UNIVERSITÀ DEGLI STUDI DI VERONA

DEPARTMENT OF COMPUTER SCIENCE
DOCTORAL PROGRAM IN COMPUTER SCIENCE
CYCLE XXXIV, 2021

Patient-specific simulation for autonomous surgery

Author:
Eleonora TAGLIABUE

Supervisor:
Prof. Paolo FIORINI
Co-supervisor:
Dr. Diego DALL'ALBA

PhD program coordinator:
Prof. Massimo MERRO

Thesis referees:
Prof. Christian DURIEZ
Prof. Michael YIP

Submitted in fulfillment of the requirements for the degree of Doctor of Philosophy at the Department of Computer Science

S.S.D. ING-INF/05

March 2022

Patient-specific simulation for autonomous surgery — Eleonora Tagliabue
PhD thesis
Verona, 18 March 2022
ISBN xxxxx-xxxx-xxx

“They tried to bury me,
but they did not know I was a seed."

–Dinos Christianopoulos

Sommario

Un sistema robotico chirurgico autonomo deve interagire con un ambiente anatomico complesso, che si deforma ed è caratterizzato da proprietà spesso incerte. Tale sistema potrebbe trarre vantaggio dalla presenza di una simulazione dell'anatomia di ciascun paziente. Ad esempio, la simulazione può fornire un ambiente sicuro e controllato in cui effettuare la progettazione, la verifica e la validazione dell'intervento autonomo. In aggiunta, può essere utilizzata per generare una grande mole di dati personalizzati, che possono essere sfruttati per l'apprendimento di modelli e/o compiti. L'obiettivo di questa Tesi è di analizzare le varie modalità con cui la simulazione può supportare un sistema autonomo e di proporre delle soluzioni per favorire il suo utilizzo nella chirurgia robotica.

Inizialmente, si considerano tutte le fasi necessarie per la creazione di tale simulazione, dalla scelta del modello sulla base delle informazioni disponibili in fase pre-operatoria, al suo aggiornamento in fase intra-operatoria per compensare un'eventuale parametrizzazione imprecisa. Si propone di sfruttare reti neurali allenate con dati sintetici sia per la generazione di un modello specifico di ciascun paziente, sia per lo sviluppo di un metodo per l'aggiornamento dei parametri di tale modello, partendo direttamente da dati acquisiti da sensori intra-operativi.

In una seconda parte, si testa come la simulazione possa assistere un sistema robotico chirurgico autonomo, sia durante la fase di apprendimento di un'azione chirurgica sia durante la sua esecuzione. I risultati ottenuti confermano che la simulazione può essere usata per allenare con efficienza algoritmi che richiedono molteplici interazioni con l'ambiente, compensando così i rischi connessi all'acquisizione di dati su veri sistemi robotici chirurgici. Infine, si propone un framework modulare per la chirurgia autonoma che incorpora funzioni deliberative per gestire ambienti anatomici reali, caratterizzati da proprietà incerte. L'integrazione di un ambiente di simulazione all'interno di tale framework si dimostra fondamentale per la pianificazione ottima e l'arricchimento e il monitoraggio dell'esecuzione reale.

I contributi presentati in questa Tesi hanno il potenziale di accellerare lo sviluppo ed ottimizzare le prestazioni di sistemi robotici chirurgici autonomi, rendendo sempre più realistica la loro applicazione all'interno di contesti clinici reali.

Abstract

An Autonomous Robotic Surgical System (ARSS) has to interact with the complex anatomical environment, which is deforming and whose properties are often uncertain. Within this context, an ARSS can benefit from the availability of patient-specific simulation of the anatomy. For example, simulation can provide a safe and controlled environment for the design, test and validation of the autonomous capabilities. Moreover, it can be used to generate large amounts of patient-specific data that can be exploited to learn models and/or tasks. The aim of this Thesis is to investigate the different ways in which simulation can support an ARSS and to propose solutions to favor its employability in robotic surgery.

We first address all the phases needed to create such a simulation, from model choice in the pre-operative phase based on the available knowledge to its intra-operative update to compensate for inaccurate parametrization. We propose to rely on deep neural networks trained with synthetic data both to generate a patient-specific model and to design a strategy to update model parametrization starting directly from intra-operative sensor data.

Afterwards, we test how simulation can assist the ARSS, both for task learning and during task execution. We show that simulation can be used to efficiently train approaches that require multiple interactions with the environment, compensating for the riskiness to acquire data from real surgical robotic systems. Finally, we propose a modular framework for autonomous surgery that includes deliberative functions to handle real anatomical environments with uncertain parameters. The integration of a personalized simulation proves fundamental both for optimal task planning and to enhance and monitor real execution.

The contributions presented in this Thesis have the potential to introduce significant step changes in the development and actual performance of autonomous robotic surgical systems, making them closer to applicability to real clinical conditions.

Acknowledgments

This Thesis is the result of a journey that lasted three years, composed of both satisfactions and challenges. All the achievements, both scientific and personal, would not have been possible without all the people who walked alongside me, whom I sincerely have to thank.

First and foremost, a profound thank you goes to my supervisor, Paolo. Thanks for trusting me since the beginning and for always giving me freedom to decide which direction to take, in all its meanings. Thank you for your endless support and for your ability to never letting me feel in the wrong place. A sincere thank you goes to Diego. Your enthusiasm towards my work has been fundamental to keep my motivation high, especially in discouraging moments. Thanks for all the meetings, for your insights, for all the projects you involved me in.

Thanks to all the brilliant people I had the opportunity to collaborate with during these years. Thanks to Altair Lab members, who have always been there to offer help and technical suggestions, but also to enjoy aperitivos. A sincere thank you goes to Marco, for your help, your incredible efficiency and your trust. Thanks to Micha, for suggesting to join our efforts, opening the route to such a nice collaboration. Thanks to the colleagues from 1.64a, for making working hours more enjoyable.

I would like to thank Stéphane, for welcoming me multiple times in Strasbourg and for always making me feel a team member, even once back in Italy. Thanks to Igor, for patiently teaching me so many concepts in numerical methods and simulation. Thanks to the whole MIMESIS team, for all the stimulating discussions, for your feedback and for teaching me how to play la coinche. A special thank goes to Hugo, Anne-Laure, Andrea and Fred, for all the moments spent together and for giving me a reason to visit again.

I am thankful to all my friends in Verona, with whom I could share so many good moments. Thanks to my volleyball mates, I have been so lucky to find such a great group to spend evenings with. Thanks to Giovanni, Ameya and Sanat, my Rozzampia team, for confirming that culture is not a barrier to friendship. A special thank you is for Giovanni. Thank you for your support, for sharing your values, for always providing some good food for thought. It's been a pleasure to share this adventure with you, with all the ups and downs.

Thanks to all the people who have always been there to support me. Thanks to Vale, the best mate I could wish to share this travel with. Thanks for your emphaty, for truly and deeply sharing with me both the achievements and the toughest moments. Thanks to Andre, for being able to charge my batteries and my mood all the times. Your energy, ambition and genuineness have been really inspiring. Thanks to Napo and Tia: you are the ones I know I can call at any time, being sure you'll be there. Thanks to Giuli, for all the laughs and the toughtless moments. Thanks to Marti and Mari, we started sharing our adventures on day 1 on Polimi desks and, since then, we have never stopped. Thanks to Fra, for your contagious enthusiasm and for being there to discuss anything, anywhere, anytime.

Last but not least, a huge thank you goes to my family. Thanks to my parents Alberto and Paola, for your continual support, your trust and your pride. Thanks to Frenz, for being my motivation to never settle.

Contents

Sommario i

Abstract ii

Acknowledgements iii

List of abbreviations viii

1 Introduction 1
- 1.1 Simulation and robot autonomy . . . 2
- 1.2 The Autonomous Robotic Surgery project . . . 4
  - 1.2.1 Simulation and autonomous robotic surgery . . . 5
- 1.3 Contributions and structure of the Thesis . . . 6
  - 1.3.1 Structure of the Thesis . . . 7
- 1.4 Conclusion . . . 7

2 Deformable models for surgical simulation 9
- 2.1 Introduction . . . 9
- 2.2 Challenges of surgical simulation . . . 10
  - 2.2.1 Challenge of patient-specific simulations . . . 10
  - 2.2.2 Challenge of interactive simulations . . . 12
  - 2.2.3 Challenge of stable simulations . . . 12
- 2.3 Continuum mechanics methods . . . 13
  - 2.3.1 The finite element method . . . 16
  - 2.3.2 Alternatives to the standard finite element method . . . 17
- 2.4 Heuristic methods . . . 18
  - 2.4.1 Mass spring models . . . 19
  - 2.4.2 Position based dynamics . . . 20
  - 2.4.3 Chain Mail . . . 21
- 2.5 Machine learning based methods . . . 22
- 2.6 Conclusion . . . 25

I Learning patient-specific models 27

3 Learning patient-specific pre-operative models 28

3.1 Introduction . . . 28
3.2 Deformation modeling in breast US scanning . . . 29
3.3 US scanning of a breast phantom . . . 31
3.3.1 Experimental setup . . . 31
3.3.2 Modelling assumptions . . . 33
3.3.3 Evaluation metrics . . . 33
3.4 The finite element method . . . 34
3.4.1 Penalty method . . . 35
3.4.2 Lagrange multipliers method . . . 35
3.4.3 Prescribed displacements method . . . 36
3.4.4 FE-based breast model . . . 36
3.5 Position Based Dynamics . . . 37
3.5.1 PBD-based breast model . . . 38
3.6 The U-Mesh framework . . . 42
3.6.1 U-Mesh validation on a synthetic dataset . . . 44
3.7 Analysis of the FE-based models . . . 45
3.7.1 Discussion . . . 46
3.8 Comparison of the different methods . . . 49
3.8.1 Discussion . . . 51
3.9 Conclusion . . . 52

4 Learning intra-operative model update **57**
4.1 Introduction . . . 57
4.2 Intra-operative estimation of tissue attachment points . . . 59
4.2.1 BA-Net: Binary Attachment Network . . . 60
4.3 Model update with BA-Net during soft tissue manipulation . . . 62
4.3.1 Experiments on synthetic phantoms . . . 63
4.3.2 Experiments on ex-vivo human kidney . . . 65
4.4 Experimental results on synthetic phantoms . . . 67
4.4.1 Discussion . . . 69
4.5 Experimental results on ex-vivo human kidney . . . 73
4.5.1 Discussion . . . 74
4.6 Conclusion . . . 75

II Simulation for autonomous surgery **79**

5 Learning a surgical task in simulation **80**
5.1 Introduction . . . 80
5.2 UnityFlexML: a framework to learn surgical tasks in simulation . . . 82
5.2.1 Robot platform . . . 82
5.2.2 Simulation environment . . . 83
5.3 Background on learning-based methods . . . 83
5.3.1 Reinforcement learning . . . 83
5.3.2 Learning from demonstrations . . . 85
5.4 Learning tissue retraction within UnityFlexML . . . 86

5.4.1 Experimental setup . . . 86
5.4.2 Learning methods . . . 88
5.4.3 Evaluation metrics . . . 89
5.5 Autonomous tissue retraction in simulation . . . 90
5.5.1 Results and discussion . . . 91
5.6 Sim-to-real autonomous tissue retraction . . . 93
5.6.1 Results and discussion . . . 93
5.7 Conclusion . . . 94

6 A deliberative framework for autonomous robotic surgery 97
6.1 Introduction . . . 97
6.2 Dealing with uncertainty in autonomous robotic surgery . . . 98
6.3 DEFRAS: DEliberative Framework for Robot Assisted Surgery . . . 99
6.4 Implementation of tissue retraction in DEFRAS . . . 102
6.4.1 Tissue retraction task . . . 102
6.4.2 Task planning . . . 103
6.4.3 Sensing . . . 104
6.4.4 Acting . . . 105
6.4.5 Monitoring . . . 105
6.4.6 Learning . . . 106
6.4.7 Simulated environment . . . 107
6.5 Autonomous tissue retraction with DEFRAS . . . 108
6.5.1 Experimental setup . . . 108
6.5.2 Validation of DEFRAS in simulation . . . 109
6.5.3 Validation of DEFRAS on the dVRK . . . 111
6.6 Experimental results of autonomous TR in simulation . . . 112
6.6.1 Discussion . . . 114
6.7 Experimental results of autonomous TR on the dVRK . . . 115
6.7.1 Discussion . . . 115
6.8 Conclusions . . . 116

7 Conclusion 119
7.1 Future research directions . . . 120

Bibliography 125

List of abbreviations

AP	Attachment Points
ARSS	Autonomous Robotic Surgical System
BA-Net	Binary Attachment Network
BEM	Boundary Element Method
CNN	Convolutional Neural Network
CPU	Central Processing Unit
DEFRAS	DEliberative Framework for Robot Assisted Surgery
DL	Deep Learning
DRL	Deep Reinforcement Learning
dVRK	da Vinci Research Kit
FE	Finite Element
FEM	Finite Element Method
FRAS	Framework for Robot Assisted Surgery
GPU	Graphic Processing Unit
LfD	Learning from Demonstrations
ML	Machine Learning
MOR	Model Order Reduction
MSM	Mass Spring Model
PBD	Position Based Dynamics
PBM	Patient-specific Biomechanical Model
POD	Proper Orthogonal Decomposition
PSM	Patient Side Manipulator
RMIS	Robot-assisted Minimally Invasive Surgery
ROI	Region Of Interest
TR	Tissue Retraction

Chapter 1

Introduction

Nowadays, robotic systems are exploited in many practical applications across a wide range of fields. Traditionally, they have been used to perform repetitive actions without risking to suffer from fatigue, or jobs that are too hazardous for humans. For example, robots are used in manufacturing and assembly lines (e.g., in automotive or electronics industries), to find objects underwater or to explore other planets. However, robots are nowadays increasingly present also in our daily life, if we think about all the existing types of household assistants. Thanks to the advancements in technology and artificial intelligence, the latest generation of robotic systems have become increasingly autonomous. They can carry out some basic decision-making that allow them to perform tasks in autonomy, even in environments which might be not completely known in advance.

Among others, the use of robots has revolutionized the medical field. For example, mobile robots are streamlining supply delivery, cleaning and disinfection in hospitals; more complex robotized systems have been developed to enable enhanced forms of therapy (e.g., stereotactic radiotherapy by the CyberKnife system [1]) and rehabilitation [2].

Robots have come into play in the surgical domain as well, transforming the way surgeries are performed. In orthopedics, robotic systems can assist surgeons on several interventions, such as knee (Mako [3] or Rosa [4]) and spine surgery (Mazor X Stealth [5]). In the last few decades, robots have been increasingly adopted in laparoscopic surgery, where they interact with an environment (i.e., visceral anatomy) which is a way more dynamic and complex than the one faced by orthopedic robots. In Robot-assisted Minimally Invasive Surgery (RMIS), the surgeon sits at a console, few meters away from the surgical robot, and controls the articulated surgical tools through some joysticks, while having immersive visualization of the surgical site as provided by the laparoscopic camera. The most popular RMIS platform is the da Vinci surgical system [6], which is largely used in urology, gynecology and general surgery (Fig.1.1). Compared to open surgery, robotic surgery combines enhanced dexterity and stereo vision with the smaller incisions on patient's body typical of traditional MIS, thus enabling surgeons to perform more precise surgeries and improve the overall procedure outcome. The introduction of advanced instrumentation and sensing technologies have been boosting RMIS utilization in recent years, making robotic surgery one of the most successful areas in robotics. Research in surgical robotics has become a very active field thanks to the presence of some initiatives like the da Vinci Research Kit (dVRK) [7, 8]. The dVRK is a research platform that provides a network of academic laboratories worldwide with clinically dismissed da Vinci

systems, equipped with dedicated electronics to access robot data.

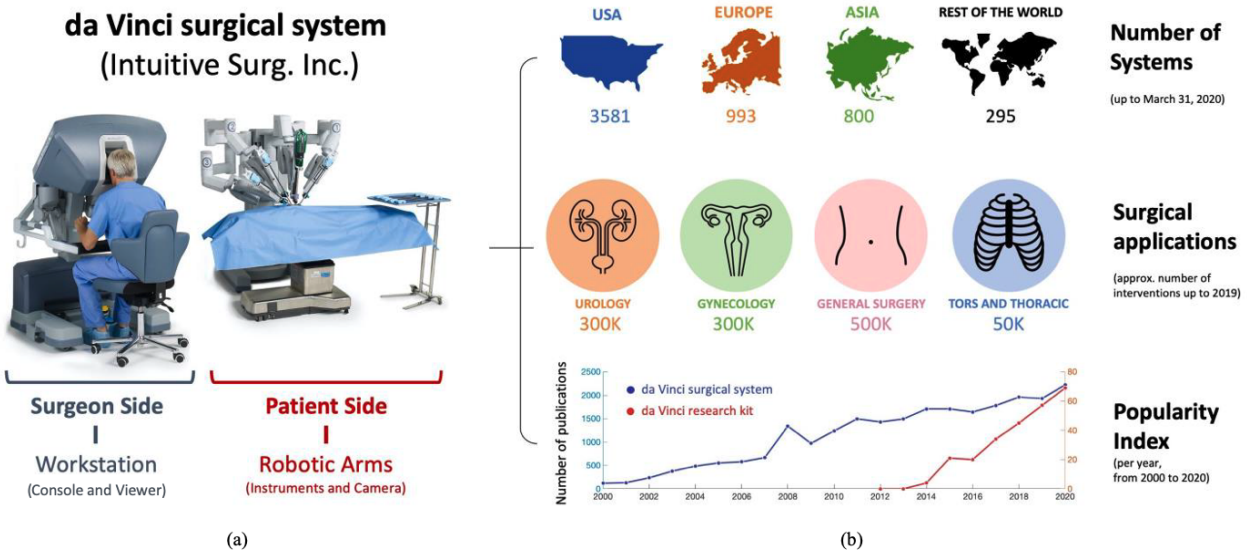


Figure 1.1: (a) The da Vinci surgical system: the surgeon teleoperates the robotic arms at the patient side from the surgeon console; (b) From top to bottom: distribution of da Vinci surgical systems worldwide in 2020; surgical applications and total number of interventions until 2019 using the da Vinci surgical system; number of publications citing the da Vinci surgical system and the da Vinci Research Kit as of [7]. Adapted from [7].

In accordance with the latest trends in general service robots, one of the most popular research areas in surgical robotics deals with automating aspects of RMIS [7, 9]. At the moment, surgical robots do not perform any actions in autonomy, mainly for technical and legal reasons, although most of them have all the needed mechanical and computing capabilities. Unlikely other fields where autonomous robots could be seen as a threat to workers, the potential benefit of autonomy in robotic surgery is unanimously recognized. For example, an autonomous surgical robot can execute simple repetitive parts of the intervention, limiting human errors and the fatigue of surgeons, who shall just monitor the overall execution and focus only on the most critical tasks.

As a final consideration, one of the reasons why autonomous robots in general are only slowly progressing lies in the lack of a unified theoretical framework for planning and executing autonomous tasks, especially for risky tasks (like surgery). These tasks cannot be generally represented by hand-crafted policies and pre-defined sets of actions, but needs to be learned from experience by examining how humans address unpredictable situations. As robotic automation gains widespread adoption, the need for automation architectures grows.

1.1 Simulation and robot autonomy

In general, creating autonomous systems introduces some major challenges, both in the design and in the validation phase. In this context, a significant support to the process can be provided by computer simulation. In fact, simulation can represent a safe, fast and cheap solution that allows to understand how autonomous robots should be designed and controlled

for safe operation and improved performance [10]. In particular, we have identified two main ways in which simulation can be of help in automating robotic tasks.

First of all, simulation provides a virtual environment where to *test, validate and verify* the behavior of the autonomous system in a safe and controlled way. Simulations can be customized to model multiple agents, environmental conditions and interactions between them. In this way, it is possible to analyze the response of the system to different settings, allowing to identify possible problems and predict potentially dangerous situations. Simulation can be employed for this purpose at development stage, to optimize system behavior and design fail safe strategies, as well as during task execution whenever decision making is involved, to foresee the outcome of a possible action.

Secondly, simulation represents a versatile environment to *generate large amounts of data* that can be used to train Machine Learning (ML) algorithms. Such algorithms are being increasingly used in robotics for their capacity to learn behavior models (e.g., of the robot, of the environment) and/or control policies directly from data, but they require a large database for successful learning. Simulation can be used to generate a large amount of realistic synthetic data, compensating for the lack of real world data.

These advantages provide a strong motivation to an extensive use of simulation in robotics. However, there are still some barriers to overcome, as highlighted by Choi et al. in more details [10]:

1. *Lack of solutions*: to date, there is a scarcity of simulation environments offering all the potential features of interest in robotics, due to the challenges involved in their development. Obtaining a realistic simulation for robotics requires multi-disciplinary expertise since it involves the modelling of (i) robot dynamics, (ii) perception systems (e.g., sensors), (iii) the environment, which might be unstructured, evolving and dynamic, and (iv) the interaction and communication with the surroundings, where multiple agents might be present. This whole process becomes even more difficult when a human is present in the loop and his/her behavior (mechanical and/or cognitive, depending on the application) should also be modelled. Moreover, the availability of experts from such different fields, devoted to the development of the same platform, is not easy to have, especially in academic research laboratories, which usually gather people of the same field.

2. *Difficulty in model choice and calibration*: defining the scenario to use for a robotic simulation involves a series of modeling choices (e.g., modelling human interaction, multiple environmental conditions, etc.) and the selection of a large set of parameters. This process can become daunting, especially with increasing complexity of the models involved (e.g., when the environment contains deformable components and/or contacts with friction). Moreover, model parametrization can be a time-consuming and tedious process, which relies on either ad-hoc parameter identification strategies or fine tuning from trial-and-error attempts until the desired behavior is obtained. The model selection process also requires to consider the trade-off between accuracy and computation time, depending on the application; sometimes, choosing a complex model without the availability of good parameters might lead to worse results that take longer than what can be obtained with simpler models.

3. *Uncertainty is not generally modelled*: simulations should incorporate sources of un-

certainty that characterize the real environment where the robot operates. These uncertainties can be due to the presence of complex and dynamic behaviors that cannot be precisely modelled (e.g., friction or impacts), the lack of knowledge of model parameters and the presence of sensor noise. However, incorporating these uncertainties represents an open issue, due to the limited knowledge of the environments where robots operate.

The presence of these open issues should not hinder the use of simulation in robotics, but instead encourage research and development in this area, since the benefits of its use for the design of the robots of the future are clear.

1.2 The Autonomous Robotic Surgery project

This Thesis lies within the *Autonomous Robotic Surgery* project[1], which aims at developing methodologies that will enable the execution of a surgical intervention by a robotic system in complete autonomy. One of the main challenges an Autonomous Robotic Surgical System (ARSS) has to face is how to properly interact with the anatomical environment, which is composed of soft tissues that deform in response to the interaction with surgical instruments as well as to physiological effects (such as breathing or heart beating). Moreover, tissue behavior is very complex to model and its properties are highly variable among different patients and difficult to measure. The complexity and uncertainty of the environment make the development of an ARSS a challenging task, since it requires to consider many different aspects. For example, the definition of the intervention plan and its instantiation on a specific patient already introduce some major difficulties. First of all, it is necessary to identify the sequence of actions needed to complete the desired surgical task in general terms. This requires the integration of both a-priori knowledge from textbooks and the experience, actions and reasoning of multiple surgeons, that can be only learnt from data of real interventions. Once the surgical plan is defined, it must be adjusted to cope with the current patient, according to his/her specific clinical situation and geometric and mechanical properties.

Even though a patient-specific intervention plan is available, the ARSS must be capable of adapting it during execution based on current situation, since the anatomical environment might behave differently from what expected from pre-operative knowledge. To cope with such an uncertain environment, it is key to incorporate strategies for real-time situation awareness, reasoning and control, that allow prompt reaction and online replanning in response to unknown or unexpected situations.

Automating a process in an environment as critical as the surgical one, where errors can be deadly, also requires the availability of an integrated framework that can be used for both the design, test and validation of the intervention plan in the pre-operative phase, and the control and monitoring of task execution intra-operatively, handling environmental uncertainties.

[1]The *Autonomous Robotic Surgery* project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme under grant agreement No. 742671.

1.2.1 Simulation and autonomous robotic surgery

A great support to the autonomous execution of surgical actions by an ARSS can be provided by the availability of a simulation of the anatomical environment. Such simulation is based on a Patient-specific Biomechanical Model (PBM), which is created in a pre-operative phase from geometry and physical properties of the surgical areas extracted from each patient diagnostic images and/or initialized with values from the literature (Fig. 1.2).

First of all, simulation represents a test bench where to *design and verify* the surgical plan before its actual execution on the real system. This step is essential in safety-critical scenarios as surgery, since it allows to optimize the plan accounting for each patient's characteristics, and test it before execution to visualize possible problems, identify any major errors and predict potentially critical situations. Robotic actions verification is also required while the intervention is taking place, especially when actions replanning is needed. In order to guarantee that the intra-operative verification is reliable, it is essential that the simulation continuously represents an up-to-date replica of the current surgical situation. The pre-operative PBM should therefore be continuously updated to reflect all the changes introduced by the surgical manipulations based on real data. Fig. 1.2 highlights the role of simulation for surgical plan verification for an ARSS.

