medial Binary Models (RAMBPs) that have the advantages of efficiency and robustness to textures with different high noises. These features are used as the basis for different nerve detection and tracking techniques and evaluated in terms of accuracy, consistency, time complexity, and handling different nerve situations. However, we believe that convolutional neural network (CNN) models are also effective for ultrasound image textures. Thus, to learn more optimal features, we propose to merge RAMBP and CNN models to achieve better performance for nerve detection and tracking.

For robotic control of the UGRA, the goal of this contribution is to guide the anesthetist where the remote motion center (RCM) imposes a very difficult task, where it is important to ensure that the needle moves within the constraints of the insertion point and obstacle avoidance in order to prevent patient harm and reach the targeted nerve. To address this need, two control strategies were presented while using the RCM Cartesian control: a teleoperated UGRA needle insertion system with haptic force feedback; a CoBotic needle insertion system. The teleoperated UGRA control enables needle insertion under RCM constraints with force feedback applied to the haptic device. On the other hand, the CoBotic UGRA control enables the needle movement control by applying hand force by the operator user on the robot arm while respecting the RCM constraints. Finally, we discussed avoiding risky regions by applying small repulsive forces to the end-effector once the needle's tip gets close to a risky region. A topological image has been constructed by using a neural network segmentation technique and a repulsive force has been computed with respect to the distance between the needle tip and the risky region.

Keywords : Nerve tracking, Visual tracking, Deep-learning, Feature extraction, Multi-Modal based object tracking, Computer vision, Ultrasound images, Needle insertion kinematics, Remote center of motion constraints, Telerobotics, Medical robotics, Regional anesthesia.

Laboratoire PRISME
Université d'Orléans
63 Av. de Lattre de
Tassigny, 18000 Bourges