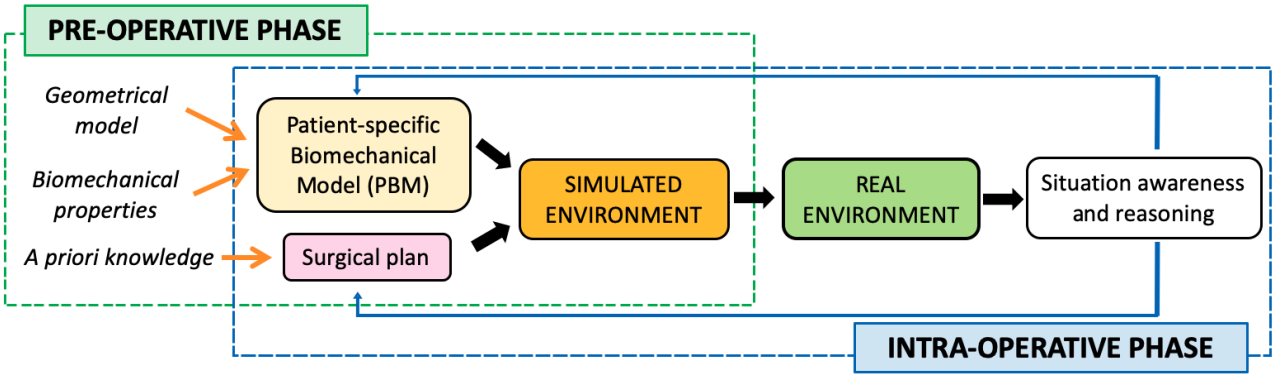


Figure 1.2: In an ARSS, a PBM is instantiated pre-operatively with patient-specific geometry and biomechanical properties. Such model is employed to initialize the simulation environment where the surgical plan is verified in the pre-operative phase. In the intra-operative phase, both the PBM and the surgical plan are adjusted depending on the current surgical situation.

The availability of an accurate simulation of the anatomical environment can support the ARSS via the *generation of realistic* – and even patient-specific – *data* that can be used for learning purposes. This is very important since the complexity of both the environment and the cause-effect flow governing surgical actions make ML algorithms ideal for learning tasks, but they are challenged by the impracticality to acquire real world data in the medical field. In fact, such data are generally difficult to get due to the technical, ethical and privacy issues involved in data collection process. By providing a way to generate large amounts of realistic data, computer simulation allows to compensate for this limitation. Generated data can also be exploited for the design of novel data-driven models or methods that might be more informative or more efficient than traditional approaches and can assist the ARSS during task execution.

Furthermore, simulation can *provide* the ARSS *with critical information* during task execu-

tion. It can guide the ARSS towards the structures of interest, which might be hidden from the partial view available intra-operatively, or warn it in case it approaches critical regions which might be not directly recognizable on intra-operative images. It can complement information from real sensors, for example estimating the interaction forces between instruments and tissues when direct force measurement systems are not available, as is the case of most current surgical robotic systems [11]. Moreover, since simulation represents the ARSS internal knowledge of the environment, a continuous monitoring of the discrepancy between the simulated and the real scenarios allows to assess if the execution on the real setting is coherent with what the ARSS expects based on its internal knowledge, and adopt specific strategies to accommodate safety in case it is not.

1.3 Contributions and structure of the Thesis

In this Thesis, we address the main aspects where simulation can be of help in robotics, especially considering the design of autonomous systems (Section 1.1). We focus on the challenging context of surgery, characterized by a complex environment which deforms due to the robot interaction and whose properties need to be tuned for each patient. Therefore, the main goal of this PhD Thesis is to investigate the different ways in which a patient-specific simulation can support an autonomous robotic surgical system. We first address all the phases needed to create a simulation environment that can be employed to such purpose, from the pre-operative design and selection of the most appropriate model, to its intra-operative update. Then, we test how simulation can assist the autonomous execution, both in task learning phase and during real surgical task execution.

In particular, the main contributions of this Thesis can be summarized as follows:

1. We tackle the problem of creating a patient-specific simulation environment where the robot plan can be designed, tested, validated and verified. We examine advantages and disadvantages of different approaches in light of the requirements of surgery and propose some solutions for their parametrization, thus addressing the difficulty in model choice and calibration.

2. We develop an open source framework supporting efficient, realistic and stable simulation of deformable tissues which can be used to such purpose. The framework provides a solution where multiple interactions with the anatomy can be safely and efficiently performed, and can be used as a starting point for further development.

3. We investigate the feasibility of using simulations to generate large amounts of data needed to train learning-based strategies. This approach is tested for different purposes, i.e. to generate a highly efficient deformation model, to design a novel strategy for intra-operative update of model parametrization and to learn a surgical task.

4. We present a modular architecture that provides all the key functionalities for autonomous robotic surgery. In particular, it integrates deliberative capabilities, which are essential to deal with the uncertainty and dynamism of real anatomical environments.

5. We contribute to advancing research in autonomous surgical robotics by making all the developed methods open source. In this way, we take an action towards reproducibility and verifiability of science, and encourage the use of simulation in surgical robotics.

With this Thesis, we address the main limitations that currently prevent extensive use of simulation in surgical robotics and we emphasize the multiple benefits that simulation can bring to the development of autonomous surgical systems. The contributions presented in this Thesis have the potential to introduce significant step changes in the development and actual performance of autonomous robotic surgical systems, making them closer to applicability to real clinical conditions.

1.3.1 Structure of the Thesis

This manuscript begins with an overview of the world of surgical simulation in Chapter 2. We identify the main challenges to face and review the existing approaches that can be used to model the deformable anatomy. The remaining of this manuscript is divided in two parts.

Part I presents our contributions to the development and parametrization of the patient-specific model that drives the simulation supporting the ARSS. In particular, Chapter 3 addresses the problem of pre-operative model selection and its patient-specific parametrization. We conduct a thorough evaluation of existing physics and non-physics based methodologies that can be used to create personalized surgical simulations, considering the main performance criteria of interest for an ARSS, and we compare them with a newly proposed data-driven approach. Whereas, Chapter 4 tackles the problem of intra-operative correction of the simulated model based on real data and present a complete pipeline for its applicability to the clinical settings. Methods presented in Part I leverage on simulations to generate large amounts of data that are used to learn deformation models and strategies for parameters update.

In Part II, we discuss how simulation can support autonomous surgical task execution. In Chapter 5 we show that simulation can provide a virtual scenario where surgical tasks can be successfully learnt with approaches that require multiple interactions with the environment, thus compensating for the impracticality and riskiness to acquire large amounts of data from real surgical robotic systems. Chapter 6 presents a framework for autonomous robotic surgery that integrates all the features needed for the execution of surgical tasks within the real deformable and uncertain anatomical environment.

Finally, we draw conclusions and reason on future research in Chapter 7.

1.4 Conclusion

This Thesis addresses the different ways in which simulation can provide support to autonomous robotic surgery. First of all, the generation of a simulation that accounts for each patient's geometry as well as known biomechanical properties allows to have an environment encoding all the knowledge available to the ARSS. Such environment can be used both for safe design, test and validation of a surgical task/plan in the pre-operative phase and to generate large amounts of patient-specific data that can be used to learn models and/or control

policies. Furthermore, simulation can support an ARSS in the intra-operative phase by providing additional information about the environment and allowing to continuously monitor the real execution.

Throughout this Thesis, we review the main aspects where simulation can assist an ARSS and we propose some solutions to address the main challenges and limitations that are currently preventing its extensive use in autonomous surgery. We first tackle the generation of a patient-specific simulation which meets all the requirements of surgery. We present some methods to address the difficulty of patient-specific model parameterization, both in the pre-operative and intra-operative phases. Then, we test the possibility of using simulation as a tool to generate data that can be used to learn parameters and/or tasks. Finally, we test how simulation can support autonomous task execution on a real surgical robotic system, both in pre-operative and intra-operative stage.

Despite its many benefits, reliance on simulation is not largely common in robotics yet. This is even more true in robotic surgery, due to the complexity to model the deformable anatomical environment, whose properties are highly variable between different subjects and often uncertain. By reviewing all the aspects where simulation can significantly contribute to the development of autonomous robotic surgical systems, addressing the main challenges involved, and publicly sharing the developed methods, we encourage a more widespread use of simulation in autonomous surgery.

Chapter 2

Deformable models for surgical simulation

2.1 Introduction

Modelling and simulation of the deformable behavior of the anatomy can support surgical applications in several ways.

First of all, deformation models can support *deformable image registration* techniques. Several medical applications, from radioteraphy to pre-operative planning, require to align images of the same patient acquired at different times, in which patient's internal configuration might have changed due to physiological effects, repositioning or evolution of the pathology. All these changes cause the imaged anatomies to be in a different deformed configuration. The image registration process can thus benefit from being informed by a biomechanical model able to compensate for such effects [12].

In computer-assisted interventions, models of the anatomical environment are exploited to *enhance the surgical scene* and *provide the surgeon with critical information*, eventually improving the overall quality of surgery. A model able to account for the deformations arising from both surgical manipulations and physiological movements can be used to facilitate intra-operative navigation. Deformation models can be employed to augment the surgeon' view, providing guidance towards the structures of interest and/or highlighting the regions to avoid, which might be hidden in the current view due to the limited field of view of the endoscopic camera or the presence of other tissues [13, 14, 15]. Another example application is given by needle-based procedures, such as biopsy, brachyteraphy or radiofrequency ablation, where the target might move from its pre-operative position following tissue deformations caused by the needle-tissue interaction. Since the resolution of available intra-operative imaging is too low to provide reliable guidance, being able to model such interaction can ease the needle insertion process and enable successful navigation towards the area of interest [16].

Deformation models can also be used in the context of *surgical training*, which has become fundamental with the increasing popularity of MIS. The limited working space available during MIS procedures compromises surgeon's dexterity, thus making adequate training a crucial factor for success [17, 18]. Virtual simulators have been increasingly adopted as training tools, since they are safe and enable unlimited practice sessions. Training quality is correlated

to the degree of realism and smoothness in the representation of the interactions between the anatomical environment and the surgical tools provided by simulators [17, 19].

Modeling and simulation of the deforming anatomical environment has the potential to bring enormous benefits to the *design and validation of medical robots* [10]. The role of simulation becomes crucial for the development of “smart" medical robots, where simulation can both help to design systems with improved safety and performance, and support autonomous planning and control of robotic surgical tasks [20].

After reviewing the main challenges involved in realistic surgical simulation in Section 2.2, this Chapter provides an overview of the main categories of deformable models for surgical simulation, highlighting their advantages and disadvantages. We first discuss the preferred modelling strategy for applications requiring high simulation accuracy, which relies on a mathematical description of soft tissues behavior through the laws of continuum mechanics (Section 2.3). However, this kind of simulations generally suffer from high computational cost. When high solution speed is required, some simplifications are usually introduced into the physics of the environment, in order to achieve a simulation which is at least physically plausible. This class of methods, referred to as heuristic models, will be described in Section 2.4. Finally, we present how machine learning techniques can be exploited to predict tissue deformations in Section 2.5.

2.2 Challenges of surgical simulation

In all the mentioned applications, it would be desirable to rely on a deformation model which represents real tissues behavior in an accurate way, while requiring minimal computation time and staying numerically stable. However, these factors have conflicting computational requirements. Reaching high accuracy levels generally implies usage of complex nonlinear models, which in turns means higher computational cost and higher probability of numerical instability. On the other hand, model simplifications can be introduced to minimize computation time whenever performance is the main factor of interest. Simultaneously meeting all these requirements is very difficult, especially with increasing complexity of the simulation. As a consequence, a variety of models have been proposed to approximate the behavior of soft tissues, optimized in different ways to meet the specific needs of the application they address [21].

2.2.1 Challenge of patient-specific simulations

The first main challenge of surgical simulation is the proper representation of the manipulated tissues, to achieve an accurate modelling of their behavior. With the only exception of training simulators, which can rely on anatomical models extracted from available datasets [22] and approximated deformation properties, all the other mentioned applications require patient-specific model parametrization to reach clinically acceptable precision.

The first step for personalized simulation is the reliance on *organ geometry* from actual patient’s data (Fig. 2.1). The 3D model of the anatomy of interest is generally constructed starting from pre-operative volumetric images such as computerized tomography (CT) or magnetic resonance imaging (MRI). These imaging techniques provide a 3D view of the patient’s internal anatomy, where the different structures can be identified and labelled with a

process called segmentation. Although several attempts have been made towards automating the segmentation task [23], they are extremely context-dependent, still leaving manual segmentation as the most widely applicable and reliable approach. Afterwards, a 3D surface model of the tissues is generated from the segmentation mask, commonly relying on the marching cubes algorithm [24]. In some approaches, like ChainMail (Section 2.4.3), the generated 3D surface already represents the input to the simulation. However, many simulation approaches also require the discretization of the volume enclosed within the generated surface into elementary entities, such as tetrahedrons or hexahedrons, in a process called 3D meshing. Tetrahedral elements are the preferred ones for the discretization of anatomical shapes, since they can easily represent irregular geometries with sufficient detail. The 3D meshing process is generally automated, and relies on Delaunay triangulation [25]. The accuracy of mesh-based simulations depends on the both the number and the regularity of the generated elements, which in turns is influenced by the quality of the starting surface.

Overall, the generation of a good geometrical model is essential for reliable simulations, but it requires extensive manual processing which is both time consuming and subject-dependent, thus representing a first challenge in the generation of patient-specific simulations. Moreover, it is clear that all the inaccuracies in the model generation phase will introduce uncertainties in the physics quantities predicted by simulations [26].

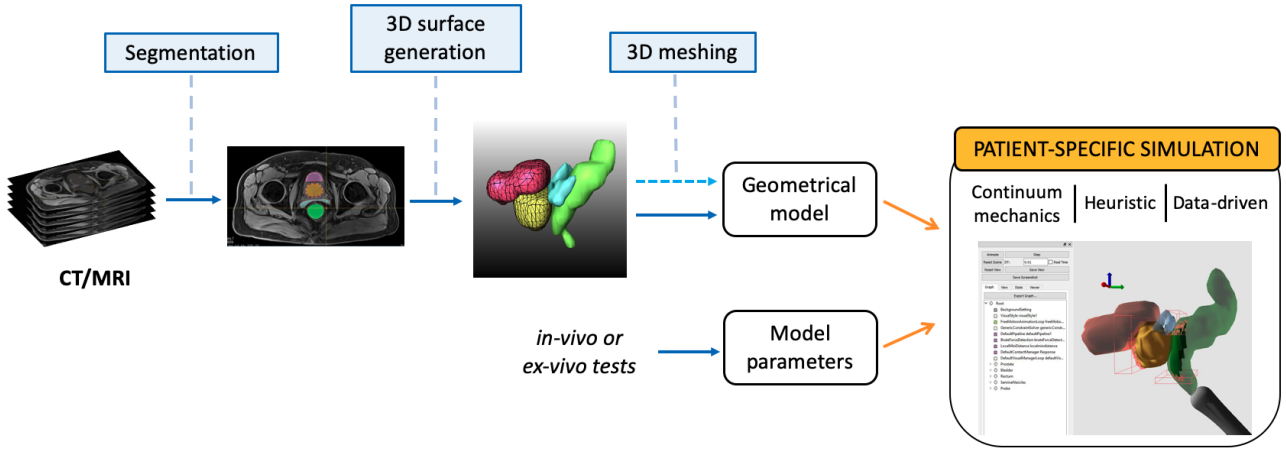


Figure 2.1: General workflow for patient-specific simulation. A geometrical model of the anatomy is obtained from segmentation of the patient's volumetric images (CT or MRI) and subsequent 3D surface generation. The obtained surface might be eventually meshed into a full 3D volume, as required by some simulation approaches. Either in-vivo or ex-vivo tests are performed to identify model parameters describing the deformable behavior of the anatomy. The geometrical model and the model parameters initialize the simulation environment, which can be based on different analytical models.

The generation of a reliable patient-specific simulation requires the selection of the most suitable *analytical model* to describe the complex anatomical behavior. This represents an additional challenge since tissues behavior is usually characterized by non-linear material laws, often including heterogeneity (i.e., the presence of multiple materials with different mechanical properties) and anisotropy (i.e., materials showing different mechanical properties in different directions). Existing models can be classified in three main categories. The first category contains models that describe anatomical behavior through the laws of continuum

mechanics, which can be solved using ad hoc numerical techniques. The second category consists of heuristic methods, which rely on simplified models of soft tissues geometry with specific constraints that allow to simulate elastic properties. Finally, the third category rely on machine learning techniques to represent tissues behavior. These three categories of models will be investigated in the following Sections.

The choice of the model influences the set of *model parameters* that describe tissues behavior. Model parameters, which include both material properties and boundary conditions, have to be careful selected ideally accounting for patient-specific characteristics. To this end, tissues mechanical properties can be determined through ex-vivo and/or in-vivo measurements. Ex-vivo measurements are obviously easier to perform since there are fewer constraints on the working space for the experimental setup, and conditions can be made more repetitive. However, material properties have proven to vary substantially from the true in-vivo properties, making these experiments less accurate for realistic modelling [27]. In-vivo experiments provide more realistic data, but conditions are less stable and boundary conditions for modelling a lot more difficult to define. By providing information about organs internal composition, tissues biomechanical properties might be inferred from traditional medical imaging techniques (CTs or MRIs). However, there is not an established relationship between tissue composition and their mechanical behavior. Elastography-based imaging has the potential to provide a direct estimate of organ stiffness, by measuring how acoustic strain waves propagate into the tissues [28, 29]. Such techiques still rely on ad-hoc setups and are not easily usable in the standard clinical environments.

2.2.2 Challenge of interactive simulations

The computation time required by the simulation is another critical issue. Applications like intra-operative guidance and surgical simulation require high solution speeds, compatible with the required update rates for smooth visual and (potentially) haptic feedback. The update rate required for visual feedback, that allows the human sensory system to achieve continuous motion of the environment, is in the range 20-60 Hz. In case haptics is also present, stable and smooth kinestetic sensation are guaranteed by an update rate between 300 and 1 kHz [30]. Being able to meet such strict time requirements becomes more and more challenging with increasing complexity of the simulation, for example if the organ mechanical behavior is described by nonlinear laws, if the interactions among multiple organs need to be accounted for, or if challenging surgical actions must be modelled, such as cutting and suturing, which involve topological changes. Computation of the numerical solution to the complex non-linear equations that accurately describe the behavior of soft tissues can be very expensive in these cases, thus not compatible with real-time performances. Applications where simulation interactivity is a fundamental requirement usually rely on modelling assumptions to simplify the deformation models or on numerical methods that approximate the exact solution, leaving accuracy aside for the sake of rapidity.

2.2.3 Challenge of stable simulations

An additional challenge of simulations involving deformable objects is represented by simulation stability. This property is desirable in all applications, since instability means a diverging solution, which is thus invalid and not usable. Visually, when a simulation becomes unstable,

the modelled object “explodes", due to undesired movement of the mesh vertices caused by the propagation of errors in the numerical methods governing the resolution of the equation system. In these cases, small perturbations in the input produce large variations in the obtained deformations.

Guaranteeing simulation stability is particularly challenging when the equation system to solve is stiff. In these kinds of systems, the upper bound of the time step is constrained by stability rather than accuracy requirements. This is generally the case in surgical simulations, where the objects to model are described by non-linear material laws, with anisotropic, heterogeneous and nearly-incompressible properties, which can generate ill-conditioned problems. While solving stiff systems, numerical integration schemes (i.e., those used to advance the simulation in time) might require impractically small time steps to guarantee stability. This happens, for example, with explicit integration schemes [21]. On the other hand, implicit methods allow to use larger time steps while remaining stable, but the equation system to solve at each time step is slightly more complex [20]. Several advanced integration schemes have been proposed to guarantee a convergent solution, i.e. simulation stability, to any kind of input. Schemes with this property are said to be unconditionally stable. However, they obviously come at the expese of an increased computational cost.

2.3 Continuum mechanics methods

The deformable behavior of soft tissues can be described with the laws of continuum mechanics. Each point in the undeformed reference configuration $\mathbf{X}$ is mapped to its corresponding point in the deformed configuration $\mathbf{x}$ (Fig. 2.2) through the deformation gradient $\mathbf{F}$ [31]:

$$\mathbf{F} = \frac{\partial \mathbf{x}}{\partial \mathbf{X}} = \mathbf{I} + \frac{\partial \mathbf{u}}{\partial \mathbf{X}} \tag{2.1}$$

where $\mathbf{u} = \mathbf{x} - \mathbf{X}$ is the displacement vector.

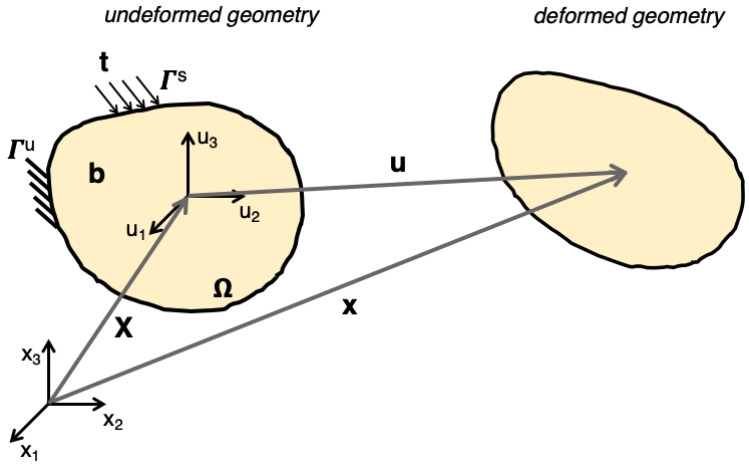


Figure 2.2: A body defined over the domain Ω, subject to a body force $\mathbf{b}$, surface traction $\mathbf{t}$ and boundary conditions on Γ, passes from its undeformed configuration $\mathbf{X}$ to a deformed configuration $\mathbf{x}$.

If a body defined over the domain Ω is in *static* equilibrium under an applied body force $\mathbf{b}$ (Fig. 2.2), imposing the balance of linear momentum leads to:

$$\nabla \cdot \boldsymbol{\sigma} + \mathbf{b} = 0 \quad in \quad \Omega \tag{2.2}$$

where $\nabla \cdot \boldsymbol{\sigma}$ represents the divergence of the Cauchy stress $\boldsymbol{\sigma}$. The Cauchy stress $\boldsymbol{\sigma}$ is a measure of the internal state of stress of the solid, which is related to surface tractions $\mathbf{t}$ via $\boldsymbol{\sigma}\mathbf{n} = \mathbf{t}$, where $\mathbf{n}$ is the outward unit normal to the surface.

Whereas, if the body is not in static equilibrium, its behavior is described by the dynamics laws of motion (Newton's second law). In this case, Eq. (2.2) must be extended to include the contribution of the inertia term, leading to:

$$\nabla \cdot \boldsymbol{\sigma} + \mathbf{b} = \rho\ddot{\mathbf{u}} \quad in \quad \Omega \tag{2.3}$$

where ρ is the object density. An additional term representing a damping force can be present.

Equations (2.2) and (2.3) are subject to some conditions on the boundary Γ of the object:

$$\begin{cases} \mathbf{u} = 0 \quad \text{on} \quad \Gamma_u \\ \boldsymbol{\sigma}\mathbf{n} = \mathbf{t} \quad \text{on} \quad \Gamma_s \end{cases} \tag{2.4}$$

where Γ_u and Γ_s are defined such that $\Gamma = \Gamma_u \cup \Gamma_s$.

Regardless of being in a static or dynamic situation, computation of realistic organ deformations using the continuum mechanics approach relies on the definition of the physical laws governing the mechanical response of the tissues to the applied sollicitations. Such laws are called constitutive laws and contribute to the definition of the internal forces. In particular, they put in relation the tissue stress, represented by $\boldsymbol{\sigma}$, to the tissue strain, described by the Green strain tensor $\mathbf{E}$, defined as:

$$\mathbf{E} = \frac{1}{2}(\nabla\mathbf{u} + \nabla\mathbf{u}^\top + \nabla\mathbf{u}\nabla\mathbf{u}^\top) \tag{2.5}$$

being $\mathbf{u}$ the displacement (i.e., the difference between the positions in the deformed and undeformed configurations).

Several different constitutive relations have been suggested in the literature, ranging from a linear law to very complex mathematical expressions [27, 32]. For elastic solids, the Cauchy stress is linearly related to the Green strain tensor by the elasticity matrix $\mathbf{D}$:

$$\boldsymbol{\sigma} = \mathbf{D} : \mathbf{E} \tag{2.6}$$

where ":" represents a contraction operation (i.e., $\sigma_{ij} = D_{ijkl}\epsilon_{kl}$ following Einstein summation convention[1]). In the elastic scenario, material behavior is fully described by only 2 characteristic constants: the Young's modulus E and Poisson's ratio ν. The simplest type of elastic materials are linear elastic materials, which are described by a linear relationship also between strain and displacement (i.e., neglecting the second order term in Eq. (2.5)). Although the linear elastic formulation is often preferred due to its computational convenience, it is not

[1]According to Einstein summation convention, when an index is repeated twice in a single term, it implies summation of that term over the range of the repeated index, e.g. $\mathbf{u} = u_j\mathbf{e_j} = u_1\mathbf{e_1} + u_2\mathbf{e_2} + u_3\mathbf{e_3}$, where $\mathbf{e} = [\mathbf{e_1}, \mathbf{e_2}, \mathbf{e_3}]$ represents a basis vector in 3D.

capable of handling large deformations correctly, thus it is not always adequate to describe tissues deformation [20].

A common way to describe more complex material behaviors is by means of a strain energy density function W. A material is called hyperelastic if W exists and the relationship between $\boldsymbol{\sigma}$ and $\mathbf{E}$ is obtained from:

$$\mathbf{S} = \frac{\partial W}{\partial \mathbf{E}} \tag{2.7}$$

$$\boldsymbol{\sigma} = J^{-1}\mathbf{F}\frac{\partial W}{\partial \mathbf{E}}\mathbf{F}^T \tag{2.8}$$

where $\mathbf{S}$ is the second Piola-Kirchhoff stress, $\mathbf{F}$ is the deformation gradient in Eq. (2.1) and J is the determinant of $\mathbf{F}$ [31].

The simplest hyperelastic formulation is the St. Venant–Kirchhoff, which represents an elastic material able to handle large displacements, since it uses the full Green strain tensor of Eq. (2.5). The strain density function describing the St. Venant-Kirchhoff material is given by:

$$W(\mathbf{E}) = \frac{\lambda}{2}[tr(\mathbf{E})]^2 + \mu[tr(\mathbf{E}^2)] \tag{2.9}$$

where tr indicates the trace of the tensor, λ and μ are the Lamé constants, and are uniquely related to the Young's modulus E and Poisson's ratio ν via

$$\lambda = \frac{\nu E}{(1+\nu)(1-2\nu)} \quad \mu = \frac{E}{2(1+\nu)} \tag{2.10}$$

Other popular constitutive models are the Neo-Hookean [33] and Ogden [34] model. In these cases, W is defined using the three principal stretches $\lambda_1, \lambda_2, \lambda_3$ (i.e., the three eigenvalues of the deformation gradient). The strain energy function for a Neo-Hookean material is defined as:

$$W(\lambda_1, \lambda_2, \lambda_3) = A(\lambda_1^2 + \lambda_2^2 + \lambda_3^2 - 3) \tag{2.11}$$

where A is a material constant, linked to μ by $A = \frac{\mu}{2}$.

The Ogden model has been widely used in rubber-like tissue modelling and it is described by:

$$W(\lambda_1, \lambda_2, \lambda_3) = \sum_{i+1}^{N} \frac{\mu_i}{\alpha_i}(\lambda_1 + \lambda_2 + \lambda_3 - 3) \tag{2.12}$$

where N, μ_i and α_i are material parameters, which must be fitted experimentally.

The system of equations describing the behavior of deformable objects through continuum mechanics laws can include four main types of nonlinearities:

1. *Geometric nonlinearity*: nonlinearity between strain $\mathbf{E}$ and displacement $\mathbf{u}$;
2. *Material nonlinearity*: nonlinearity between stress $\boldsymbol{\sigma}$ and strain $\mathbf{E}$;
3. *Force nonlinearity*: nonlinearity between force $\boldsymbol{b}$ and stress $\boldsymbol{\sigma}$ (when applied force depends on deformation);
4. *Kinematic nonlinearity*: nonlinearity between prescribed displacements (boundary conditions) and displacement $\mathbf{u}$;

Continuum mechanics laws cannot be solved analytically, thus requiring numerical solution schemes [31].

2.3.1 The finite element method

The preferred numerical scheme to solve continuum mechanics laws relies on the Finite Element (FE) method [31].

The FE method converts the system of partial differential equations describing the dynamic equilibrium motion equation Eq. (2.3) into a system of algebraic equations which can be solved numerically, after discretizing the domain both in space and time. Geometric discretization of the domain Ω is achieved by creating tissue 3D meshes, which describe the soft body as composed of elementary volumetric components (usually tetrahedral or hexahedral elements). The continuous solution $\mathbf{u}$ at any point in space is obtained by interpolating the values of the discretized displacements $\mathbf{u}_n$ at the element nodes using shape functions ϕ^n (i.e., polynomial functions that are generally linear or quadratic) [35]:

$$\mathbf{u} = \sum_{n=1}^{N_n} \phi^n \mathbf{u}_n \tag{2.13}$$

where N_n is the total number of nodes in the mesh. Each node n iin the mesh s associated to a shape function ϕ^n, which must have local support and piecewise continuous. By discretizing Eq. (2.3) with the FE method, we obtain this discrete problem on each element e:

$$-\mathbf{K}_e\mathbf{u}_e + \mathbf{f}^{ext} = \mathbf{M}_e\ddot{\mathbf{u}}_e \tag{2.14}$$

where $\mathbf{K}_e$ is the element stiffness matrix, $\mathbf{M}_e$ is the element mass matrix and $\mathbf{f}^{ext}$ is the external force applied to element e. The minus symbol in front of the stiffness matrix is added to explicitly formulate the equation as difference between external and intenal forces (making it more intuitive). If damping is present, the additional term $-\mathbf{C}_e\dot{\mathbf{u}}_e$ appears on the left handside of Eq. (2.14), where $\mathbf{C}_e$ is the element damping matrix. Global mass, stiffness and damping matrices of the system can be assembled from elementary ones, leading to a global system of equations that can be written in the form:

$$\mathbf{f}(\mathbf{x}, \mathbf{v}) = \mathbf{M}\mathbf{a} \tag{2.15}$$

with $\mathbf{a} = \ddot{\mathbf{u}}$, $\mathbf{v} = \dot{\mathbf{u}}$, $\mathbf{x} = \mathbf{u}$, and $\mathbf{f}(\mathbf{x}, \mathbf{v})$ represents the net force applied to the object (difference between external and internal forces $\mathbf{f}^{ext} - \mathbf{C}\mathbf{v} - \mathbf{K}\mathbf{x}$).

A time integration scheme (implicit, explicit or other variants) is then used to discretize the problem over time, allowing to formulate the above Eq. (2.15) as a linear system. The obtained set of linear equations can then be solved using either direct or iterative solvers. Direct solvers compute the solution exactly, either calculating the actual inverse or a factorization of the system matrix. Although these methods are often too costly, some optimized libraries exist that allow to parallelize these operations on CPU [36]. On the other hand, iterative solvers such as the conjugate gradient (CG) produce a sequence of approximate solutions approaching the exact one [37]. These methods can be very fast especially if they are tuned to stop when acceptable accuracy is reached, even if it is before convergence. However, they can converge slowly for ill-conditioned problems.

By relying on a physics-based description of soft tissues behavior, FE-based approaches can simulate anatomical responses to mechanical stimuli with high precision, given that a reliable constitutive model of tissue behavior and its mechanical parameters are available.

For this reason, the FE method represents the preferred approach to model the deformable anatomy [38]. Solution accuracy increases with finer domain discretization (i.e., higher number of elements), but at the cost of increased computational load. As a consequence, FE models allow to achieve highly accurate results whenever computation efficiency is not a major concern, e.g. deformable image registration in applications where a computation time up to several minutes is compatible with the clinical workflow [39, 40, 41].

In order to employ FE models in applications with more stringent time constraints, various techniques have been proposed to simplify their computational complexity. Some of them have focused on optimizing the solving process, proposing numerical solution approaches like matrix condensation [42] or pre-computing some quantities, as in [43]. The Total Lagrangian Explicit Dynamics (TLED) [44] reduces the computational load by precomputation of spatial derivatives and eliminating the need of iteratively solving a large system of equations. As an example, simulations based on the TLED formulation have demonstrated high efficiency in the context of image registration for neurosurgery [45]. However, the solution is only stable under very small time steps due to the use of explicit time integration. An improved version of TLED, which relies on direct Jacobian (DJ-TLED), has been recently proposed and demonstrated to achieve a significant speedup with respect to standard TLED on simulation of brain deformations [46]. Another optimized approach is the Multiplicative Jacobian Energy Decomposition (MJED) [47], which relies on decomposition of the strain energy into simple terms that can be precomputed.

One of the most popular modeling choice to obtain clinically acceptable accuracy levels while guaranteeing computational efficiency is to model soft tissues relying on the corotational formulation of linear elasticity. The corotational model takes advantage of the computationally efficient linear elastic formulation even to model large deformations, without introducing ghost forces [48, 49]. This formulation has been extensively used to model soft tissues in applications like intra-operative image guidance, e.g. in liver surgery [13, 50, 51, 52], brain surgery [53] and needle-based procedures [54, 55, 56, 57].

Some other works have accelerated the solving process by implementing the solvers on the Graphic Processing Unit (GPU), whose high number of cores allows to obtain significant speedups on computationally demanding problems [58, 59, 60, 61]. For example, NiftySim is a finite element toolkit that provides GPU-based implementation of TLED approach for high-performance biomechanical simulation [59]. However, in addition to requiring specific hardware, not all approaches are suitable for parallel computing on the GPU.

2.3.2 Alternatives to the standard finite element method

In addition to the optimized solution processes or formulations of the standard FE method mentioned above, some alternative approaches have been proposed to cope with the main limitations of FE models.

A possible strategy to lower the simulation time required by FE-based approaches is through Model Order Reduction (MOR) techniques [62]. The basic idea behind MOR is to reduce the number of degrees of freedom for the calculation of the solution by first creating a database of models obtained by precomputing the tissue response to different conditions and/or loads in an offline step; in the online phase, the solution to the current applied load is obtained by interpolating among the saved precomputed modes. The most popular MOR approach is Proper Orthogonal Decomposition (POD), where the solution to a high-dimensional problem

is encoded as a linear combination of few basis vectors defining the precomputed modes that best describes the behavior [63, 64]. The higher the number of modes considered to compute the solution, the higher the accuracy, but the higher the computational load as well. Although POD has been successfully used to achieve efficient simulations of non linear materials [65], it is not always able to capture complex non linear behavior of anatomical tissues [66]. A generalization of POD which is better suited for the simulation of hyperelastic materials is represented by proper generalized decomposition, where the solution is expressed as the sum of separable functions representing general solutions to the problem, thus does not rely on knowledge of complete-model solutions [67].

Other approaches have been proposed to avoid the 3D meshing process required by the standard FE method. In addition to being a time-consuming and tedious process, obtaining a high quality 3D mesh is very difficult due to the complex shape of anatomical organs, but it is essential to guarantee the accuracy of the results. Moreover, even when a high quality mesh is available, large deformations may lead to element distortion and, in turns, failure of the solution method.

The Boundary Element Method (BEM) is a numerical solution approach that deduces volume deformations from deformations at the surface of the object, thus allowing both to skip the 3D mesh generation process (only surface discretization is needed) and providing fast solution speeds [68]. However, since the internal behavior is not explicitly modelled, this approach only works for isotropic and homogeneous objects, preventing from its use in simulations involving topology modifications (e.g., cuts). The most popular application of BEM in surgery is for virtual simulators [68, 69].

Meshless methods represent a class of methods where the domain is discretized by a set of particles filling the volume of interest [70]. State variables at a given point are approximated by interpolating and weigthing variable values at neighbouring particles. However, some meshelss approaches still rely on grids of background cells for numerical integration [71, 72], thus not completely eliminating the finite element layer. Whereas, other approaches like smooth particle hydrodynamics interpolate the variables of interest directly on particles using a smoothing kernel, without the need of any background grid [73]. In general, the simplicity of meshless methods and their capability to handle changes of topology without remeshing are the main reasons for their employment for surgical simulation [74, 71]. However, the use of non linear materials might lower the computational performances, making meshless methods not always suitable for applications requiring fast update rates. Moreover, since they all rely on explicit time integration schemes, very small time steps are required for robust solution, which is a limitation in interactive scenarios.

2.4 Heuristic methods

Although modelling and simulation of anatomical deformations is traditionally performed using the FE method to reach high accuracy levels, solving continuum mechanics laws is generally too computationally expensive for certain applications. Heuristic methods introduce modelling assumptions that simplify the formulation to obtain more efficient methods, but sacrificing accuracy.

2.4.1 Mass spring models

Mass Spring Models (MSMs) consist of a set of masses distributed throughout a body and linked by a network of springs, as shown in Fig. 2.3. Deformation dynamics is governed by Newton's second law. Therefore, the position of the mass points is obtained by balancing the contributions of internal and external forces:

$$\mathbf{Ma} + \mathbf{Cv} + \mathbf{K}(\mathbf{x} - \mathbf{x}_0) = \mathbf{f_{ext}} \tag{2.16}$$

where $\mathbf{a}, \mathbf{v}, \mathbf{x}$ are, respectively, the acceleration, velocity and position each mass point, and $\mathbf{x}_0$ is the spring rest length. $\mathbf{M}$ is the mass matrix that contains the values of the point masses, while $\mathbf{C}$ and $\mathbf{K}$ are the damping and the stiffness matrices respectively, defining the damping and stiffness coefficients that describe the springs connecting each point to the points in its neighborhood. $\mathbf{f}_{ext}$ represents the external forces contribution. A time integration scheme is then used to discretize the problem in time, allowing to formulate the equation above as a linear system, which can eventually be solved exploiting numerical methods.

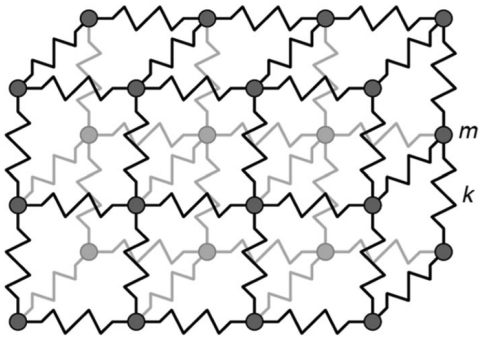


Figure 2.3: In MSMs, the object is modelled as an ensemble of lumped masses m connected via a network of springs k [75]. Image from [20].

MSMs have been widely used for their easy implementation and their high speed. In fact, MSMs do not need to preprocess all elements to provide an estimate of the global deformations like FEM, thus being much faster and able to simulate large deformations in near real time. One of the main issues of using this approach is that MSMs parameters are not directly related to tissues mechanical properties. Moreover, they are influenced by the distribution of the mass points and the arrangement of the springs. As a consequence, specific optimization methods are needed to tune model parameters for a given mesh to realistically approximate certain global mechanical behaviors [76, 77]. Enhanced versions of MSMs have been proposed to model sophisticated behaviors, for example by introducing nonlinear springs or using hybrid models [78]. However, only few of these approaches have been validated against real data, and simulation accuracy highly depends on the appropriateness of the employed optimizaion process. Owing to this, MSMs have been mainly used for medical training simulators, whose main requirement is computational efficiency, while accepting visually plausible deformable behavior [79, 80, 81].Furthermore, MSMs are advantageous to model complex tasks which involve topology modifications, such as cutting or suturing, since they can be implemented by simply removing or adding connections between vertices [82]. However, due to recent research advancements in computer graphics and computing

architectures, the usage of MSMs for surgical simulation is likely to be superseded by other deformable models that are more physically realistic, while also having competing efficiency [20].

2.4.2 Position based dynamics

Position Based Dynamics (PBD) is a simulation approach that computes the time evolution of a dynamic system by directly updating positions, as first described by Müller et al. in [83]. Simulated objects are discretized as sets of particles, described by their positions $\mathbf{p}_i$ and velocities $\mathbf{v}_i$, subject to a set of positional constraints $C_j(\mathbf{p}_1, ..., \mathbf{p}_n) \succ 0$ (symbol $\succ$ denotes either $=$ or $\geqslant$). In the PBD approach, deformation calculation becomes a constraint-function optimization problem. The simulation workflow starts with a prediction step in which simplectic Euler integration is performed to guess new particle positions and velocities. Then, non-linear Gauss-Seidel solver is used to find the correction $\Delta\mathbf{p}$ to apply to the estimated positions in an iterative fashion, so that each constraint equation (after linearization) is individually satisfied:

$$C(\mathbf{p} + \Delta\mathbf{p}) \approx C(\mathbf{p}) + \nabla C(\mathbf{p})\Delta\mathbf{p} \succ 0 \tag{2.17}$$

Since the resulting system is under-determined, the position update $\Delta\mathbf{p}$ is constrained to ensure the preservation of linear and angular momenta, which corresponds to forcing $\Delta\mathbf{p}$ to lie in the direction of the constraint gradient ∇C. The position update is further weighted by the inverse of the mass matrix $\mathbf{M}$ and multiplied by a parameter $k \in [0, 1]$ which represents the stiffness of the constraint:

$$\Delta\mathbf{p} = k\lambda\mathbf{M}^{-1}\nabla C(\mathbf{p})^T \tag{2.18}$$

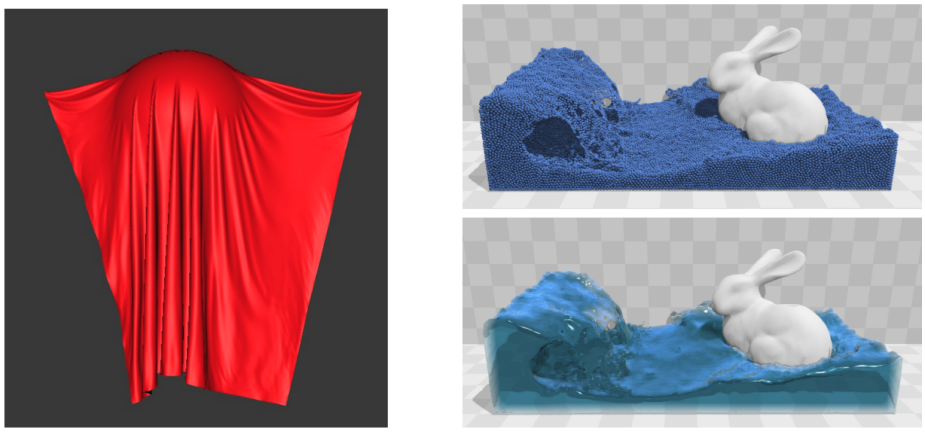
The Lagrange multiplier λ which solves Eq. (2.17) is thus unique and given by:

$$\lambda = \frac{C(\mathbf{p})}{\nabla C(\mathbf{p})\mathbf{M}^{-1}\nabla C(\mathbf{p})^T} \tag{2.19}$$

Finally, computed $\Delta\mathbf{p}$ are used to correct both the positions and the velocities. Different types of properties and behaviors can be simulated with PBD by implementing specific constraints and appropriately manipulating modelling parameters [84] (Fig. 2.4).

By directly updating particle positions, the PBD approach has the potential of achieving a better trade-off between accuracy and computation time with respect to methods based on continuum mechanics. In addition to improving computation performances, most of the constraints do not require the availability of a 3D mesh, but particles are generally placed in space to fill a surface-delimited volume. Further advantages of using particles to discretize objects emerge when topological modifications need to be modelled, since all element-related issues such as element inversion and distortion are avoided. Like all the heuristic models, the main difficulty PBD has to face is the choice of the model parameters, which lack of any connection with real mechanical properties.

Although the PBD approach has been mainly applied in computer graphic fields, the enhanced speed, controllability and unconditional stability of this method are its most appealing features for its application to medical simulation as well, where interactions between multiple organs and tools have to be modelled and solved in real-time [84]. The most popular exploitation of the PBD concept in the medical field has been in the development of training simulators for surgical procedures involving dissection, exploiting the capability of PBD



a b

Figure 2.4: Simulation of (a) cloth and (b) fluid using the PBD method, relying on different constraints. Images taken from [84].

methods to handle topological changes involved in such tasks maintaining real-time interactive performance [85, 86]. Some other works have focused on the achievement of particular types of deformations by coupling the PBD formulation with mass-spring models [87, 88] or for the simulation of knot tying procedure [89]. Camara et al. employed the PBD scheme to create a patient-specific biomechanical model of the kidney for the real-time simulation of intra-operative US images [90]. Some recent works have demonstrated that, whenever an ad-hoc parameter optimization strategy is designed, PBD-based models can achieve accuracy levels that make them suitable for surgical planning, for example in the context of kidney surgery [91], neurosurgery [92] and general laparoscopy [93].

2.4.3 Chain Mail

In the Chain Mail approach, the deformable object is discretized into a set of uniformly spaced elements, called *chain elements*, that are linked to each other as the links of a chain (Fig. 2.5) [94]. Each chain element enforces a bounding region for its neighboring chain elements. Volume deformations are obtained by adjusting the position of the chain elements while respecting the geometric constraints imposed by the neighbouring elements. After position adjustment, a relaxation scheme is employed to minimize the the internal energy of the

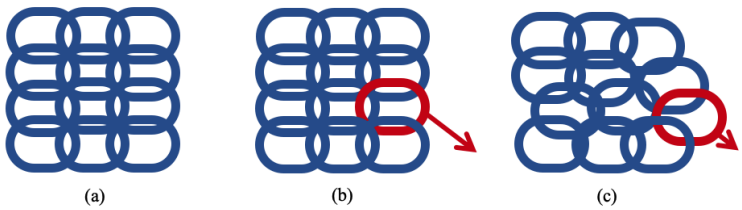


Figure 2.5: In ChainMail, the object is composed of a series of chain elements (a); as soon as an element moves due to an external input (b), the neighbouring elements move in turns to satisfy geometric constraints (c).

system, proportional to the displacement of the elements with respect to each other, until convergence. The chain-like structure guarantees the propagation of the deformations over the entire volume. Several works have proposed to extend the basic Chain Mail algorithm to model additional features, such heterogeneity, volume conservation or force-displacement relationship [20]. Similarly to PBD, due to its reliance only on position adjustments to obtain a volumetric deformation, Chain Mail is characterized by high computational efficiency, being able to handle deformations of large volumetric datasets [95, 96]. Advanced interactions can be easily modelled by removing or adding elements [97]. Generation of a 3D mesh is not required with this approach, since the volume is filled with regularly spaced chain elements, thus simplifying the model generation pipeline. However, its main drawback lies again in the selection of the parameters governing the geometric constraints (e.g., the margin of free mobility between the links) to achieve a realistic behavior. Owing to these features, Chain Mail models have been successfully employed in virtual surgery simulators, e.g. for prostate brachytherapy [98], endoscopy [99] and knee arthroscopy [100].

2.5 Machine learning based methods

Recently, Machine Learning (ML) has started to revolutionize several fields (vision, language processing, image recognition, genomics) due to the availability of larger and larger databases and the acceleration provided by powerful GPUs. ML comprises all those methods able to approximate a relationship between some input data and the output without any mathematical formulation of the problem, learning it directly from data. In general, ML approaches learn a function f that maps an input vector $\mathbf{x}$ to an output vector $\mathbf{y}$. The function depends on some parameters θ that are learnt from a huge amount of ground truth data during the so called training phase, such that:

$$f(\mathbf{x}, \theta) = \mathbf{y} \tag{2.20}$$

Then, in the inference phase, the learnt set of parameters θ is used to predict the value of $\mathbf{y}$ given a new observation $\mathbf{x}$. In general, ML based methods can achieve high inferring speed, which makes them useful for many applications where the prediction speed is a critical parameter.

Neural Networks Among the various ML techniques, the use of Neural Networks (NN) has gained increasing popularity. The basic unit of a NN is called neuron. In the simplest NN, called perceptron, each neuron applies a weight and an offset to each component of its input vector; obtained values are then summed and passed through an activation function that adds non linearity to produce the output. Multiple neurons stacked together in a row and receiving the same input constitute a layer. When multiple layers are placed next to each other, they generate a multi-layer neural network, whose basic structure is shown in Fig. 2.6.

The workflow of a multi-layer neural network with L hidden layers is detailed in Alg. 1. For each layer k, the input from the previous layer $\mathbf{a}^{[k-1]}$ is weighted by $\mathbf{W}^{[k]}$, a matrix whose number of rows matches the number of neurons in k-th layer and number of columns is the number of neurons in the layer $k-1$. A bias term $\mathbf{b}^{[k]}$ might also be added. ϕ represents the

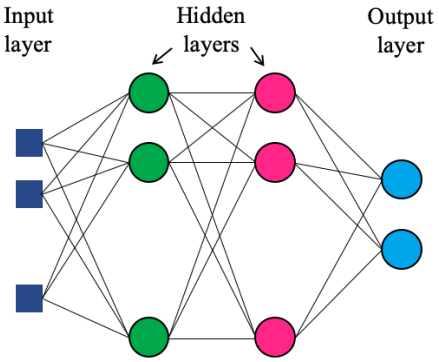


Figure 2.6: Basic structure of a multi-layer neural network.

activation function of the hidden layers, while ψ is the output layer activation function. The learnable parameters of such network is composed of the set of all the $\mathbf{W}^{[k]}$ and $\mathbf{b}^{[k]}$, which can be very high depending on the dimension of the network. Moreover, being the input of a vector $\mathbf{x}$, this approach might be inefficient when dealing with images or higher dimension inputs.

Algorithm 1 Mathematical formulation of a multi-layer neural network.

Input: Input vector $\mathbf{x}$, number of layers L
Output: Output vector $\mathbf{y}$
Init Current layer number $k = 0$, $a^{[0]}(\mathbf{x}) = \mathbf{x}$
while $k < L$ **do** ▷ *For all the hidden layers*
    $\mathbf{z}^{[k]}(\mathbf{x}) = \mathbf{W}^{[k]}\mathbf{a}^{(k-1)}(\mathbf{x}) + \mathbf{b}^{[k]}$
    $\mathbf{a}^{[k]}(\mathbf{x}) = \phi(\mathbf{z}^{[k]}(\mathbf{x}))$
    $k+ = 1$
$\mathbf{z}^{[k]}(\mathbf{x}) = \mathbf{W}^{[k]}\mathbf{a}^{(k-1)}(\mathbf{x}) + \mathbf{b}^{[k]}$ ▷ *Last layer*
$\mathbf{y}^{[k]}(\mathbf{x}) = \psi(\mathbf{z}^{[k]}(\mathbf{x}))$
return $\mathbf{y}$

Convolutional Neural Networks Deep Learning (DL) is a subfield of ML which has shown particular promise to model complex non linear relationships and has demonstrated strong abilities at extracting high-level representations of complex processes. Building blocks of DL are NNs with many multiple layers between the input and the output, creating the so called Deep Neural Networks (DNNs). One of the most popular class of DNNs is represented by Convolutional Neural Networks (CNNs), which have had an outstanding success especially in the computer vision field due to their ability to cope with high dimensional data such as images and videos [101]. A CNN contains several types of layers: convolutional layers,subsampling layers (e.g. pooling layers) and fully connected layers. As the name suggests, the building block of a CNN is represented by convolutional layers. This kind of layers repeatedly apply a filter to the input providing as output a map of activations called feature map. The feature map summarizes the presence of detected features in the input. In CNNs, filters coefficients are learned during training. The output of the convolutional layer is usually passed through a

non linearity, generally using the ReLU (REctified Linear Unit) activation function. Subsampling layers aim at scaling down the spatial dimension of the representation to reduce the number of parameters and computations in the model. It usually does it by taking the mean or the maximum on patches of the input (mean-pooling or max-pooling). Fully connected layers generally compose the last layer of a CNN. This layer is connected to all the activations of the previous layer and learns non linear combinations of the high-level features extracted by the previous convolutional layers. Fully connected layers can be followed by a final activation function depending on the application. The different CNN models which have been proposed rely on different number and arrangement of these main types of layers [101]. For example, Fig. 2.7 shows the architecture of LeNet, a CNN that has been used for digits recognition starting from an input image [102].

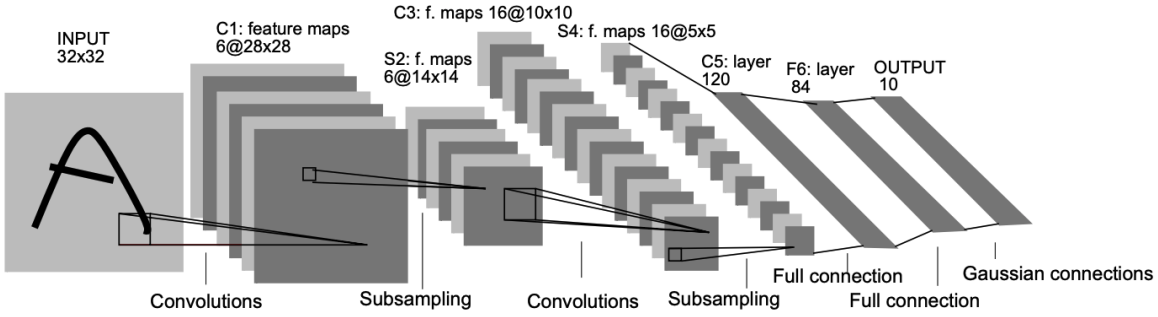


Figure 2.7: LeNet, a CNN for digits recognition starting from a 32×32 input image [102]. Convolutional, subsampling and fully connected layers typical of CNN architectures can be seen.

Some attempts that exploit learning methods to estimate the deformation of biological tissues have been made. By implicitly encoding soft tissue mechanical behavior in the trained ML models, they proved successful to predict the entire 3D organ deformation starting either by applied surface forces [103, 104, 105] or by surface displacements [106, 107, 108]. The greatest advantage of employing ML algorithms to simulate tissues deformation is the significant computational gain achieved with respect to the FE method, as highlighted in the recent survey by [109].

One of the main challenges of using a ML model to predict anatomical deformations is that its accuracy highly depends on the quality and on the amount of data used to train it. In an ideal scenario, such a model would be trained with an infinite amount of real patient-specific noise-free data, which is in practice not possible since acquiring large amounts of volumetric deformations of an organ is a challenging problem. As a consequence, most of the works which rely on ML to predict tissue deformations have trained their models using data generated with finite element simulations [109]. This approach has many advantages. First of all, being based on continuum mechanics theory, synthetic data generated with this strategy are highly realistic. Moreover, they allow to avoid all the ethical and legal issues connected with data acquisition on human subjects. Also, such data are automatically annotated and they can be augmented directly at simulation setup phase by using different parameter settings. Data generated with the FE method have been used by Lorente et al. to train several regression models that approximate the mechanical behavior of the liver during breathing [108]. Martinez et al. [110] have investigated the capability of tree-based methods to es-

timate breast deformation due to compression between biopsy plates. They have trained the proposed methods on different patient geometries but with very specific FE simulation representative of the scenario of interest, without including any randomness in the dataset. Neural networks have been used by Tonutti and Rechowicz to predict the displacement of brain tumors and of rib cage surface respectively, starting from the acting forces [104, 105]. However, both these works do not predict whole volume deformation but only surface displacements. Being trained with simulated data from a single geometry, the generated ML algorithm acts as a patient-specific model. However, this implies that network re-training is required for every new patient. Methods based on NNs have been also used to predict tissue deformations in augmented surgery. Morooka et al. have trained a NN to predict liver deformations for a given input force [103]. In their model, they have used Principal Component Analysis (PCA) to compress the size of the deformation modes, which allows to reduce the number of neurons of the output layer, thus the training time. Other works have estimated liver deformations from the known displacement of a partial surface [111, 106, 107, 14]. Among these, the works of Pfeiffer et al. are able to generalize to new patients since models have been trained on several different random geometries of the organs [106, 14].

From all these works, it has emerged that the main advantage of using neural networks to predict anatomical deformations relies on a prediction speed in the order of few milliseconds, which is not affected by the complexity of the model used to generate the dataset. The high inferring speed makes this method useful for many applications such as surgical navigation and real-time simulation.

2.6 Conclusion

Several different methods have been proposed to simulate the deformable behavior of the soft anatomical tissues, each achieving a different trade-off between accuracy, computation efficiency and numerical stability. Although it would be ideal to satisfy all these requirements, such a model does not exist due to the complexity to model the anatomical environment. Therefore, the choice of the approach to use depends on the specific requirements of the application of interest. In this Chapter, we have provided an overview of the main deformable models available for surgical simulation, providing some example applications.

Methods based on continuum mechanics rely on numerical methods to directly solve the physics equations describing tissues deformations. The most popular solution process exploits the finite element method, where the global tissue behavior is obtained by assembling the response of elementary components in which the tissue is discretized. Although methods based on continuum mechanics can model very complex behaviors, e.g. non linear materials, heterogeneous tissues and anisotropy, numerical solution of such complex equations via the finite element method is computationally expensive and cannot take place in real time. As a consequence, these methods have been generally used for applications where the real-time performance is not a strict requirement. Even though advanced and optimized formulations have been proposed, the improved computational performance is generally achieved by sacrificing the accuracy to some extent.

On the other hand, heuristic methods rely on simplified models that provide optimized computational performance, but they cannot be easily extended to represent all the possible mechanical behaviors of tissues. This modelling approach suffers from ambiguity in speci-

fication of model parameters to reproduce tissues mechanical behavior, since a well-defined relationship between model parameters and material constitutive law does not exist. This means that an ad-hoc optimization process is always required to tune model parameters such that an acceptable level of accuracy is achieved. However, whenever reliable parameters are selected, these models can achieve reasonable levels of realism. Due to the simplicity of the models and the fact that they do not generally rely on 3D meshes, this category of methods has been mainly employed for virtual training simulators, which involve topological changes.

The last category of methods takes advantage of machine learning to predict tissue deformations. In fact, a trained ML algorithm significantly reduces the time required to provide an estimate of tissue deformed state starting directly from data, avoiding the computationally expensive resolution of complex system of equations. In order to cope with the lack of real tissue deformation data, the majority of works that predict tissue deformations with ML have relied on training datasets generated with finite element simulations. ML algorithms have proved able to learn complex deformable behaviors reaching the same level of accuracy of the biomechanical simulations used to generate the data while obtaining considerable time gains, passing from minutes (required for the finite element method) to milliseconds (when using ML). Despite being used mainly for applications with strict time requirements such as augmented reality, this approach seems particularly promising and applicable to a wide variety of applications.

Part I

Learning patient-specific models

Chapter 3

Learning patient-specific pre-operative models

3.1 Introduction

In the context of autonomous robotic surgery, the availability of a patient-specific simulation that can account for organ deformations due to surgical manipulations is fundamental to enable tailoring and testing of the surgical plan before its execution on the real environment. Such simulation accounts for each patient's characteristics by relying on a Patient-specific Biomechanical Model (PBM) created from diagnostic images and data available before the intervention (Fig. 3.1). In particular, a PBM is a model that includes (i) the 3D geometry of the anatomy of interest, extracted from each patient's diagnostic images, and (ii) the parameters defining its mechanical behavior, either obtained from imaging techniques, from ad-hoc optimization strategies or from values from the literature.

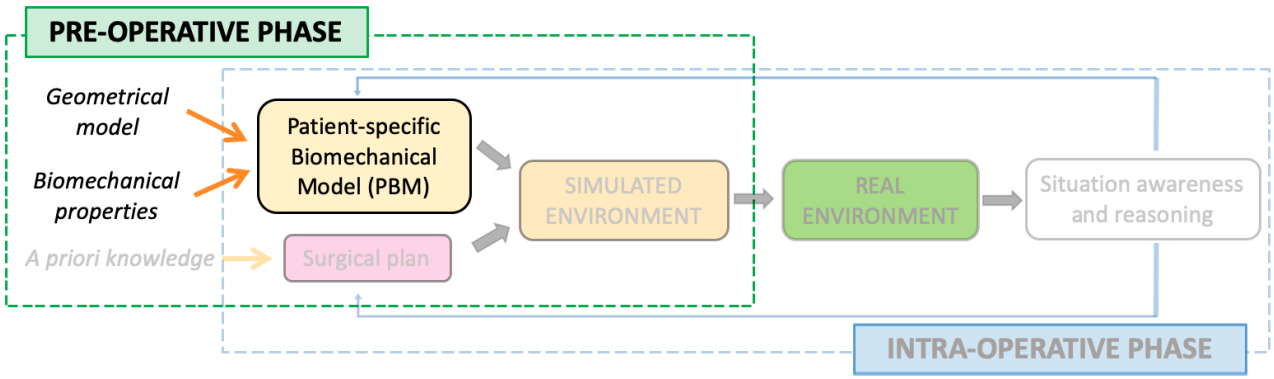


Figure 3.1: A PBM is characterized by both the 3D geometry of the anatomy of interest and its biomechanical properties. The PBM is created in the pre-operative phase from available data from diagnostic images and a-priori knowledge. The generated PBM drives the simulation of the environment that supports the ARSS.

As emerged in Chapter 2, there are several methodologies that can be used to simulate anatomical behavior. The aim of this Chapter is to idenfity the modeling approach which allows to achieve the best compromise among simulation accuracy, computation time and

stability, when simulating the interaction of a rigid tool with the deformable anatomy, the simplest but most common interaction. In particular, we first consider the three main strategies for modeling soft-rigid interaction when the personalized simulation relies on the finite element method. Then, we consider a heuristic model, i.e. position based dynamics, and we propose a strategy for the identification of patient-specific parameters. Finally, we develop a machine learning based approach that exploits a convolutional neural network to predict tissue deformations and is trained with simulated data only, and we evaluate its performance with respect to the other aforementioned methods.

The different approaches considered in this Chapter are tested in the context of robotic ultrasound (US) scanning of the breast, where the tool-tissue interaction is the one taking place when the US probe comes in contact with the breast tissues. This scenario has been selected because it combines all the aspects that make the real-time simulation very challenging, being the breast one of the most difficult structures to model due to the huge non-linear deformations it undergoes during the scanning process. Furthermore, this application has been chosen for its clinical relevance, since the development of autonomous robotic ultrasound systems for the breast represents a very active and promising field [112, 113, 114, 115, 116, 117].

Even though we consider breast US scanning for performance assessment, it is important to highlight that the considered methodologies can be applied to describe general tool-tissue interactions, and the obtained results are thus valid for modeling anatomical deformations induced by the interaction with any medical tool.

3.2 Deformation modeling in breast US scanning

Ultrasound (US) imaging is extensively used in several routine procedures, mainly due to its cost-effectiveness, non-invasiveness and real-time capabilities. Its main limitation is the low image quality, which highly depends on proper acoustic coupling between the probe and the tissues. The identification of the optimal transducer positioning that allows to obtain acceptable image quality heavily relies on the radiologist' expertise and requires the sonographer to apply a certain level of compression to the anatomy, which can reach several centimeters depending on the imaged tissue [118, 119, 120].

The US scanning process would benefit from the availability of a model that describes the deformations induced on the anatomy by the US probe. Such model can provide support to image fusion techniques commonly used for intra-operative tracking of interal regions identified on high-resolution pre-operative images (MRI/CT) but not visible on US due to the poor image contrast. Several commercial and research platforms have implemented image fusion techniques that align pre-operative and intra-operative data exploiting rigid or affine registration methods [121]. However, such approaches fail when dealing with highly deforming anatomies, where accurate tracking has to compensate for the large deformations arising due to compression forces applied by the US probe [118]. Moreover, a model of probe-tissue interaction can also be exploited to correct for deformations in the 3D US reconstruction process [119, 122, 120, 123]. Eventually, the development of computer-based ultrasound training systems that allow radiologists to practice the scanning technique have to realistically simulate probe-induced deformations in real-time [90, 124]. Further to enhancing image registration and guidance, a deformation model can support the development, test

and validation of robotic ultrasound systems (Fig. 3.2), which have the potential to improve the performance of manual acquisition systems, due to the high precision, dexterity and repeatability that robotic manipulators can bring [125, 126, 123, 127]. Without suffering of any human factors like tremors or lacks of concentration, robotic US systems enable a precise handling of the transducer and can thus follow a pre-defined path.

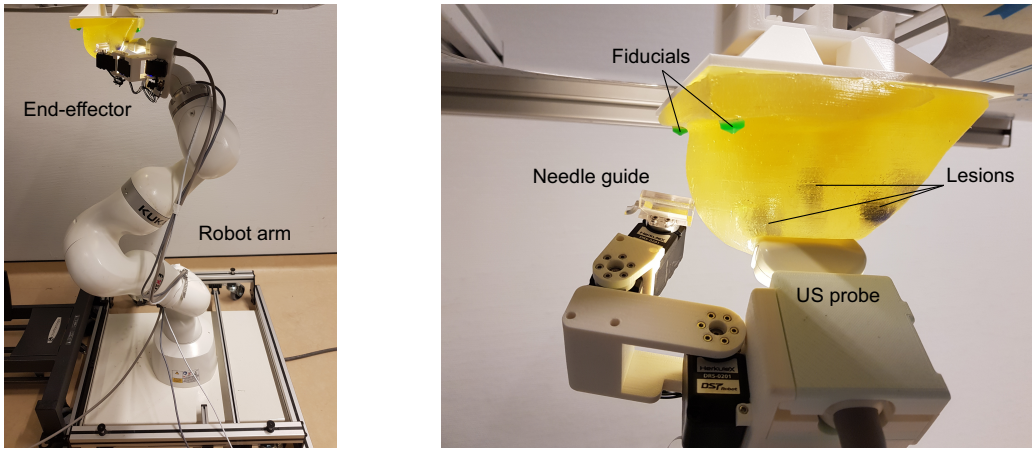


Figure 3.2: Robotic ultrasound system developed within the MURAB project [115].

Ultrasound scanning is commonly employed to provide real-time image guidance during biopsy procedures, where a needle is directed towards screening-detected suspicious lesions to evaluate their malignancy [128]. For example, US-guided biopsy represents the preferred technique for breast cancer diagnosis. However, proper needle placement in the breast with US is a challenging task due to its highly deformable nature, making the availability of an accurate model of the deformations particularly interesting to enable tracking of biopsy targets identified on pre-operative images. Such model is also fundamental for the design and test of autonomous robotic US systems for the breast (Fig. 3.3), a very active and promising field [113, 112, 114, 115, 116, 117].

Accurate modeling of the probe-tissue interaction for the breast is a complex problem, since it combines all the aspects that make the real-time simulation very challenging, due to the high non-linear deformations the breast undergoes during the scanning process. Biomechanical models relying on the FE method have been successfully employed to describe breast behavior for different applications [130, 61, 131, 132, 133]. Although both linear and non-linear models have been proposed, nonlinear materials have shown superior performance in terms of accuracy [134]. However, such models suffer from high computational cost, which prevents their employment for applications requiring real-time performances [61]. An efficient approach to model breast biomechanics is the one proposed by Han et al. in [135], which relies on both TLED formulation and a GPU-based solver. Despite the significant simulation speedup achieved, solving the FE system took around 30 s, which is still not compatible with real-time.

An emerging approach which has the potential of being both accurate and fast, exploits neural networks to estimate soft tissue behavior. The potentiality of applying machine learning techniques to predict breast deformations has been already shown in [110], where several tree-based methods have been employed to estimate breast deformation due to compression between biopsy plates. These methods have been trained on 10 different patient geometries

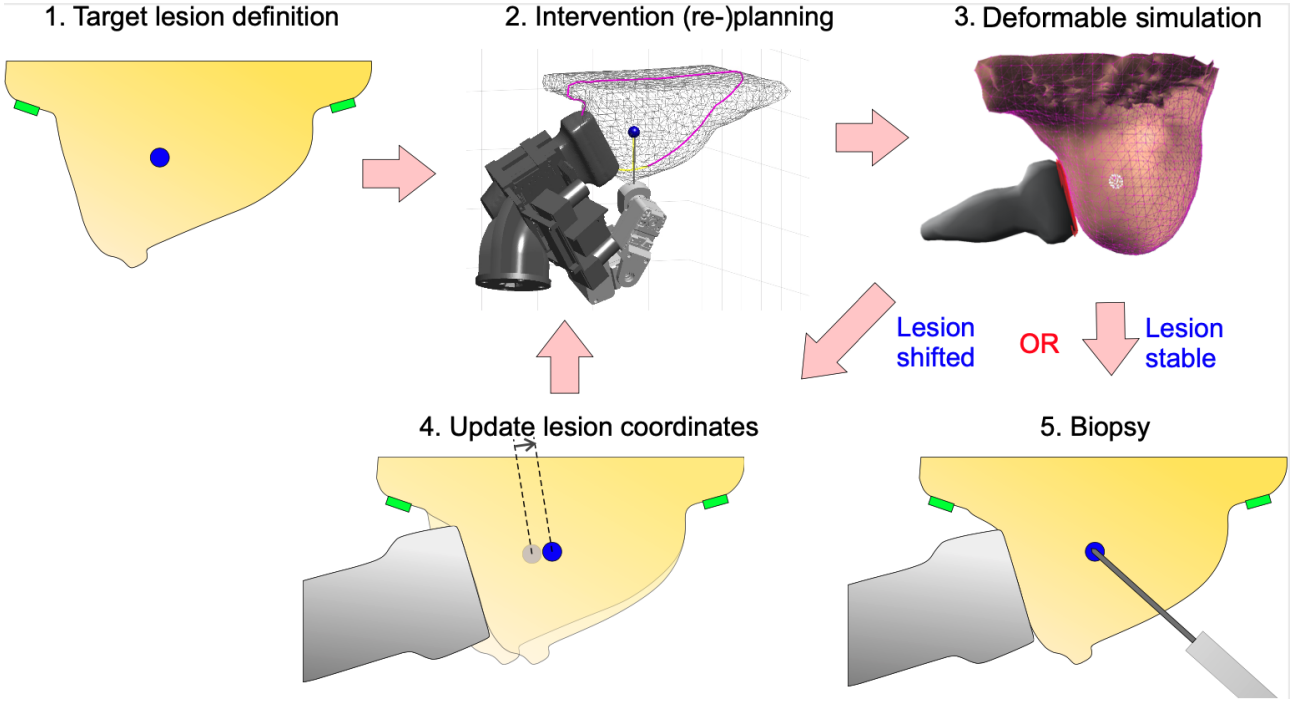


Figure 3.3: Steps for planning autonomous robot US scanning of the breast, as done in the MURAB project [115, 129]. The presence of a deformation model supports the robot planning phase, allowing to keep track of the lesion position due to the compressional effects induced by the US probe.

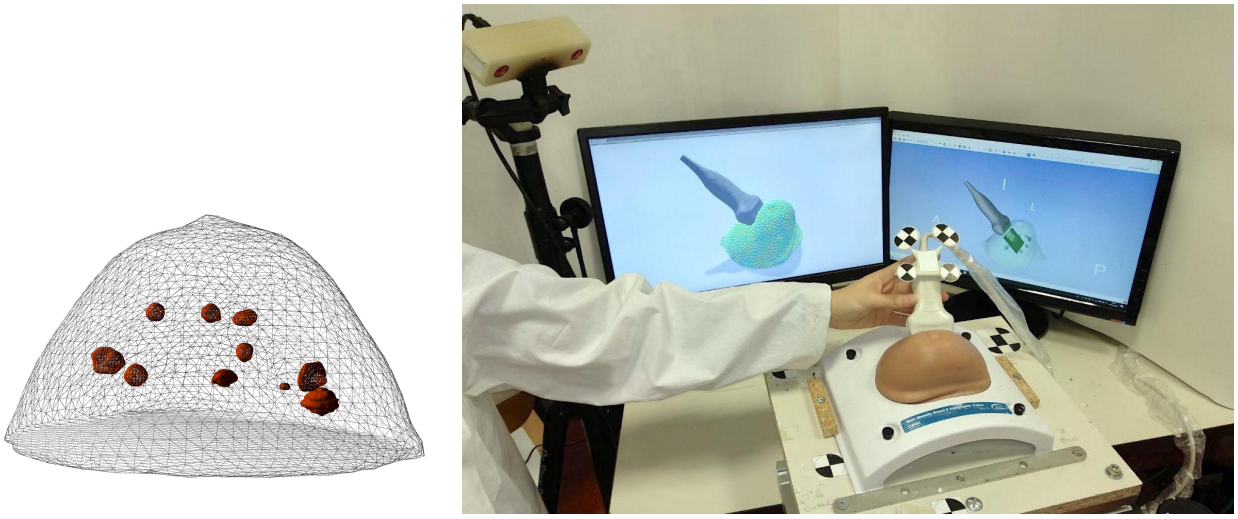
with a very specific FE simulation, where the upper plate is displaced vertically towards the lower one.

3.3 US scanning of a breast phantom

In this Chapter, we evaluate different approaches to model the deformations induced on a deformable tissue by a rigid tool, on the specific task of US scanning of the breast.

3.3.1 Experimental setup

We consider freehand US scanning of a realistic multi-modality breast phantom with some internall stiff masses (Model 073; CIRS, Norfolk, VA, USA). The 3D geometry models of the phantom surface and 10 inner lesions (diameter of 5-10 mm) are obtained by segmenting the corresponding CT image, relying on ITK-SNAP and MeshLab [136, 137] (Fig. 3.4a). A Freehand Ultrasound System (FUS) based on a Telemed MicrUs US device (Telemed, Vilnius, Lithuania) equipped with a linear probe (model L12-5N40) is used to acquire US images of the 10 segmented lesions. The dimension of the probe surface is (50x10 mm). The Micron-Tracker Hx40 (ClaronNav, Toronto, Canada) optical tracking system is used to track US probe in space (Fig. 3.4b). We perform all experiments with an acquisition frequency of 5 MHz and a depth setting of 50 mm, while all the other parameters are kept to default values provided by the manufacturer. The spatial and temporal calibration methods used in the study



a b

Figure 3.4: (a) External surface and 10 inner lesions of the CIRS breast phantom. (b) Experimental setup. The FUS system allows to map the real positions of the CIRS breast phantom and the US probe, as well as the US image acquired in real-time, to the 3D Slicer scene (right monitor). Information about probe spatial transformation is communicated to the simulated environment (left monitor).

are based on the PLUS toolkit, a software and hardware framework for building research FUS [138]. The overall probe spatial calibration error is below 1 mm ($\pm$0.7147), and below 0.5 mm ($\pm$0.334) for the pointer used for fiducial points localization required for the rigid registration. Thanks to the FUS, we can know in real-time the position and orientation of the US image plane and therefore extract three-dimensional position of any pixel belonging to the image. Landmark-based rigid registration is performed to refer the CT-extracted 3D model, the US probe and the US images to the same common coordinate system, exploiting 3D Slicer functionalities [139]. Thanks to this registration process, knowledge about the 3D pose and the geometry of the US probe directly allows to identify the contact surface between the breast and the probe. The registration process does not only enable us to extract the area on the breast surface which is in contact with the probe, but also to know in real-time the 3D position of any point belonging to the US image. In this way, it is possible to refer lesions position extracted from US images to the 3D space, which represent ground truth 3D positions of the lesions.

In our experiments, we consider one lesion at a time and we reposition the US probe on the surface of the breast such that the considered lesion is visible on the US image. In order to validate our model, we manually extract lesions position from US image acquired at rest (i.e., without applying any deformation, when the probe is only slightly touching the surface) and we consider it as a landmark to track. We then impose four deformations of increasing extent (5 mm, 10 mm, 15 mm and 20 mm) for each lesion, and save the corresponding US images. Ground truth lesion positions at the different deformation levels are then converted from the US image space to the 3D FUS coordinate system leveraging the registration described before.

3.3.2 Modelling assumptions

In order to consistently compare the different methods, we model the scenario relying on the same assumptions and approximations:

1. We consider patient-specific models, initialized with optimized parameters that account for the specific properties of the considered breast phantom.
2. The breast is modelled as a homogeneous object, despite the presence of stiffer internal parts. We assume this approximation to hold in our setup since stiff masses are small and inserted in a homogeneous material, similarly to [140, 130].
3. Input of the methods is always represented by a displacement, to ensure the widest applicability and generalisation capability of the methods. We do not consider scenarios where deformations are driven by forces, which would require force sensing apparatus, difficult to incorporate within standard clinical devices [119].
4. The US probe is modelled as a rigid body, which moves at fixed velociy of 0.01 m/s^2.
5. As boundary conditions, all the points belonging to the lowest phantom surface are constrained in all directions.
6. Gravity load is not explicitly considered, since the geometry model is already acquired within the gravitational field (being extracted from a CT image).
7. The same fixed time step is used for all the simulations ($h = 0.02\,s$).

3.3.3 Evaluation metrics

The different modeling approaches considered, which will be presented in next Sections, are evaluated with respect to the three main performance criteria, i.e. accuracy, time and stability. To analyze the results, input deformations are classified into five ranges (D10, D15, D20, D25, D30) based on the probe displacements. This is needed since applying deformations of the same exact extent is almost impossible during freehand US. Displacement ranges indicated as D15, D20 and D25 have a fixed length of 5 mm each and are centered at 15, 20 and 25 mm respectively. D10 and D30 contain the extreme cases under 12.5 mm or above 27.5 mm.

For each lesion, at each deformation level, accuracy is evaluated by comparing model-predicted lesion positions $\mathbf{X_{model}}$ with the real lesion coordinates in the 3D space $\mathbf{X_{US}}$, extracted from US images and referred to the 3D space thanks to the tracking system described in Section 3.3.1. Localization error relative to the considered models ϵ^{model} at deformation l and tumor m is computed as:

$$\epsilon_{l,m}^{model} = ||\mathbf{X_{model}}(l,m) - \mathbf{X_{US}}(l,m)||_2 \tag{3.1}$$

where $||.||_2$ is the Euclidean distance.

The performances of the presented methods are also evaluated in terms of computation time. To do so, we keep track of the computation time t_{comp} needed to perform a simulation time step h. When dealing with dynamic simulations, one usually evaluates the capability of a method to meet the real-time requirement (i.e. $t_{comp} \leqslant h$) and/or to guarantee interactivity,

which translates into ensuring that simulation runs at least at 25 frames per second (i.e. $t_{comp} \leqslant 0.04\,\text{s}$), when only visual feedback is required.

As a final metric, we assess stability of the methods by evaluating their capability to complete the experiments from the beginning to the end. For this analysis, we use the word "experiment" to refer to the process involving the application of the four increasing input deformations. This means that we perform one experiment per tumor, for a total of 10 (i.e., number of lesions) experiments. For each method, at each discretization level, we evaluate the percentage of experiment which is successfully accomplished for each tumor, before the occurrence of any instabilities. If a method was able to complete all the 10 experiments, the associated average percentage would be 100%.

3.4 The finite element method

The first class of methods that we consider to model the breast deformations induced by the US probe during the scanning process relies on the Finite Element (FE) technique. As introduced in Section 2.3.1, the FE method is a solution technique that converts the system of partial differential equations describing the dynamic equilibrium motion equation (Eq. (2.3)) into systems of algebraic equations of the form:

$$\mathbf{M}\mathbf{a} = \mathbf{f}(t, \mathbf{x}, \mathbf{v}) \tag{3.2}$$

where $\mathbf{a}, \mathbf{x}, \mathbf{v}, \mathbf{f}$ are the acceleration, position, velocity and force (both internal and external) vectors respectively, and $\mathbf{M}$ is the mass matrix. In the solution process, such equations are discretized both in space, relying on a 3D mesh that represents the domain of interest, and time, using a time integration scheme. For this analysis, we perform numerical integration using a backward Euler scheme, which offers a good trade-off between robustness, convergence and stability. Velocities and positions are updated based on accelerations at the end of each time step h:

$$\mathbf{v}^{t+h} = \mathbf{v}^t + h\mathbf{a} \qquad \mathbf{x}^{t+h} = \mathbf{x}^t + h\mathbf{v}^{t+h} \tag{3.3}$$

$$\mathbf{M}\mathbf{a} = \mathbf{f}(\mathbf{x}^{t+h}, \mathbf{v}^{t+h}) \tag{3.4}$$

We consider the first order approximation of $\mathbf{f}$ (one per time step):

$$\mathbf{f}(\mathbf{x}^{t+h}, \mathbf{v}^{t+h}) \approx \mathbf{f}(\mathbf{x}^t, \mathbf{v}^t) + \mathbf{K}(\mathbf{x}^{t+h} - \mathbf{x}^t) + \mathbf{C}(\mathbf{v}^{t+h} - \mathbf{v}^t) \tag{3.5}$$

where $\mathbf{K}$ is the stiffness matrix and $\mathbf{C}$ the damping matrix. Substituting (3.3) and (3.5) into (3.2) provides the final linearized system:

$$\underbrace{\left(\mathbf{M} - h\mathbf{B} - h^2\mathbf{K}\right)}_{\mathbf{A}}\,\mathbf{dv} = \underbrace{h\mathbf{f}(\mathbf{x}^t, \mathbf{v}^t) + h^2\mathbf{K}\mathbf{v}^t}_{\mathbf{b}} \tag{3.6}$$

where $\mathbf{dv} = h\mathbf{a} = \mathbf{v}^{t+h} - \mathbf{v}^t$. The obtained set of linear equations is solved for $\mathbf{dv}$ using either direct or iterative solvers.

Starting from these equations, there are three main ways to model the interaction between a rigid object and a deformable model: (i) imposing penalty forces [124], (ii) describing the contact as a constraint [141] or (iii) directly displacing the surface nodes [118, 122]. In the following, the details of each approach are provided.

3.4.1 Penalty method

In the penalty method (*Penalty*), contacts are solved by applying a spring-like force $\mathbf{f_{pen}}$ proportional to the amount of penetration δ at each contact point, in the direction $\mathbf{n}$ normal to the surface [142]:

$$\mathbf{f_{pen}} = k_s \delta \mathbf{n} \tag{3.7}$$

This force is treated as an external force and contributes to the right hand side of (3.2). The higher the value of the proportionality coefficient k_s, called contact stiffness, the better the constraint is satisfied. However, large values of k_s make the condition number of the system matrix $\mathbf{A}$ worse, often causing problems in convergence and instabilities in the simulations. The selection of k_s is also problem-dependent, and heavily depends on the ratio of the material stiffness between the contacting objects, making this method limited for our applications. Despite these stability issues, the penalty method is the easiest to implement and can be very fast.

3.4.2 Lagrange multipliers method

Differently from the penalty approach, methods based on Lagrange multipliers (LM) allow to solve the contact condition exactly by treating contacts as constraints [143]. The equation system (3.2) is extended to include constraints contribution $\mathbf{H}^T\boldsymbol{\lambda}$:

$$\mathbf{Ma} = \mathbf{f}(t, \mathbf{x}, \mathbf{v}) + \mathbf{H}^T\boldsymbol{\lambda} \tag{3.8}$$

which, after integration, leads to:

$$\mathbf{Adv} = \mathbf{b} + \mathbf{H}^T\boldsymbol{\lambda} \tag{3.9}$$

In this case, the contact force is represented by the unknown vector $\boldsymbol{\lambda}$ of LM, which imposes the impenetrability condition defined by Signorini's law [144]. Defining $\boldsymbol{\delta}$ as the vector of distances between contacting objects, two states are possible for each potentially colliding point P according to the Signorini's formulation:

1. either the point P is a contact point, then $\boldsymbol{\delta}(P) = 0$ and $\boldsymbol{\lambda}(P) \geqslant 0$
2. or the point P is not a contact point, then $\boldsymbol{\delta}(P) \geqslant 0$ and $\boldsymbol{\lambda}(P) = 0$

The equation system to solve becomes more complex since, at the beginning of each time step, both the multiplier values and the new positions are unknown. The solving process involves three main phases:

Free motion A free configuration $\mathbf{dv}_i^{\text{free}}$ is obtained for each interacting object (in the following, i denotes the index of simulated body) by solving the corresponding equations (3.9) independently, and setting $\boldsymbol{\lambda} = 0$.

Collision detection The free motion results in new configurations of bodies, making it necessary to detect possible collisions. The output of this phase is represented by the constraint matrices $\mathbf{H}_i$ and actual violations of the constraints $\boldsymbol{\delta}^{\text{free}}$ due to the free motion.

Collision response After linearization of the constraint laws [144], we obtain:

$$\boldsymbol{\delta} = \boldsymbol{\delta}^{\text{free}} + h \sum_i \mathbf{H}_i \mathbf{dv}_i^{\text{corr}} \tag{3.10}$$

With $\mathbf{dv}_i^{\text{corr}}$ being the unknown corrective motion ($\mathbf{dv} = \mathbf{dv}^{\text{free}} + \mathbf{dv}^{\text{corr}}$) when solving Eq. (3.9) with $\mathbf{b}_i = 0$. By gathering (3.9) and (3.10), we get:

$$\boldsymbol{\delta} = \boldsymbol{\delta}^{\text{free}} + \underbrace{h \left[\sum_i \mathbf{H}_i \mathbf{A}_i^{-1} \mathbf{H}_i^T \right]}_{\mathbf{W}} \boldsymbol{\lambda}. \tag{3.11}$$

We obtain the value of $\boldsymbol{\lambda}$ using a projected Gauss-Seidel algorithm[145] that iteratively checks and projects the various constraint laws [146]. Finally, the corrective motion is computed as follows:

$$\mathbf{x}_i^{t+h} = \mathbf{x}_i^{\text{free}} + h\mathbf{dv}_i^{\text{corr}} \quad \text{with} \quad \mathbf{dv}_i^{\text{corr}} = \mathbf{A}_i^{-1} \mathbf{H}_i^T \boldsymbol{\lambda} \tag{3.12}$$

The LM approach is the method of choice to obtain a stable and robust handling of contacts, but to the detriment of computational performances. Another advantage of this method is that interaction forces are accurately estimated, which can be very helpful within robot control loops, provided that the real mechanical properties of the organ of interest are known.

3.4.3 Prescribed displacements method

The last method considered models probe-tissue interaction as a Dirichlet boundary condition on the organ surface (*PrescrDispl*). Due to the fact that the US probe is represented as a rigid body, we can assume that when the anatomy is deformed during the scanning process, points on the organ surface below the US probe are displaced of the same exact amount as the probe itself. This modeling strategy is less general than the previous two approaches, since it relies on two major assumptions: (i) probe motion is completely known a-priori, which allows us to pre-compute contact points, and (ii) the contacting surface does not change during the scanning, i.e. no relative motion exists between the contact surfaces. From a formulation point of view, this approach is very similar to the penalty method described in 3.4.1, except that it does not require the collision detection phase. Despite this fact limiting its general applicability, it makes the model promising to achieve high computational performances. Furthermore, it represents a displacement-zero traction problem and as such it has the advantage that it does not require patient-specific mechanical properties [147].

3.4.4 FE-based breast model

In this analysis, the breast is modelled as a Neo-Hookean hyperelastic material (Eq. (2.11)), with Young's modulus and Poisson's ratio set in accordance with the values provided by Visentin et al. in [140], which are estimated on the same multimodal phantom used in this work. The choice of hyperelastic formulation is motivated by its popularity in breast biomechanical modeling and the fact that linear elasticity would not have been able to cope with the large input deformations applied. However, it is worth noticing that linear elastic or corotated

formulations usually represent the preferred modeling choice for problems explicitly modeling contacts between objects, to avoid introducing additional complexity [118, 124, 60]. In fact, the combination of highly irregular geometries, non-linear constitutive models and contacts can lead to ill-conditioned problems and introduce significant numerical issues, often causing instabilities in the simulations. Nevertheless, in this work we have to rely on hyperelastic constitutive laws in order to accurately describe breast behavior. To guarantee high efficiency, we perform all FE-based simulations within SOFA framework, the state-of-the-art engine for interactive FE-based medical applications [148]. Being SOFA open source, it is possible to implement all three approaches, thus allowing to compare the performance of the different modeling strategies within a common framework. Moreover, by choosing SOFA we had the possibility to make our implementations publicly available to the community[1].

In order to thoroughly evaluate the considered methods, all the simulations are repeated considering increasing levels of spatial discretization (i.e., volume mesh resolution). Due to the fact that mesh resolution is one of the variables most influencing the performances of FE-based simulations, the introduction of this additional variable in the evaluation allows us to estimate the relative impact that the choice of the method and the discretization level have on the performances. In all cases, collision detection is performed using the default pipeline provided by SOFA [148] on a surface mesh composed of 1,004 triangles, which proves able to maintain a good accuracy while keeping the number of active constraints to a minimum. The conjugate gradient algorithm [37] is used to solve the system of equations of *Penalty* and *PrescrDispl* methods. The maximum number of allowed CG iterations is set to 25, a value which proved able to speed up the solving process while keeping the simulation accuracy aligned with that of the other methods. Simulations relying on *LM* method require the use of a direct solver, for the computation of the matrix $\mathbf{W}$. To this purpose, we exploit the state-of-the-art solver Pardiso, whose multithreading implementation allows to achieve enhanced performance [36].

3.5 Position Based Dynamics

Instead of predicting volume deformations based on time integration of Newton's second law, the position-based dynamics approach models objects as an ensemble of particles whose positions are directly updated, as a solution of a quasi-static problem subject to geometrical constraints, as described in Section 2.4.2. Simulation behavior and performance are not only influenced by the relative position, dimension and number of particles in space, but also by the constraints acting among particles. Large deformations of soft bodies are usually achieved by defining positional constraints among rigid clusters of adjacent particles. This kind of constraint is called *region-based shape matching*. For all the particles which lie within a cluster, goal positions $\mathbf{g}_i$ are determined after estimating the optimal transformation $\mathbf{T}$ that matches initial and deformed positions (denoted by $\mathbf{p}_i^0$ and $\mathbf{p}_i$, respectively) in a least-square fashion:

$$\mathbf{g}_i = \mathbf{T} \begin{pmatrix} \mathbf{p}_i^0 \\ 1 \end{pmatrix} \tag{3.13}$$

[1] https://gitlab.com/altairLab/probe-tissue-simulation

Since clusters can overlap (i.e., particles may belong to multiple clusters) the final goal position for a particle is obtained by averaging goal positions of the corresponding regions. Position corrections are then computed as:

$$\Delta \mathbf{p}_i = \alpha(\mathbf{g}_i - \mathbf{p}_i) \tag{3.14}$$

where $\alpha \in [0, 1]$ is the stiffness used to enforce the constraint. As a consequence, realistic elastic behavior is obtained by appropriately selecting cluster parameters. For example, the higher the number of clusters, the more degrees of freedom the body will have, as shown in Fig. 3.5. Efficient implementations of the PBD approach together with the region-based shape matching constraint are currently available in several software libraries, such as the NVIDIA FleX [149].

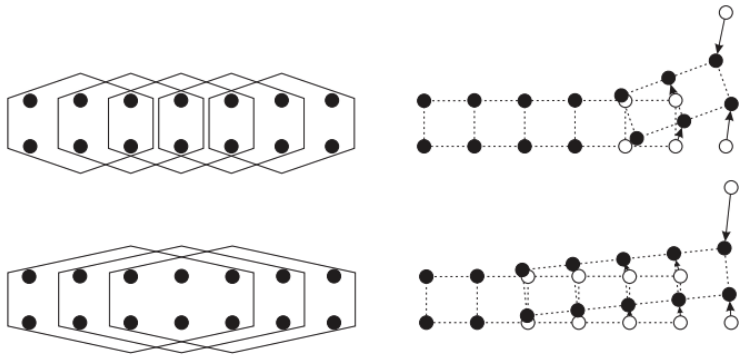


Figure 3.5: In region-based shape matching, the number and the size of clusters influence the final deformed configuration of the object. (Left) The same initial configuration of particles is associated with smaller (top) and bigger (bottom) clusters (represented by hexagonal regions). (Right) Final configurations obtained, in each case, when one particle is moved. Smaller clusters allow to model finer deformations. Image taken from [84].

3.5.1 PBD-based breast model

In order to employ the PBD method for the prediction of breast deformation, it is necessary to identify the parameters able to approximate tissues behavior. To obtain a patient-specific PBD model, we propose to identify optimal PBD parameters in the pre-operative phase with a two-steps strategy (Fig. 3.6). In particular, we first obtain a reasonable starting guess of deformation parameters by performing an initial calibration on a minimalistic phantom whose mechanical and echogenic properties approximate those of the clinical scenario of interest. This first step provides an initial guess of parameters values which is common for all the patients. Since we expect that such parameters will not be able to accurately model each anatomy, in the following phase we perform a subject-specific tuning of their values to account for inter-patient variability. The main advantage of performing an initial calibration on a minimalistic phantom is that it provides an initial acceptable guess of the parameters that allows to start the refinement process from a point closer to the optimum and also to restrict the search space, making the pre-operative optimization more efficient.

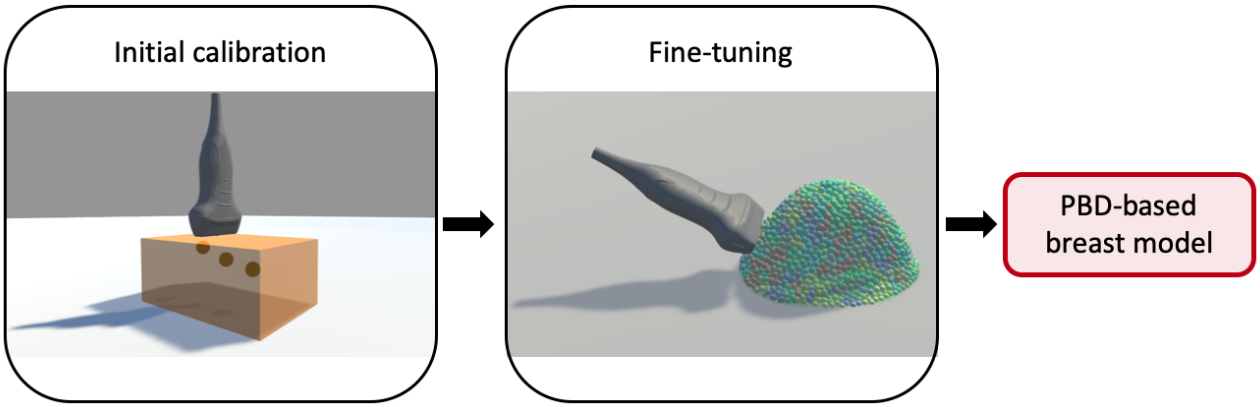


Figure 3.6: The proposed pipeline to identify model parameters. An initial calibration on a box-shaped deformable phantom allows to estimate a reasonable initial guess of parameters. Fine-tuning on the scenario of interest allows to adjust values of the parameters in a patient-specific way, obtaining the final PBD-based breast model.

In general, PBD simulations are controlled by a high number of parameters, but tuning all of them is out of the scope of this research. We focus on the optimization of the parameters defining the clusters of region-based shape matching constraint present in Nvidia FleX implementation, which control objects' deformable behavior: *cluster spacing* (i.e., the distance between adjacent clusters), *cluster radius* (i.e., the radius of each cluster region) and *cluster stiffness* (i.e., the extent to which adjacent cluster are constrained to each other). Although it is well known that other PBD parameters can have an impact on soft body behavior, we decide to keep their values fixed for all the simulations and to set them in accordance with previous works (Table 3.1) [91]. Moreover, following the assumptions stated in Section 3.3.2, deformable objects are treated as homogeneous and, as a consequence, we identify global parameters.

Table 3.1: PBD parameters kept constant for all the simulations

Parameter	Value
Time step	0.02 s
Simulation substeps	3
Substep iterations	9
Relaxation type	Local
Gravity	9.81 m/s^2
Volume sampling	7
Particle spacing	5 mm
Shape friction coefficient	0.35
Particle friction coefficient	0.25
Damping factor	12
Collision distance	3 mm
Self-collision	True

Initial calibration

Simulation parameters of the PBD deformation model are initialized on a handcrafted box-shaped calibration phantom (155x100x70 mm), made of ballistic gel as described in [150]. In addition to correctly approximating the consistency of the clinical scenario of interest, ballistic gel also has realistic echogenic properties. The realism of the setup is further enhanced with the inclusion of three stiffer beads with a diameter of 18 mm within the phantom, placed at three different heights (47 mm, 53 mm, 63 mm). The first box in Fig. 3.6 shows the corresponding simulation environment, where the virtual US probe is modelled as a rigid body which follows in real-time its physical tracked counterpart. Probe-tissue interaction is modelled as a contact problem, handled by the default collision detection and response implementation provided by the Unity engine. As boundary condition for the simulation, we fix all the points which belong to the lowest phantom surface.

The experimental protocol followed for parameters optimization consists of five acquisitions for each phantom inclusion. The initial rest condition is obtained by only slightly touching the phantom with the US probe, without inducing any deformation. Afterwards, four US images are acquired in correspondence of the center of each bead, by applying downward probe displacements of 5 mm, 10 mm, 15 mm and 20 mm.

Estimation of optimal model parameters for the calibration phantom is performed with the genetic algorithm scheme. By generating a population of possible solutions at each iteration in a stochastic way, this methodology eventually evolves towards an optimal solution. This scheme is known for being able to offer good characteristics of exploration and exploitation of the search space [151]. In this work, we rely on the implementation provided in MATLAB (MATLAB R2018b, Mathworks, Natick, MA, USA). We minimize the prediction error, formulated by the following bound-constrained problem:

$$\mathbf{p}^*{}_{gel} = \underset{\mathbf{lb}<\mathbf{p}<\mathbf{ub}}{\arg\min} \quad \sum_{n=1}^{N}\sum_{l=1}^{L} ||\mathbf{X_{PBD}}(\mathbf{p}, l, n) - \mathbf{X_{US}}(l, n)|| \tag{3.15}$$

where $\mathbf{p}$ is the vector of parameters to optimize, $\mathbf{X_{PBD}}(\mathbf{p}, l, n)$ is the position of the tracked PBD particle, $\mathbf{X_{US}}(l, n)$ is the reference bead position, both relative to bead n at deformation level l, and the symbol $||.||$ represents the Euclidean distance. $\mathbf{X_{US}}(l, n)$ are selected on US images as points belonging to bead contours which lie closer to the US probe; this choice is motivated by the better visibility of interfaces on US, and allows to avoid additional inaccuracies induced by image segmentation necessary, for example, for centroid computation. The tracked PBD particle (which gives $\mathbf{X_{PBD}}(\mathbf{p}, l, n)$) is chosen as the one with minimum distance from the point on bead contour closest to the US probe, in the rest configuration. In order to describe the entire model as deformable, we constrain all the particles to fall within at least one cluster by imposing cluster radius to be at least half of cluster spacing. Lower and upper bounds ($\mathbf{lb}$, $\mathbf{ub}$) for cluster spacing parameter are set to $[5; 35]$ mm, where the minimum value is constrained by particle spacing (Table 3.1), while the maximum value corresponds to half of the smallest gel dimension. Consequently, cluster radius is restricted to the range $[2.5; 15]$ mm. Instead, cluster stiffness is left free to vary within the entire acceptable range $[0; 1]$.

Fine-tuning

The PBD model of the clinical scenario of interest shares the same fixed parameters of Table 3.1. In order to obtain a patient-specific simulation, some experiments are conducted to refine the values of cluster spacing, radius and stiffness parameters before applying the model to predict lesions displacement due to US probe interaction. This process, which we refer to as *fine-tuning*, consists of tracking the position of a single US-visible landmark subject to four probe-induced deformations (15, 20, 25, 30 mm) in a similar fashion to what has been done for the calibration phantom (Fig. 3.6). In our case, we select one of the internal lesions of the phantom as a reference for this procedure. It is worth stressing out that the choice of a lesion as landmark for the fine-tuning solely depends on the fact that it is clearly visible on US for the CIRS phantom we use. Despite its unlikeliness, in case no lesions at all can be detected on US images (one lesion is enough for this procedure), the fine-tuning process can be performed by tracking any other internal structure (like ducts or cysts). Likewise the calibration described above, optimal simulation parameters $\mathbf{p}^*_{breast}$ are chosen as those minimizing the prediction error of our model as of Eq. (3.15), where in this case we only consider $n = 1$ tracked landmark. Due to the fact that the calibration procedure on the ballistic gel phantom has given us a more precise idea of the range where optimal parameters lie, we perform the fine-tuning exploiting the direct search strategy implemented in MATLAB, which has proven convergence to local optimum and is more efficient than genetic algorithm, provided that it starts from a good initialization [152]. In particular, the starting parameter vector is initialized with optimal values obtained for the ballistic gel phantom ($\mathbf{p}^*_{gel}$) and the range for lower and upper bounds for each parameters is restricted to 40% of the initial range, centered in the starting point.

Once optimal parameters are found, the PBD model is updated and used to infer the displacement of each of the other 9 segmented lesions under four deformations, after the probe is moved such that the corresponding lesion can be seen on the US image.

Optimization results

This Section reports the results relative to the calibration of simulation parameters on the ballistic gel phantom, and to their fine-tuning on the breast phantom. Table 3.2 reports optimal values for cluster spacing, radius and stiffness parameters estimated through the genetic algorithm strategy (for the calibration phantom) and the direct search method (for the breast phantom). The Table also shows the average error and standard deviation over all deformations on the breast phantom, when each set of parameters is used, when predicting the position only of the lesion selected as landmark for the fine-tuning process. The fine-tuning process has allowed to achieve a reduction of 24% in the overall mean target error.

Even though the fine-tuning process is able to further optimize the parameters, Table 3.2 shows that errors obtained with the initial set can be considered already acceptable. We expect that once the model is applied to the true clinical context, where the high inter-patient variability in geometries and boundary conditions is unlikely to be described by a single phantom, the role of the fine-tuning will emerge more clearly. A simulation scene relying on the PBD approach using the estimated parameters is made publicly available at https://gitlab.com/altairLab/breastsimulationpbd.

Table 3.2: Optimal values of cluster *spacing*, *radius* and *stiffness* parameters estimated with the proposed optimization strategies for the calibration and breast phantoms. Last columns report the mean error and standard deviation over all the deformations in mm, when each set of parameters is used to predict the position of the landmark used for the fine-tuning process.

	Cluster spacing	*Cluster radius*	*Cluster stiffness*	**Mean Error**	**STD**
Calibration phantom	9.6001	9.1674	0.4524	6.64	2.00
Breast phantom	11.1626	8.5424	0.4649	5.07	1.62

3.6 The U-Mesh framework

In this Section, we propose a novel approach where a neural network is trained to predict the deformation of internal breast tissues starting from the surface displacements induced by the US probe. Inspired from [107], the proposed method consists in a U-Net architecture that is trained on a single patient geometry before surgery, and can thus be seen as a patient-specific model. The training dataset is composed of synthetic data generated with FE simulations. In contrast to the work of [110], FE simulations that compose the training set are generated with several random input displacements, making our approach able to generalize to different probe positions and compression extents. By relying on a neural network, this method has the potential to provide an accurate prediction of breast tissue deformations due to US probe pressure in nearly real-time.

Training data generation

The training data set consists of pairs of $(\mathbf{u_s}, \mathbf{u_v})$ where $\mathbf{u_s}$ is the input partial surface displacement (i.e., the displacement of the points belonging to the breast-probe contact area) and $\mathbf{u_v}$ is the volumetric displacement field. Even though the data generation process takes place in an offline phase, in order to generate enough training data with FE simulations within clinically acceptable times, it is important to have simulations that are both accurate and computationally efficient. For this reason, training data are generated by modeling the breast with a St Venant-Kirchhoff model (Eq. (2.9)), which is the simplest and most efficient extension of a linear elastic material to the nonlinear regime. Even though choosing a simple linear elastic model would have been even more efficient, it would not be able to appropriately describe the large deformations undergone by the breast. Elastic properties are set in accordance with the values estimated in [140] for the same breast phantom considered here. Following the indications in Section 3.3.2, we constrain the motion of all the nodes belonging to the lowest phantom surface.

We choose to discretize the domain into 8-node hexahedral (H8) elements not only for their good convergence properties and lock-free behavior, but also because it is the required structure for the input to the network. To do that, the 3D breast geometry is embedded in a regular grid of hexahedral elements (see Fig. 3.7a). An iterative Newton-Raphson method is used to solve the non-linear system of equations approximating the unknown displacement.

In order to train our model to estimate breast volume deformation in response to pressure

Algorithm 2 Training data generation strategy for the U-Mesh framework.

Input: Number of samples N, breast surface Γ, breast surface normals Γ_N, US probe surface size S
Output: U-Mesh input $\mathbf{u_s}$, U-Mesh output $\mathbf{u_v}$
Init current sample $id = 0$
while $id < N$ **do**
    Select a random node $p \in \Gamma$;
    Select $\boldsymbol{n} \in \Gamma_N$ at point p;
    Select an oriented bounding box A of size S, centered in p and normal to Γ;
    Select all the surface points $P \in A$;
    Select $\boldsymbol{d} = \boldsymbol{n} + \alpha\boldsymbol{n}$ with $\alpha \in \left[-\frac{\pi}{4}, \frac{\pi}{4}\right]$;
    Select $\boldsymbol{f} \mid |f| \in [0.0, 0.8]$ N;
    Apply $\boldsymbol{f}$ along direction $\boldsymbol{d}$ to the P selected points;
    Run the simulation;
    Compute $\mathbf{u_s}^{id}$, i.e. the displacement at P;
    Compute $\mathbf{u_v}^{id}$, i.e. the displacement of all volume points;
    $id+ = 1$
return $\mathbf{u_s}$, $\mathbf{u_v}$

imposed with the US probe, we generate a dataset of N samples within the SOFA framework [148]. Each sample in the dataset is obtained from the simulation of a probe-induced deformation, generated by applying a force $\boldsymbol{f}$ to a subset of points on the breast surface falling within a region of the same size of the US probe lower surface, which represents probe-tissue contact area. The applied force $\boldsymbol{f}$ has a different direction and magnitude for each sample. Force direction is chosen to allow some angle deviation from the normal to the surface, thus enabling to include samples where the probe compression is a bit tilted, as can be the case in freehand US acquisitions. The maximal allowed force magnitude (i.e., 0.8 N) is set such that the amount of maximal deformation reproduced in the training dataset never exceeds too much that observed in real clinical settings. Data generation strategy is detailed in Alg. 2.

The U-Net architecture

The objective of our work is to find the relation function f between the partial surface deformation under the US probe and the deformation inside the breast. Let $\mathbf{u_s}$ be the surface deformation and $\mathbf{u_v}$ the volumetric displacement field. In order to find f a minimization is performed on the expected error over a training set $\{(\mathbf{u_s}^n, \mathbf{u_v}^n)\}_{n=1}^N$ of N samples:

$$\min_\theta \frac{1}{N} \sum_{n=1}^{N} \| f(\mathbf{u_s}^n) - \mathbf{u_v}^n \|_2^2 \tag{3.16}$$

where θ is the set of parameters of the network f. We propose to use the same architecture as in [107], that is a U-Net [153] adapted to our application (Fig. 3.7b). The network consists of an encoding path that reduces the high dimensional input into a reduced space, and a decoding path that expands it back to the original shape. The skip connections transfer features along matching levels from the encoding path to the decoding path through crop and copy operations.

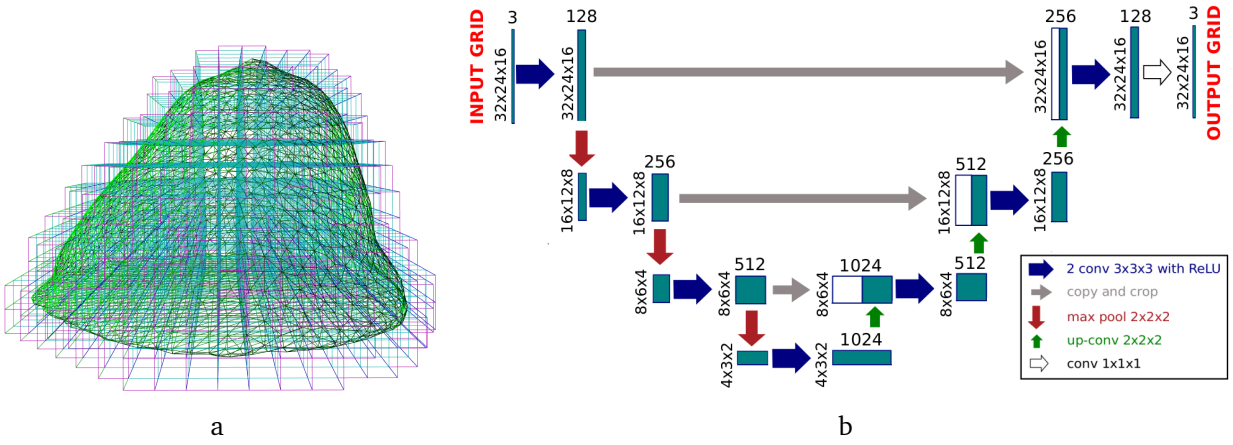


Figure 3.7: (a) Breast surface geometry obtained the pre-operative CT scan immersed in a hexahedral grid for FEM computations. (b) U-Net architecture for a padded input grid of size $32 \times 24 \times 16$.

As Fig. 3.7b shows, the encoding path consists of k sequences ($k = 3$ in our case) of two padded $3 \times 3 \times 3$ convolutions and a $2 \times 2 \times 2$ max pooling operation. At each step, each feature map doubles the number of channels and halves the spatial dimensions. In the lower part of the U-Net there are two extra $3 \times 3 \times 3$ convolutional layers leading to a 1024-dimensional array. In a symmetric manner, the decoding path consists of k sequences of an up-sampling $2 \times 2 \times 2$ transposed convolution followed by two padded $3 \times 3 \times 3$ convolutions. At each step of the decoding path, each feature map halves the number of channels and doubles the spatial dimensions. There is a final $1 \times 1 \times 1$ convolutional layer to transform the last feature map to the desired number of channels of the output (3 channels in our case). The design of the U-Net is based on a grid-like structure due to this up- and down-sampling process. By directly meshing our deformable object with regular hexahedral elements, generated data are already in the format required by the network.

In this work, we discretize the breast phantom into 2,174 hexahedral elements and we simulate several random probe-induced displacements with the pipeline described above (Alg. 2). Overall, we generate $N = 1,000$ samples, divided in 800 for training and 200 for testing. The maximal nodal deformation present in the generated dataset is $79.09\,\text{mm}$ (Fig. 3.8a). We use a Pytorch implementation of the U-Net, which is trained in a GeForce GTX 1080 Ti using a batch size of 4 and 100,000 iterations and the Adam optimizer.

3.6.1 U-Mesh validation on a synthetic dataset

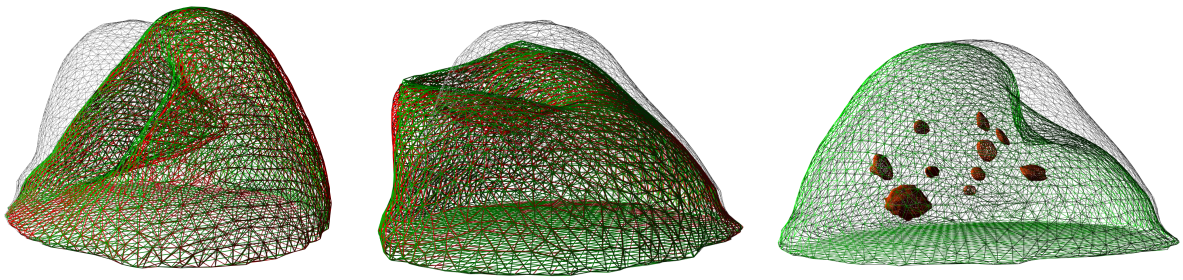
In order to assess if the U-Mesh has learnt to predict tissue deformations, we evaluate its performance with respect to those of the FE model used for data generation. We compute the Mean Norm Error (MNE) over the testing data set. Let $\mathbf{u_v}^i$ be the ground truth displacement for sample i generated with the training data generation pipeline described above and $f(\mathbf{u_s}^i)$ the U-Net prediction. The MNE between $\mathbf{u_v}^i$ and $f(\mathbf{u_s}^i)$ for sample i reads as:

$$MNE^i = \frac{1}{n} \sum_n |\mathbf{u_v}^i - f(\mathbf{u_s}^i)|. \tag{3.17}$$

Table 3.3: MNE as average $\pm$ standard deviation and MNE maximal value over the testing dataset for a breast having $2,174$ H8 elements.

MNE [mm]	**MNE**$_{max}$ [mm]
0.052 $\pm$ 0.050	0.266

where n is the number of nodes of the mesh. Average MNE, standard deviation and maximal value (MNE$_{max}$) over the testing data set are reported in Table 3.3. The maximal error is of only 0.266 mm and corresponds to the sample shown in Fig. 3.8b. The most striking result is the small computation time required to make the predictions: only 3.14 ± 0.56 ms. In contrast, the FE method takes on average 407.7 ± 64 ms to produce the solution. Obviously, the resolution of the FE mesh could be reduced to accelerate the computations but at the cost of an accuracy loss.



a b c

Figure 3.8: (a) Sample with maximal deformation (79.09 mm). (b) Sample with maximal MNE (0.266 mm). The green mesh is the U-Net prediction and the red mesh is the FEM solution. The initial rest shape is shown in grey. (c) U-Net prediction on phantom data.

Obtained results show that the U-Mesh framework has learnt how to predict tissue deformations, achieving comparable accuracy to the FE method used to generate the training dataset while guaranteeing very fast prediction time. Although the FE model used to train our network does not perform in real-time, its prediction delay of less than $1\,s$ might be considered already acceptable for our specific application. However, such good computational performance is achieved since in this preliminary evaluation we use a very simplistic model, that does not account for heterogeneity or complex boundary conditions happening in clinical cases. Usage of a more complex FE model will certainly cause an increase of computation load. On the contrary, an important feature of our approach is that the prediction time remains close to $3\,ms$ regardless of the grid resolution and of the biomechanical model used for the data generation process. This means that increasing the complexity of the model used to generate the data set will not affect the prediction speed.

3.7 Analysis of the FE-based models

Before comparing the performance of the three different methodologies considered (FE, PBD and U-Mesh), we perform a first analysis which concerns only the three FE-based approaches

that can be used to model tool-tissue interaction as presented in Section 3.4, at different discretization levels. Fig. 3.9 reports the distribution of the errors ϵ produced by each method (*Penalty*, *LM*, *PrescrDispl*) on all the tumors and all deformations, at increasing mesh resolutions. Boxplots are obtained including only errors relative to "valid" deformations, i.e. those which have been successfully simulated without any instabilities by all the three methods. In this way, we prevent the occurrence of any possible bias in the distributions which could arise if a method has been more stable than the others (and as such, it would have had more error values). From a more detailed analysis of the results it has emerged that largest errors are obtained at high input deformations and in correspondence of deeper lesions.

Boxplots in Fig. 3.10 show how the computation time t_{comp} required by each method to simulate a time step h changes at different discretization levels. Results obtained for the stability metric are reported in Table 3.4.

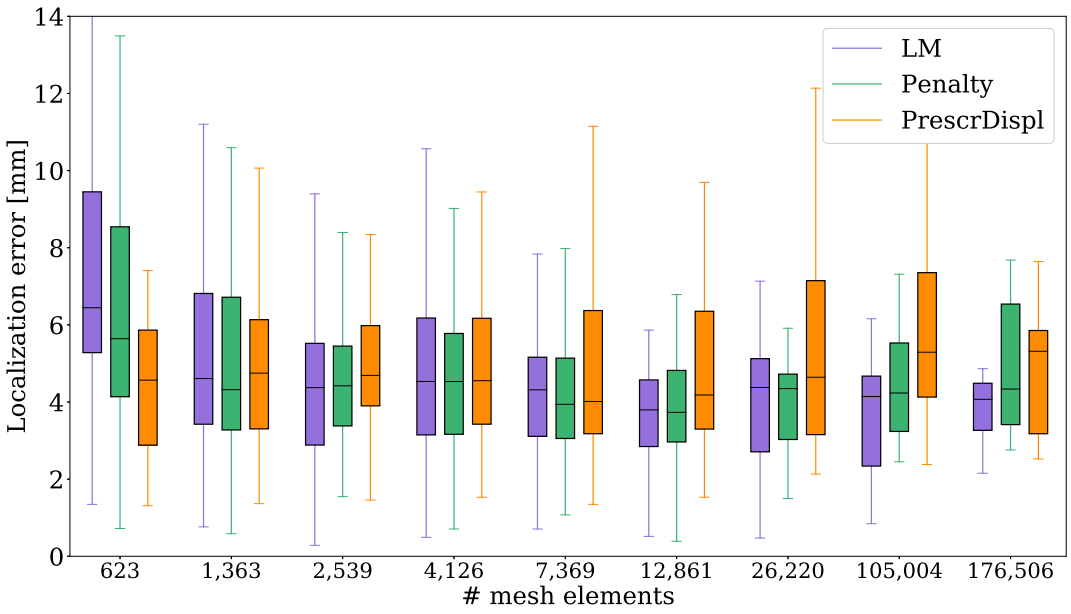


Figure 3.9: Localization errors [mm] on all the tumors and all deformations, for the different methods, at increasing volume mesh resolution.

3.7.1 Discussion

From an accuracy point of view, all the methods achieve similar performances. It is interesting to notice that increasing volume mesh resolution leads to a slight reduction in the error dispersion but not to a significant improvement in the overall accuracy, which is comparable with average lesion dimensions and thus acceptable, for example, for biopsy targeting purposes. This suggests that probe-tissue interactions can be accurately reproduced even with coarse meshes, because of the smoothness of the induced deformations. The reached accuracy level is thus not limited by a poor spatial discretization, but might have a upper bound due to registration and calibration errors and possibly the chosen temporal discretization.

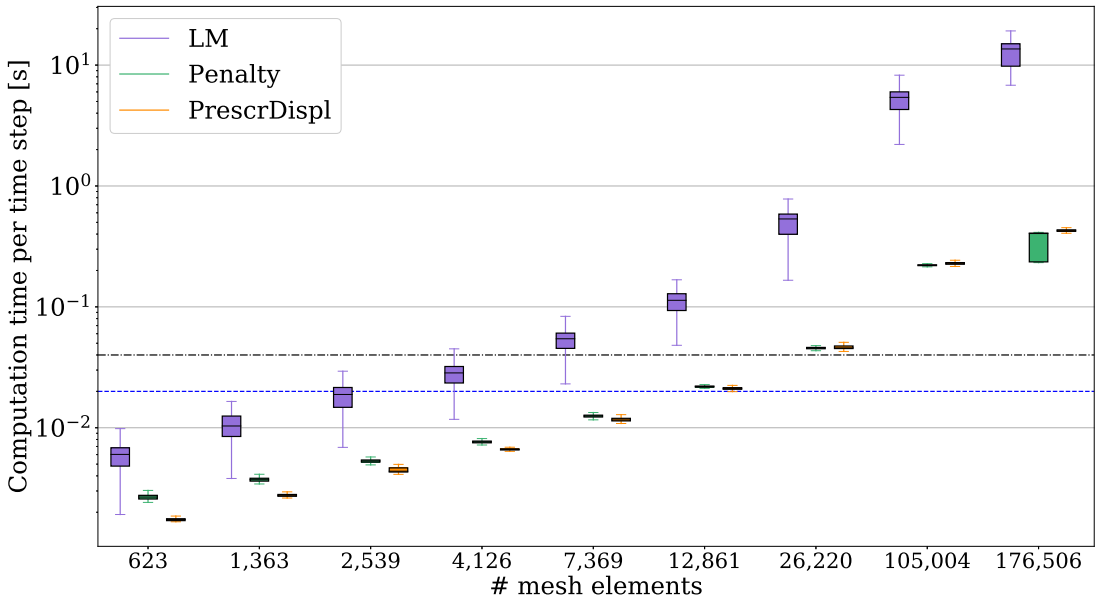


Figure 3.10: Computation time (in log-scale) required by the different methods to simulate a time step ($h = 0.02$ s) at increasing volume mesh resolution. Horizontal lines are in correspondence of the time step (blue) and of the constraint of 25 fps (black).

If we analyse the computational performances, the fastest FE methods are those which rely on the simplest equation systems, i.e. *PrescrDispl* and *Penalty*. Modelling contacts through constraints is the most time consuming approach, despite the use of an optimized solver. Fig. 3.10 shows that using fine meshes has a strong impact on the computation time, which increases for all the methods with the number of elements. A drop of computational performances at high mesh resolutions is particularly important for *LM* method, even though the number of active constraints remains constant in all simulations. It is interesting to notice that, despite relying solely on CPU, the FE approaches tested in this work can meet the real-time constraint for several different discretizations (especially *PrescrDispl* and *Penalty*). We expect that enhanced FE implementations taking advantage of the parallel capabilities of GPU would be able to further improve such computational performances, and we plan to assess this in future works. Although some GPU-based FE approaches have been already proposed, they have not been included in this study since they are either incompatible with hyperelastic simulations [60] or not available within the SOFA framework [130, 44]. The choice of relying on a common open-source simulation platform has allowed us not only to compare the different approaches but also to publicly share the simulation scenes. Furthermore, by providing general implementations of the various methods, SOFA allows to simulate any medical scenarios involving deformable structures, making it possible to exploit the tested FE approaches to model any kind of tool-tissue interaction.

PrescrDispl is the FE approach reaching the best performances in terms of stability. In general, simulations are more likely to become unstable at high deformations, when the effect of non-linearities becomes significant. An interesting result which emerges from Table

Table 3.4: Mean, median and range of the stability metric for each method, at increasing volume mesh resolution. Stability metric is computed as percentage of experiment successfully accomplished.

Mesh elements	LM			Penalty			PrescrDispl		
	Mean	Median	Range	Mean	Median	Range	Mean	Median	Range
623	100	100	93-100	100	100	100-100	100	100	100-100
1,363	100	100	88-100	100	100	90-100	100	100	100-100
2,539	97	100	64-100	100	100	100-100	100	100	100-100
4,126	99	100	84-100	100	100	100-100	100	100	100-100
7,369	99	100	75-100	100	100	92-100	100	100	100-100
12,861	98	100	72-100	93	100	59-100	100	100	100-100
26,220	96	100	60-100	72	71	42-100	100	100	100-100
105,004	88	92	56-100	87	100	52-100	100	100	94-100
176,506	69	70	49-88	53	48	25-100	98	100	80-100

3.4 is that using high resolution meshes leads to higher simulation instability, which may be due to the introduction of further numerical errors preventing simulations from being completed.

The first three rows of Table 3.7 summarize the main advantages and disadvantages of the considered methods. The best trade-off among the performance criteria is achieved when the interaction is modelled by prescribing the displacement of surface nodes. The simpler mathematical description of the physical problem, without the involvement of any external forces, allows to achieve high speed while guaranteeing high simulation stability. The limitation of this method lies in the assumption that the breast-probe contact area is a-priori known and does not vary during the procedure. This represents a major constraint when the simulation is required to run online during the scanning itself, such as in freehand acquisitions. However, this constraint is not a strong limitation in a robotic scenario, where the robot motion is commonly planned in advance, thus enabling the a-priori estimation of the contact surface. The most general ways to describe probe-tissue interaction with FE involve collision handling. The penalty method can reach close to real-time performances, but its stability highly depends on the contact stiffness value, which has to be tuned for each specific problem. This modeling strategy is likely to represent the most appropriate choice in scenarios which are less complex than the one described in this work (for example, with smaller input deformations and/or elastic materials)[124]. Describing the problem as a constraint (*LM*) is the most general and widely applicable approach. In addition to their independence on a specific parametrization, LM-based methods are able to provide a direct measure of the interaction forces through Lagrange multipliers values. The main limitation of this approach is the computation time, which becomes prohibitive if fine spatial discretization is needed. The high computation burden of *LM* may be due to the fact that the system matrix $\mathbf{A}$ is factorized at each simulation step by the direct solver. Being matrix factorization one of the most demanding steps, we performed some tests using an alternative approach where $\mathbf{A}^{-1}$ is updated less frequently (every 5 time steps, as in [60]). However, only a slight improvement in computational performances was achieved, at the expenses of stability. It means that the assumption that the system matrix $\mathbf{A}$ does not change significantly between consecutive time steps does not always hold for hyperelastic objects, causing divergence in the simulation if an

approximation of the real matrix is used.

3.8 Comparison of the different methods

In this Section, we compare the three different modeling strategies presented in Sections 3.4, 3.5 and 3.6 on the US scanning experiments described in Section 3.3.1. For the sake of compactness, results relative to the FE approach are obtained using *PrescrDispl*, which has been selected as it is the one achieving the best trade-off between the three considered metrics of accuracy, speed and stability (see Section 3.7). In particular, we report the results obtained with *PrescrDispl* using a mesh with $4,126$ elements.

Table 3.6 reports the localization errors ϵ for each phantom lesion m with respect to the applied deformation l for the three methods. Fig. 3.11 shows localization errors at increasing input displacement, averaged on all the tumors. Error values are compared with those obtained with a rigid model, i.e. when the lesion is moved of the same amount as probe displacement.

Results in terms of computation time and stability are detailed in Table 3.5.

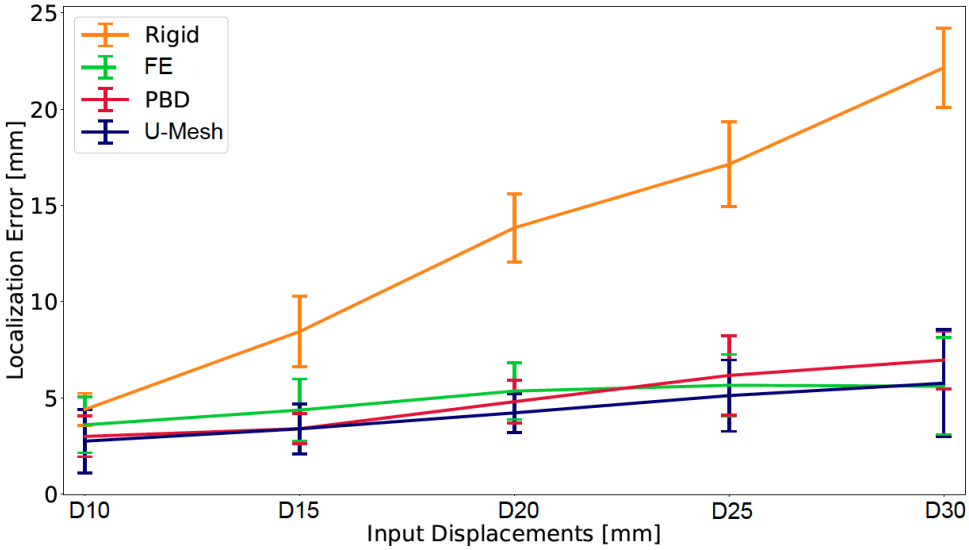


Figure 3.11: Average localization error [mm] at increasing deformation extent for the different methods. Vertical bars represent the standard deviation.

Table 3.5: The Table reports, for each method: (1) the time (median±std) required to perform a simulation time step and (2) stability, expressed as the percentage of simulation successfully accomplished from the beginning to the end (mean (min-max)).

	FE (*PrescrDispl*)	**PBD**	**U-Mesh**
Time [s]	0.007 ± 0.001	0.02 ± 0.00	0.003 ± 0.0006
Stability [%]	$100(100-100)$	$100(100-100)$	$100(100-100)$

Table 3.6: Localization errors [mm] for different tumors and different deformation ranges in the breast phantom. The first Table is for the FE method, the second Table for PBD method, while the third Table reports results obtained with the U-Mesh. Not-acquired data is reported as (-).

FE (*PrescrDispl*)

TumorID	D10	D15	D20	D25	D30	Mean	STD
1	–	2.686	2.495	3.626	5.175	3.495	1.060
2	3.621	5.964	5.804	–	7.087	5.619	1.255
3	3.793	–	6.952	8.136	9.646	7.132	2.151
4	4.142	3.886	4.954	4.822	–	4.451	0.449
5	1.255	1.275	1.590	–	3.758	1.970	1.041
6	6.581	5.871	6.410	–	6.671	6.383	0.310
7	5.462	5.614	6.414	10.768	–	7.065	2.169
8	6.983	4.835	4.874	5.418	–	5.527	0.871
9	–	5.032	5.277	5.047	6.503	5.465	0.607
10	2.682	2.831	3.341	2.966	–	2.955	0.245
Mean	4.315	4.222	4.811	5.826	6.473		
STD	1.816	1.557	1.698	2.523	1.805		

PBD

TumorID	D10	D15	D20	D25	D30	Mean	STD
1	–	3.259	3.724	5.307	6.214	4.626	1.190
2	1.467	4.475	6.700	–	8.347	5.247	2.579
3	3.689	–	5.590	7.831	11.930	7.260	3.069
4	5.141	6.011	5.396	5.684	–	5.558	0.325
5	2.190	2.018	4.501	–	6.694	3.851	1.912
6	5.644	4.319	3.735	–	3.982	4.420	0.737
7	2.810	3.961	6.374	10.636	–	5.945	2.998
8	5.581	5.659	6.120	6.683	–	6.011	0.440
9	–	4.506	3.833	4.007	4.511	4.214	0.301
10	4.606	2.990	3.193	3.774	–	3.641	0.627
Mean	3.891	4.133	4.917	6.274	6.946		
STD	1.499	1.189	1.207	2.213	2.649		

U-Mesh

TumorID	D10	D15	D20	D25	D30	Mean	STD
1	-	1.936	2.002	1.506	3.053	2.124	0.569
2	3.211	2.905	4.068	-	4.137	3.580	0.534
3	2.032	-	4.709	7.134	10.90	6.194	3.262
4	0.505	2.225	5.313	5.903	-	3.486	2.217
5	0.932	2.768	3.454	-	4.893	3.012	1.425
6	3.923	6.349	5.625	-	6.724	5.655	1.075
7	3.454	3.864	4.543	6.710	-	4.643	1.255
8	2.422	3.261	4.320	5.136	-	3.785	1.030
9	-	3.928	4.214	4.578	4.858	4.394	0.353
10	5.529	3.272	3.940	4.846	-	4.397	0.860
Mean	2.751	3.390	4.219	5.116	5.761		
STD	1.638	1.294	1.007	1.854	2.788		

3.8.1 Discussion

Our results show that all the considered methods reach comparable accuracy levels at all input displacements (Table 3.6). For all the methods, localization error gets slightly worse at larger deformations, but always remains below 7 mm on average (Fig. 3.11). This result significantly outperforms the error made when employing a rigid model, thus emphasizing the need of a deformation model in this context.

Although PBD method is the one making the largest error at high input displacements, the overall accuracy is aligned with that of the other methods (Table 3.6), proving that the proposed parameter calibration strategy has allowed to obtain a reliable patient-specific model. Performances of the PBD method in terms of computational time and stability are particularly promising. The highly optimized PBD implementation provided by NVIDIA FleX is able to keep the simulated and computation times always equal ($t_{comp} = h$), thus guaranteeing that the real-time constraint is met. Moreover, PBD intrinsic unconditional stability guarantees PBD-based simulations to always remain stable (mean and median stability of 100%, in the range 100-100). Although improved performance in terms of time efficiency and stability are also reached by the FE method tabulated in Table 3.5, it is worth highlighting again that reported results are relative to a specific FE implementation (*PrescrDispl*), which has achieved best performance on our specific scenario. Such results are not valid in a general sense, but are highly influenced by the chosen model and spatial discretization, as detailed in Section 3.7. Overall, within the considered context, a non-physically based method (PBD) has proved able to reach an optimal compromise among the different performance criteria: it achieves an accuracy which is comparable to FE simulations, meeting the real-time constraint and maintaining enhanced stability, which make it particularly suitable for employment within a robotic simulation framework. An additional advantage of the PBD approach is that it allows to avoid the time-consuming generation of high quality mesh, which represents the major bottleneck in FE simulations (especially in those inolving large deformations)[20]. Since we are targeting a patient-specific context, this represents an enormous advantage because the mesh would have to be constructed everytime, for each patient. Furthermore, thanks to its direct manipulation of positions, the PBD approach can easily handle collisions constraints. Probe-tissue interaction can thus be effectively treated as a collision problem, thus allowing to deal with any input probe position without requiring the explicit definition of the contacting surface. The main drawback of region-based shape matching constraint governing model behavior is that its parameters need to be calibrated for each scenario, since they do not have a direct physical meaning. Even though some advanced constraints that overcome this limitation have been proposed, they are not yet incorporated in Nvidia FleX framework and not even publicly available. On the other hand, all FE-based methods benefit from the possibility of using the real elastic parameters. Another limitation is that PBD cannot provide an estimation of the interaction forces, which may be useful in several applications, especially within the robotic context.

The accuracy levels achieved with the U-Mesh framework are also aligned with those obtained by the other approaches, and follow a similar trend at increasing input displacements. This proves that U-Mesh has been able to learn to predict probe-tissues deformation from simulated data and generalize to real world probe-induced deformations. The U-Mesh guarantees the very fast prediction time of 3 ms and is always stable, since it does not involve any solution process which might suffer from numerical issues, differently from all the other

simulation-based considered methods. Therefore, the U-Mesh represents an interface between precise patient-specific biomechanical FE modeling (not capable of real time) and applications requiring both high accuracy and very high speed. Even though some of the FE-based approaches considered in this Chapter could already achieve good performance in terms of computation time and stability (Section 3.7), they rely on a very simple breast model with homogeneous properties. This modeling choice does not represent a limitation in the considered setup with a breast phantom, but a more complex model handling multiple materials and anisotropy would probably be needed to reach reasonable accuracy on real patient data. Usage of a more complex FE model will certainly cause an increase in computational load, making it almost impossible to achieve real-time performances with FE simulations. The U-Mesh framework represents a promising solution in this case, since high prediction speed is guaranteed regardless of the biomechanical model used for the data generation process. The main limitation of the method remains the training process, which is burdersome and has to be repeated for every new geometry or application. However, we have shown that a limited amount of training data can be sufficient to train a U-Net such that it obtains accurate prediction within clinically acceptable times. In its current implementation, the U-Mesh is comparable to *PrescrDispl* approach, since it relies on the displacement of the breast points below the US probe as input. As a consequence, the method cannot directly generalize to scenarios where the probe-tissue contact area is not a-priori known. However, as stated before, this fact does not always represent a limitation, for example in robotic scenarios where the trajectory is planned beforehand. In addition, this approach does not provide any information about interaction forces. However, the network might be designed to provide such information as an additional output. It is worth noticing that it is not straightforwart to integrate strategies for online parameter update within the U-Mesh framework, due to its reliance on a neural network, whose parameters are network weights and updating their values would require re-training. This aspect, which is important for the development of realiable autonomous systems (Section 1.2), might be more easily incorporated within simulation-based approaches.

Table 3.7 summarizes the main advantages and disadvantages of the different methods analyzed.

3.9 Conclusion

In this Chapter, we have compared three different categories of methods that can be employed to model tissue deformations, evaluating their performance in terms of accuracy, computation time and stability. In particular, we have considered the modeling of the interaction between soft tissues and a rigid tool, in the context of breast US scanning.

Models based on the finite element method, the preferred modeling strategy for accurate medical simulations, have proved capable of achieving a good trade-off between accuracy, computation time and stability, at least when relying on a simplified model with quite coarse meshes. However, obtained performances are highly variable depending on the chosen approach (i.e., constraints-based, prescribing displacements) and the required level of spatial discretization, whose choice depends on the specific clinical requirements. Being physics based, FE-based models can directly exploit real tissues mechanical parameters, obtained from either prior knowledge or ad-hoc imaging techniques (e.g., elastography).

Table 3.7: Summary of the main advantages and disadvantages of the different modeling methods.

	Pros	Cons
Penalty	✓ Generic contact problem ✓ Low computation time	✗ Requires stiffness tuning ✗ Highly unstable
LM	✓ Generic contact problem ✓ Interaction forces retrievable	✗ High computation time
PrescrDispl	✓ Low computation time ✓ High stability ✓ Independent on real mechanical properties	✗ Requires knowledge of contact area
PBD	✓ Generic contact problem ✓ Low computation time ✓ High stability	✗ Requires patient-specific optimization
U-Mesh	✓ Comparable accuracy to FE method ✓ Low computation time ✓ High stability	✗ Dependence on training dataset ✗ Online parameters update not possible

A position based dynamics model has also shown the capability to reach comparable accuracy with FE simulations, after patient-specific parameter identification, while meeting the real-time constraint and maintaining enhanced stability. More advanced constraints can be used to enforce specific elastic behaviors relying on real mechanical parameters [154], and will be investigated in future works. In fact, the parameter optimization process represents, to date, the main bottleneck of this approach, preventing its wide exploitation in the medical field.

In this Chapter, we have also proposed to use deep learning to learn a deformation model directly from data, generated from FE simulations in our case. The presented framework, called U-Mesh, has proved able to provide extremely fast predictions, achieving comparable accuracy to other existing methods and without suffering from any stability issues. This approach is particularly promising to learn patient-specific models of increasing complexity, experiencing a degradation neither in computational performances or in stability. In future works, we will test the capability of the U-Mesh to learn more complex models. Furthermore, future research will focus on making the method able to learn tissue behavior independently from the geometry, similarly to what has been proposed in [106, 14].

A limitation of the conducted analysis is that we have considered a simplified setup with a breast phantom. In future works, we plan to design a protocol to collect real human data, thus extend our evaluation to more realistic cases. We expect that it will be more challenging to reach clinically acceptable accuracy with the considered approaches, probably requiring more complex models and, in turn, it will be more difficult to achieve a satisfactory compromise between performance criteria. Furthermore, our acquisitions are currently limited to compression experiments, being this kind of input the most relevant case, causing the greatest anatomical deformations. However, we plan to extend the experimental protocol in order to include also the sliding interactions. Moreover, we would like to improve the robustness of the experimental protocol. First of all, a robotic manipulator will be used to hold the US probe, to guarantee the application of input deformations in a more controllable and

repeatable way. To further reduce inaccuracies, we plan to rely on a more precise strategy for the selection of corresponding fiducials in real and simulated environments, for example employing an automatic routine for fiducials identification on US images, to avoid human errors involved in the landmarks placement.

To conclude, all the presented strategies allow to successfully model soft-rigid interaction, each achieving different compromises among accuracy, speed and stability. The choice of the preferred approach highly depends on the requirements of the specific clinical application.

Contributions of this Chapter

The main contributions of this Chapter can be summarized as follows:

1. We assess the performances of the three main existing strategies to model soft-rigid interaction with the finite element method in terms of accuracy, computation time and stability, and we highlight the pros and cons of each approach for robotic applications.
2. We propose to model the interaction between a rigid tool and deformable anatomical tissues using a non physics-based method, i.e. position based dynamics; we show that, once personalized parameters are available, such approach can be successfully exploited to model anatomical tissues.
3. We present U-Mesh, a neural network that is trained only with synthetic data to predict tissues deformation in a patient-specific manner; this method achieves comparable accuracy with finite element based methods while being a hundredth times faster.
4. We make our models available to the community by sharing our implementations at https://gitlab.com/altairLab/probe-tissue-simulation and https://gitlab.com/altairLab/breastsimulationpbd.

Publications linked to this Chapter

The content of this Chapter has been presented in the following publications:

1. Tagliabue E, Dall'Alba D, Magnabosco E, Tenga C, Peterlik I, Fiorini P: *Position-based modeling of lesion displacement in Ultrasound-guided breast biopsy*, International Journal of Computer-Assisted Radiology and Surgery IJCARS (2019);

2. Mendizabal A, Tagliabue E, Brunet J, Dall'Alba D, Fiorini P, Cotin S: *Physics-based Deep Neural Network for Real-Time Lesion Tracking in Ultrasound-guided Breast Biopsy*, Computational biomechanics for medicine workshop at MICCAI (2019); 2019 Oct 13; Shenzhen, (China);

3. Tagliabue E, Dall'Alba D, Magnabosco E, Peterlik I, Fiorini P: *Biomechanical modeling of probe to tissue interaction during ultrasound scanning*, International Journal of Computer-Assisted Radiology and Surgery IJCARS (2020).

Chapter 4

Learning intra-operative model update

4.1 Introduction

Ideally, an ARSS is provided with the sequence of actions to perform which is decided before the intervention based on patient's pre-operative data and a-priori clinical knowledge. Such information is exploited to create a Patient-specific Biomechanical Model (PBM), which is used to design a personalized surgical plan in simulation. However, the information available pre-operatively is often not sufficient to thoroughly characterize the anatomical environment, whose properties vary a lot among patients and are very difficult to measure and model. As a consequence, the generated surgical plan might be sub-optimal and requires correction while surgery is already taking place, depending on the current situation [9].

Whenever actions replanning is needed, the updated motion should be tested in simulation before being executed, in order to guarantee system safety and reliability. As a consequence, the simulation must continuously and closely follow the real surgical environment throughout the operation [9, 155, 156]. Being the simulation based on the pre-operative PBM, it is necessary to update the PBM parametrization during the intervention leveraging on available intra-operative data, to both compensate for inaccurate parametrization and allow the simulated behavior to closely reflect the changes introduced by surgical manipulations.

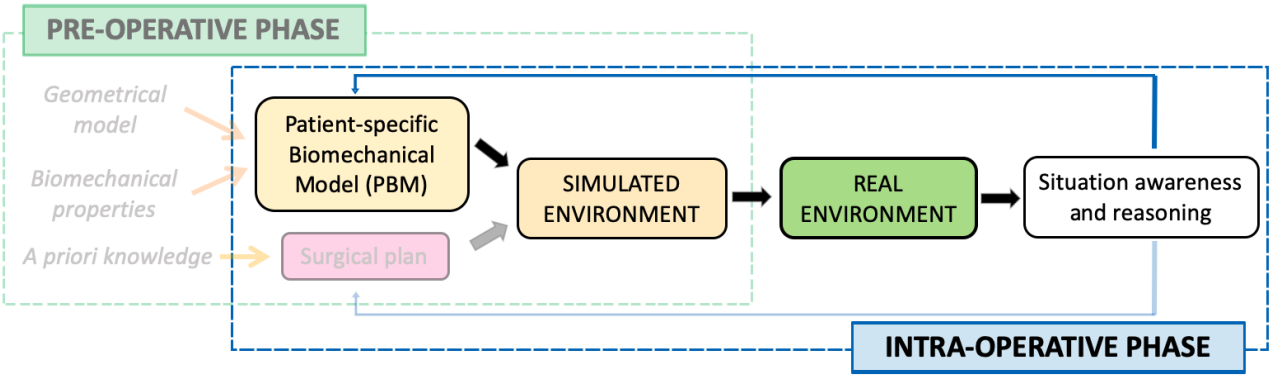


Figure 4.1: ARS pipeline we focus on in this chapter.

In particular, while patient-specific geometry and tissues' elastic properties can be extracted from pre-operative anatomical images or using ad-hoc modalities such as elastographic tech-

niques [157], there is usually no way to delineate the *Attachment Points* (APs) between neighboring organs from pre-operative data. Such APs define Dirichlet boundary conditions (i.e., they act as constraints to the motion of specific points), thus they are key to obtain an accurate simulation [147, 158, 159]. As a consequence, reliable APs can be estimated only from data that are collected intra-operatively [158, 159, 15].

The problem of intra-operative estimation of APs has been tackled by only few works. In [160], attachments are estimated by matching two anatomical configurations extracted from CT scans acquired before and after patient repositioning. This approach assumes that the entire surface is available in both configurations, preventing from its application in real clinical settings, when only a partial surface view can be obtained intra-operatively. Other works [50, 159] propose to initialize the position of attachment points based on statistical atlases. However, statistical atlases are not always able to adapt to patient-specific conditions due to the high inter-patient variability. Another line of research exploits Kalman Filters (KFs) to estimate boundary conditions in the context of liver surgery [158, 159]. The main constraints between the liver and the surrounding tissues is represented by the hepatic ligaments, whose location can be extracted from pre-operative data. As a consequence, these works have focused on the characterization of the ligaments elastic properties, based on the assumption that their location is known, which does not hold in our application. Methods based on KFs have the advantage to work with partial observations (i.e., partial visible surface) coming from intra-operative sensors. However, the filters' inference time strongly depends on their parameters initialization, which is highly sensitive to each patient's properties, possibly introducing a degradation in the performances from case to case.

In this Chapter, we present a strategy to update PBM parametrization intra-operatively by estimating APs. Further to improve simulation accuracy, precise knowledge of the location of APs can support the autonomous execution of some standard surgical actions, like tissue retraction and dissection. Tissue retraction consists in grasping and retracting soft tissues to expose a hidden region of interest such that it becomes visible and accessible to the operating surgeon. Successful retraction requires to avoid grasping close to APs, in order to prevent the application of excessive forces on highly constrained areas, thus reducing the risk of tissue tearing and/or rupture. It might happen that retraction alone is not sufficient to expose the region of interest, in particular when it lies between two tissue layers that are attached to each other. In this case, access to the desired region is obtained by tissue dissection, which consists in the separation of the two attached layers with the surgical instrument (e.g. monopolar scissors) in correspondence of the APs.

In the remaining of this Thesis, we address autonomous execution of the first steps of robotic partial nephrectomy, which consists in the manipulation of pararenal adipose tissue in order to expose the tumor to excise, thus requiring both retraction and dissection. To accomplish the task, surgeons identify the attachment points between the adipose tissue and the kidney via manipulation of the tissue itself. In the same manner, an autonomous agent has to identify attached regions during task execution, since such attachments do not have a standard location and cannot be identified from pre-operative data. However, intra-operative identification of attachment regions involves some challenges related to the fact that such areas are often hidden from the partial view of the scene provided by intra-operative sensors.

4.2 Intra-operative estimation of tissue attachment points

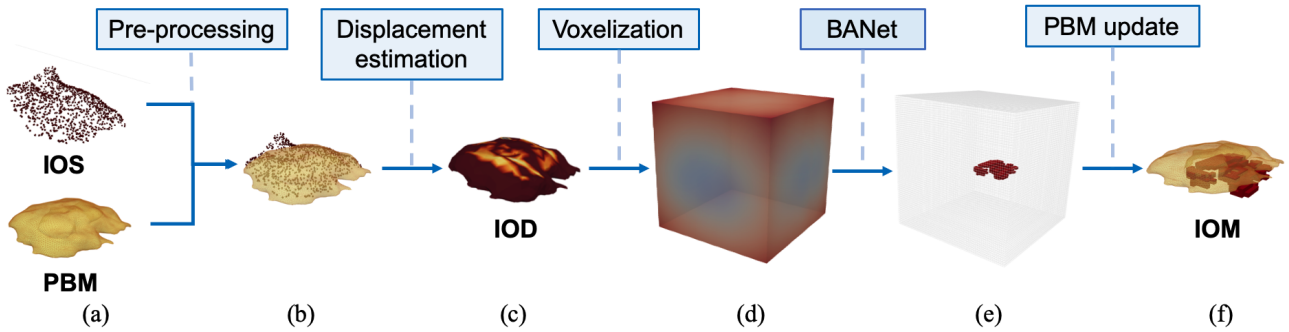


Figure 4.2: Overview of the pipeline to update PBM boundary conditions. (a) Initial data: IOS and PBM; (b) rigidly aligned data; (c) estimated IOD, where brighter color is associated with highest displacement; (d) voxelized representation, where grid cells are colored based on the signed distance field from PBM surface; (e) estimated APs in grid space; (f) PBM annotated with estimated APs, giving the intra-operative model IOM.

In order to provide an up-to-date PBM that continuously follows the current surgical scenario, we rely on a framework involving two independent processes which run concurrently. The first process is entirely dedicated to a physics-based simulation of the surgical environment. Although such simulation can be based on any approach among those presented in Section 2, in this Chapter we assume to rely on continuum mechanics methods capable of real-time performances, such as [161]. Such simulation leverages on the PBM created from pre-operative data, characterized by both the undeformed 3D geometry of the anatomy and its known mechanical properties. The second process is devoted to the strategy for updating PBM parametrization during the intervention, starting from intra-operative sensor data. In this Chapter, we focus on this second task. In particular, we present a pipeline to update APs from the 3D point cloud of the surgical scene acquired with an intra-operative camera with a very short latency. This allows the simulation to continuously reflect the changes introduced in the environment by surgical manipulations (Fig. 4.2).

Pre-processing The raw 3D point cloud acquired with the intra-operative camera passes through an initial pre-processing step. First, both color and spatial segmentation are performed to extract the current view of the deformed Intra-Operative Surface (IOS) from the full anatomical point cloud. The extracted IOS is then rigidly aligned to its corresponding portion of the PBM, based on geometric features and known spatial relations estimated at the beginning of surgery.

Displacement estimation We calculate the Intra-Operative Displacement field (IOD) which maps each point in the IOS to its corresponding one on the undeformed PBM. To achieve this, we estimate a correspondence with a nearest-neighbor pairing between the point cloud and the PBM in the functional space [162], and we refine it using the state-of-the-art ZoomOut method [163]. This non-rigid approach is entirely intrinsic, promotes isometric solutions (i.e., correspondences that preserve the surface distances between the points), and has approximately linear complexity. Thus, ZoomOut guarantees a trade-off between accuracy and

timing, making it ideal for real-time precision surgical operations, where we assume the folding and bending of the surface preserves surface geometry (i.e., boundaries and metric) without introducing dramatic stretching.

Voxelization In order to exploit convolutional filters, input information is converted into a grid-like volume of dimension $64 \times 64 \times 64$ and side length $300\,\text{mm}$, following the same voxelization process described in [106]. In particular, the PBM is encoded into the grid using its signed distance field and the IOD through a Gaussian interpolation. By representing the data in this way, our method learns to interpret the geometry, allowing it to directly generalize to new geometries.

Binary Attachment Network The APs update is performed by BA-Net (Binary Attachment Network), a DNN estimating at which points a given deformable tissue is attached to the surrounding environment. Detailed description of BA-Net is provided in the following Section (4.2.1).

PBM update Finally, estimated APs are mapped from grid space to the original PBM space, giving the Intra-Operative Model (IOM). This step completes the proposed pipeline, and the obtained IOM is used to update the intra-operative simulation running in the synchronous process.

4.2.1 BA-Net: Binary Attachment Network

BA-Net is a CNN which outputs a binary map of estimated attachment points starting from a pre-operative PBM and the IOD, encoded in a voxel grid. Despite relying on the same formalism proposed in [106, 14] for its capability to generalize to different geometric shapes, our method has a completely different goal. In fact, BA-Net does not aim at predicting a 3D displacement field that registers two anatomical configurations, but it predicts which points of the pre-operative 3D model act as attachment points. To cope with the lack of real world data where the attachment points are annotated on the corresponding pre-operative volumes, the proposed network is trained only with simulated samples.

Training data generation

The developed strategy for training data generation aims at providing samples that mimic pre-operative and intra-operative configurations of adipose tissues, those preventing direct access to the renal tumor during partial nephrectomy, as realistically as possible. In order to obtain many different random adipose tissue PBMs, we generate a set of surface meshes by applying a series of morphological operations to an icosphere of random dimension. As a following step, the generated samples are clipped within two parallel planes to keep sample thickness below $20\,\text{mm}$, and twist and bend filters are applied to pre-deform the meshes. For this process, we make sure that the average edge length of the generated geometries is comparable to the grid voxel size (i.e., approximately $4.7\,\text{mm}$), to match the mesh and grid resolutions. After tetrahedrization of the resulting mesh, we extract a subset of the surface whose points will act as attachments. To define such subset with a realistic (thus irregular) profile, we associate to each point Q of the surface the value of the metric D_Q which acts as

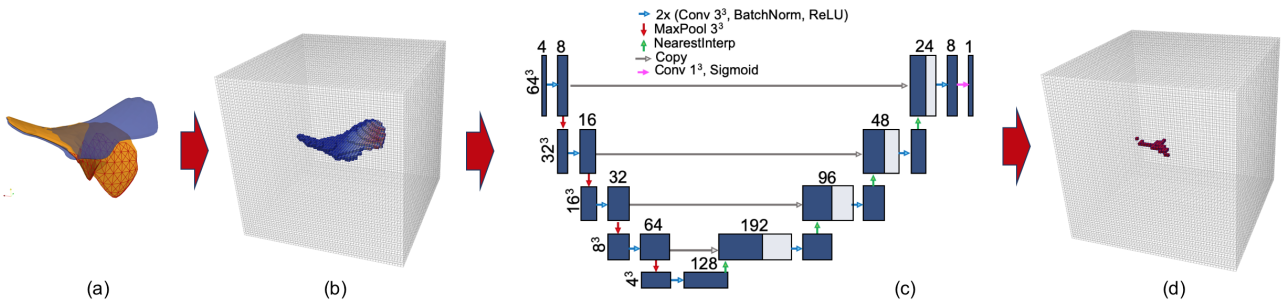


Figure 4.3: Overview of our method. (a) Training data are generated from finite element simulations. The blue mesh represents one of the randomly generated surfaces, while the orange mesh represents the corresponding deformed configuration. A red wireframe overlay is added in correspondence of the IOS, i.e., the visible portion of the deformed mesh. (b) Input to the network is a structured grid where the PBM is encoded through its signed distance field (sdf). Only grid cells belonging to the internal parts of the surface (sdf<voxel size) are displayed. Cell color is proportional to the magnitude of the associated displacement (higher displacement in red, zero displacement in blue). (c) The UNet architecture used. (d) Output of the network is a binary map of the attachment points. Such points will be then converted back from grid coordinates to PBM coordinates.

the likelihood of that point being removed from the subset. After selecting a random mesh node P as center point, we define the distance metric D associated to each point Q as:

$$D_Q = w_d \cdot d_{PQ} + w_n \cdot n_{PQ} + w_p \cdot p_Q \tag{4.1}$$

where d_{PQ} is the geodesic distance between P and Q, n_{PQ} is the angular distance between the normal of P and the normal of Q, and p is the value of perlin noise evaluated at position Q. These three contributions are weighted by w_d, w_n and w_p, whose values change for each new sample within specific bounds that can be adjusted to specify the relative importance of each term. For the extraction of attachment points, we sample w_n within a range of higher values with respect to the other two, to favor the extraction of regions belonging to the same side of the surface. Eventually, the extracted surface includes those points with the lowest values of D, until the desired percentage of surface points (between $(5, 50)\%$) is extracted.

We introduce a simulation environment to obtain tissues deformed state relying on the finite element method provided by SOFA framework [148]. Grasping action performed with a single dVRK arm is simulated by applying a force of random magnitude to a subset of surface nodes within a radius of $(4, 10)$ mm, to simulate different amounts of grasped tissue. In order to make the network independent from specific mechanical properties, we consider varying Young's modulus $(3, 30)$ kPa and Poisson's ratio $(0.4, 0.45)$ such that they cover the range of values describing adipose tissues [164]. St Venant-Kirchhoff material is chosen to model tissue mechanics, since it represents the simplest generalization of elastic material to the non-linear regime while ensuring a good trade-off between simulation time and accuracy. The IOS is then extracted from the deformed configuration relying the same method used for the extraction of attachment points, in a range between $(10, 100)\%$ of the entire surface, to simulate the partial view which is acquired by vision sensors intra-operatively. For each point of the PBM which belongs to the IOS, we compute the displacement field which brings it to its deformed counterpart (i.e., the IOD), while we associate all the remaining points with zero

displacement.

The computed IOS, together with the PBM, represent the input to our network. This input is converted into the grid-like structure required by the method following the voxelization process described in Section 4.2 above. The ground truth binary mask of attachments is defined by assigning a value of 1 to all grid cells which contain a fixed mesh node. An overview of the data conversion pipeline is provided in Fig. 4.3. Following this process, we generate a dataset composed of of 5000 samples, which are split into training and validation sets ($90 - 10\%$).

BA-Net architecture

The output of our method is a 3D binary map defined in the same domain as the input grid, where unitary values are assigned to each grid voxel containing attachment points (Fig. 4.2e). Our framework relies on the UNet architecture, which has been already successfully applied to learning deformation tasks on 3D grids [106, 14, 165, 15, 107]. These works have confirmed UNet capability of learning both high-level representations of the data in the bottleneck layers and carrying high-level information using the skip-connections.

BA-Net architecture is illustrated in Fig. 4.3c. The network first contracts the input data into a 4^3 volume, to make sure that information coming from displacement fields taking place on one side of the input surface can influence the opposite side of the surface. The encoded information is then expanded back to original 64^3 space and converted into a 3D binary map via a final 1^3 convolution layer. Similarly to [14], interpolation operators are employed in the decoding path, to save computational time required by the standard up-convolutions. In addition, dropout layers at 50% are added after each max pool and interpolation operations, which allow to improve generalization capabilities of the network. The loss function L used for training is a linear combination of the Dice Similarity Coefficient (DSC) [166] and the Binary Cross Entropy (BCE):

$$L = \frac{1}{N}\sum_{1}^{N}(1 - DSC + BCE) \tag{4.2}$$

where N is the batch size, which is 32 in our case. DSC and BCE are defined as:

$$DSC = \frac{2 \cdot (X \cap Y)}{X + Y} \tag{4.3}$$

$$BCE = -Y \cdot log(X) - (1 - Y) \cdot log(1 - X) \tag{4.4}$$

with X and Y representing the prediction and the ground truth respectively. The network is trained with AdamW optimizer [167] and one cycle learning rate scheduler [168], on a workstation with Intel Xeon CPU and NVIDIA GeForce 2080 Ti GPU.

4.3 Model update with BA-Net during soft tissue manipulation

Performances of our pipeline for intra-operative PBM update relying on BA-Net have been assessed on soft tissue manipulation experiments of (1) synthetic phantoms with varying

geometric configurations and material properties, and (2) ex vivo human kidney. In both scenarios, both simulation and real world experiments have been conducted.

In all our experiments, we assess the capability of the method to update the PBM by comparing the deformed state in the simulated environment with the available ground truth deformed configuration. The deformed state in the simulated environment is obtained by performing a finite element simulation where model APs are defined by BA-Net. A state-of-the-art direct solver [36] is used together with an iterative Newton-Raphson method to solve the non-linear system of equations in the static domain, within the open-source SOFA framework [148]. Simulated deformed configurations are obtained by considering the displacement of the surface nodes grasped by the surgical tool as driving input. This modelling choice does not only allow to obtain a deformation profile which is independent of possible inaccuracies in chosen material parameters [147], but also to tackle real surgical scenarios, where there is no information about the tool-tissue interaction force (which is lacking also in current real surgical robotic systems).

We then compute the Average Volume Error (AVE), which is the average of the Euclidean distances between corresponding points in the simulated and the ground truth configurations. Let $\mathbf{x_{ground}}^i$ be the position of the n points present in the ground truth configuration and $\mathbf{x_{sim}}^i$ be the position of the corresponding points in the configuration obtained from the simulation using BA-Net predicted APs, then the AVE for sample i is computed as:

$$AVE^i = \frac{1}{n}\sum_n ||\mathbf{x_{sim}}^i - \mathbf{x_{ground}}^i||_2 \tag{4.5}$$

4.3.1 Experiments on synthetic phantoms

The first set of experiments aims at evaluating performances of BA-Net on soft tissue manipulation of synthetic phantoms with varying geometric configurations and material properties. Since ground truth APs are available in all these experiments, we consider two metrics to evaluate the overall accuracy of APs prediction: the DSC coefficient and the True Positive Rate (TPR) [166]. The DSC coefficient (Eq. (4.3)) is a measure of the overlap between the predicted and the ground truth areas. High values of the DSC coefficient are obtained when intersection between prediction and ground truth is maximized, and union is minimized. Despite being highly correlated with prediction accuracy, the DSC is strongly impacted by errors in the delineation of the contours of the region of interest. On the other hand, the TPR (or sensitivity) measures the proportion of ground truth attachments that are also identified by the prediction:

$$TPR = \frac{X \cap Y}{Y} \tag{4.6}$$

The TPR is not influenced by errors at the boundaries, thus might be more interesting than DSC in some applications, where it is more important to ensure that the predicted region includes all the true attachments even at the cost of introducing some errors in boundaries delimitation.

Furthermore, we evaluate the simulation error when relying on BA-Net predicted attachments with the AVE described above.

Simulated data

A test dataset composed of 380 simulated samples (*Test*) is generated following the same pipeline described in Section 4.2.1. The random simulations result in samples which have a median input displacement of 38.8 mm.

In addition to assessing the simulation error obtained when relying on BA-Net using the AVE, we report the metrics values obtained when ground truth attachments are used as APs (we refer to this configuration as *Same*). Metrics for the *Same* configuration are introduced since the deformed states with BA-Net predicted APs are obtained from simulations driven by an input displacement, and are compared to ground truth deformed configurations generated by applying an input force (Section 4.2.1). Therefore, the computed AVE does not only contain the errors made by approximating the boundary conditions with BA-Net predictions, but also the error made by replacing the force input with a displacement input. As a consequence, AVE for the *Same* configuration provides a baseline value that allows to isolate the AVE contribution due to imprecise estimation of attachment points.

We also detail the metrics when naive initializations of the boundaries are used. In particular, we fix (i) no points (*Zero*), (ii) all the points belonging to the lowest surface (*All*). Comparing the metrics on the test set with those obtained on these representative configurations makes it possible to assess how BA-Net predictions impact the simulation error. We also isolate values relative to the subset of test dataset whose samples have an associated visible surface below 50%, a condition which is closer to real cases (*TestV*).

In the analysis with simulated data, we also report the values of the Maximum Volume Error (MVE). MVE for sample i is defined as:

$$MVE^i = \max ||{\mathbf{x_{sim}}}^i - {\mathbf{x_{ground}}}^i||_2 \tag{4.7}$$

This metric is not computed on real data, since it is highly influenced by the presence of errors in the computation of corresponding points.

Real world phantom data

BA-Net performances are tested on real world soft tissue manipulations performed with a single Patient Side Manipulator (PSM) arm of the da Vinci Research Kit (dVRK) [8], as shown in Fig. 4.4. We fabricate four deformable phantoms using commercially available addition curing silicone rubber (Smooth-On Ecoflex materials) with different geometric shapes (circle, clover, rectangle and drop), thickness ($6, 6, 5, 12$ mm) and elastic properties (obtained by using silicone rubber with different shore hardness). For each phantom, we choose 3 configurations of fixed points, which are schematically summarized in Fig. 4.5. The position of fixed points is defined by stitching the phantoms on a 3D printed calibration base (dimensions $144 \times 144 \times 4$ mm), called Reconfigurable Attachment Board (RAB), with regularly spaced holes in a 7×7 grid (distance between adjacent holes is 18 mm). Thanks to the RAB, each set of attachment points can be mapped to the corresponding simulated mesh to generate the virtual ground truth. For each attachments configuration, we select 3 or 4 grasping points distributed over the unconstrained portion of the phantom (see Fig. 4.5). Furthermore, for each configuration we consider two starting conditions: the former where the phantom lies flat on the RAB, the latter where a snap-fit capsule structure is added to the RAB to introduce an initial pre-deformed state to the phantoms (Fig. 4.4), similarly to the real scenario where

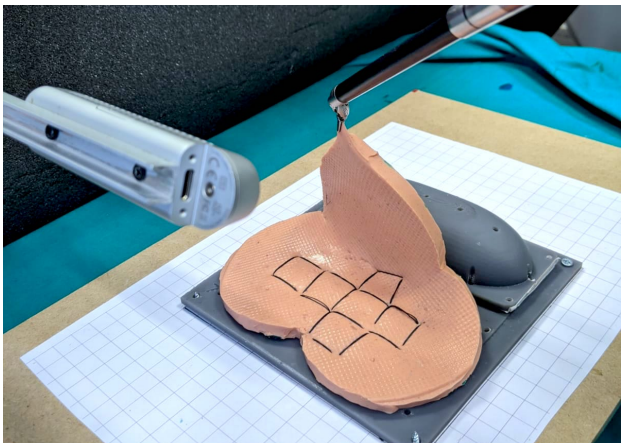


Figure 4.4: Real world experiments on phantom data. A single dVRK PSM interacts with deformable phantoms stitched to the calibration base in correspondence of the attachment points. The PSM lifts the phantoms from their rest configurations. A snap-fit capsule is placed on the experiment board to induce a pre-deformation. The RGBD camera is used to acquire the point cloud of the deformed configurations.

fat tissues lie on the kidney. In our experiment, we lift the tissue from each grasping point to the maximum feasible extent and we record the point cloud representing the current state of the surface at regular steps of 10 mm, while increasing the lifting. The point cloud is acquired by an Intel RealSense D435 RGBD camera and is automatically registered to the virtual geometry thanks the initial system calibration, which is performed following the same process described in [169]. The displacement field IOD relative to the visible phantom surface is retrieved by computing a dense point-to-point matching problem between the acquired point clouds and the undeformed phantom surface, following the process described in Section 4.2.

Similarly to the evaluation conducted on simulated data, we run forward simulations using ground truth APs as boundaries (*Same* configuration) also here. However, in real world experiments, AVE obtained with the *Same* configuration represents the error due to modelling inaccuracies, i.e., the difference between simulation and reality due to chosen simulation parameters (e.g., mesh discretization, constitutive relationship).

4.3.2 Experiments on ex-vivo human kidney

Validation of the presented pipeline is carried out in realistic conditions on an ex vivo pararenal fat tissue manipulation (Fig. 4.6a). Tissue's PBM is initialized with the specimen 3D geometry, generated from manual segmentation of its CT scan, and discretized with 65,538 tetrahedral elements. Its biomechanical properties are selected to be aligned with those observed for adipose tissues, as descrived above. Leveraging on the constructed PBM (Fig. 4.6b), it has been also possible to generate a synthetic dataset based on a real anatomical model. Such dataset allows us to evaluate method performances within a scenario which is influenced neither by sensor noise nor by inaccuracies introduced by ZoomOut, since the IOD can be estimated by directly matching corresponding points in the deformed and the undeformed configurations.

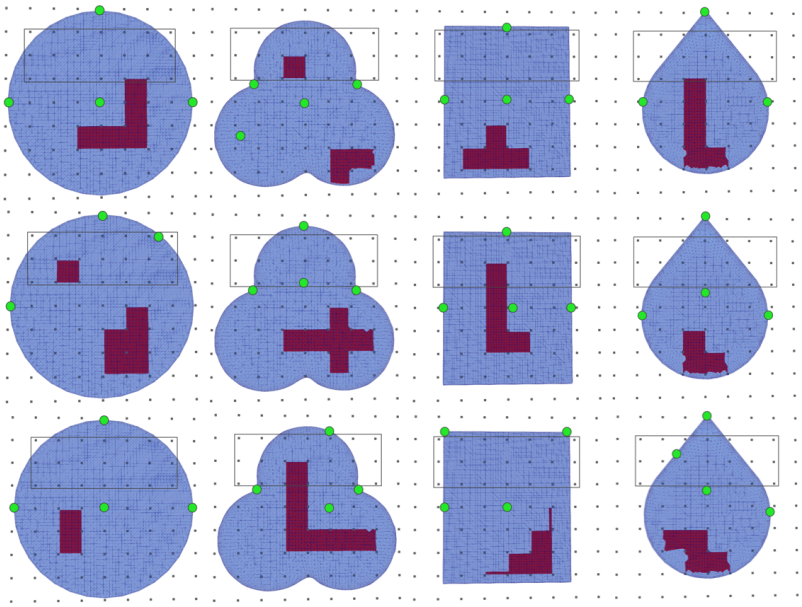


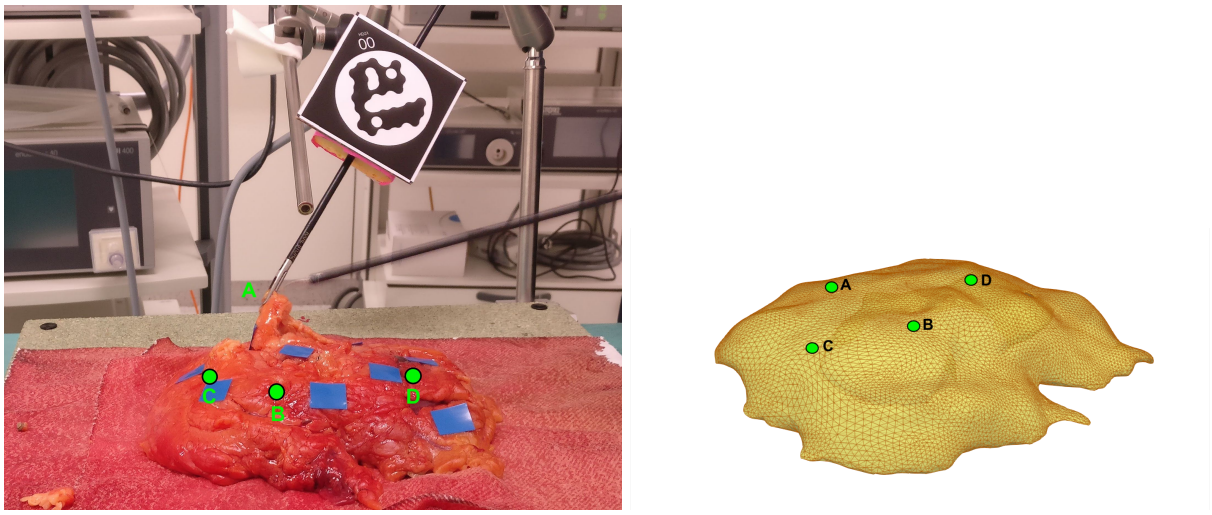
Figure 4.5: Configurations of attachments considered in the real world phantom experiments. Red areas represent the defined attachment regions; green spots correspond to grasping points. Configurations are overlaid to the grid of RAB points, which allows to uniquely map the positions between the real and simulated environment. Gray rectangle defines the position of the snap-fit capsule that allows to obtain pre-deformed configurations.

Simulated pararenal tissue manipulation

A synthetic dataset of 600 samples of adipose tissue manipulation is generated following the training data generation strategy described in Section 4.2.1, but keeping the considered PBM fixed (Fig. 4.6b).

Real pararenal tissue manipulation

The complete intra-operative pipeline (Fig. 4.2) is employed to update the PBM during real ex vivo pararenal tissue manipulation. An expert surgeon is asked to grasp the human tissue with a laparoscopic tool from four pre-defined points, whose position is known in the 3D model space thanks to CT-visible markers, and lift it to the maximum reachable extent that prevents tissue tearing (Fig. 4.6a). After pulling the tissue from all the points, the surgeon introduces a change in APs by dissecting some tissue adhesions, and repeats the acquisitions. Acquired data are relative to three different initial states (i.e., two dissection stages); however, it has not been always possible to lift from all the grasping points due to excessive tissue damage introduced by dissection. Intra-operative point clouds capturing deformed tissue states are acquired through an Intel RealSense D435 RGBD camera. Collected images allow to extract the displacement applied to the tissue by the surgeon, by tracking a STag marker attached to the instrument (Fig. 4.6a) [170]. To evaluate the presented method, point clouds at regular lifting steps of $10\,\text{mm}$ are extracted and passed through the complete pipeline described in Section 4.2. As soon as a new estimate of APs is available, we perform a FE simulation to the obtained IOM applying the same input displacement applied by the surgeon. We then calculate the AVE between the acquired point cloud and the deformed



a b

Figure 4.6: (a) An example tissue manipulation. The surgeon manipulates the pararenal fat and anterior fascia with a laparoscopic instrument, grasping from point A. Point clouds of the tissue state are acquired with an RGBD camera capturing the scene from the same perspective of the picture. (b) The PBM of the ex-vivo perinephric tissue. Letters and green circles are placed in correspondence of the grasping points considered for the real world manipulation experiments.

IOM, relying on correspondences estimated by ZoomOut.

Furthermore, these experiments on realistic clinical settings have been used to evaluate the performance of the proposed intra-operative pipeline in terms of computation time.

4.4 Experimental results on synthetic phantoms

Simulated data

Values of evaluation metrics on test dataset (*Test*) are detailed in Table 4.1. The MVE obtained in simulations with predicted attachments has a median of $7.0(4.0 - 12.7)$ mm. Median MVE for the *Same* configuration is $5.2(2.24 - 10.4)$ mm. In most of the samples, the maximum volume error is found close to the grasping point, which usually corresponds to the maximum deformation.

Table 4.1: Evaluation metrics, as median (25th-75th percentile), for the simulation test samples.

	DSC	**TPR** [%]	**AVE** [mm]
Zero	0.00 (0.00-0.00)	0 (0-0)	27.0 (13.4-52.0)
All	0.42 (0.31-0.55)	93 (84-100)	6.7 (2.5-12.4)
Same	1.0 (1.0-1.0)	100 (100-100)	0.6 (0.2-1.7)
Test	**0.82 (0.70-0.88)**	**86 (75-91)**	**1.1 (0.5-2.1)**
TestV	0.76 (0.63-0.83)	83 (69-90)	1.2 (0.6-2.4)

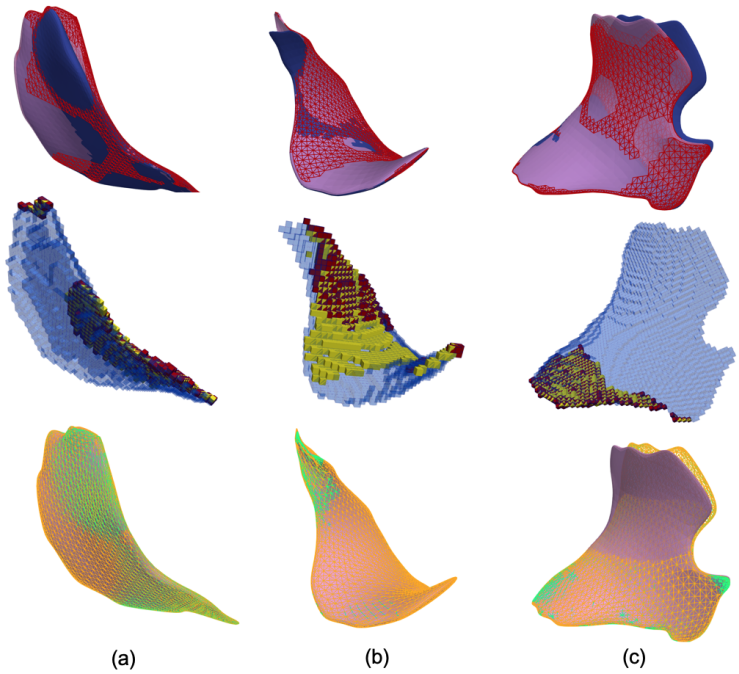


Figure 4.7: BA-Net predictions for three simulated cases (one per column). First row: initial PBM (blue), deformed surface (pink) and visible IOS (red wireframe). Second row: voxelized initial PBM (sdf $<$ grid voxel size) (blue), ground truth APs (red) and predicted APs (yellow). Last row: deformed surface (pink), deformed surface when using predicted APs (orange), and deformed surface when using ground truth APs (green). (a) Sample composed of two disjoint regions of attachments, correctly predicted by the network. (b) Sample associated to a low visible IOS (36%). In this case, BA-Net overestimates the attached region. (c) Sample characterized by a good prediction accuracy but non-zero AVE. However, green and orange meshes are perfectly superimposed, thus high AVE is due to the different simulation method and not inaccurate prediction.

Real world phantom data

Table 4.2 summarizes the results obtained for the experiments conducted on the real phantom experiments, grouped following different criteria. First, we consider the results relative to all the acquisitions associated to the same phantom (first four rows), to understand if the performances are influenced by the different geometries and properties. Secondly, we analyze metrics values at different levels of input displacement ($2, 4, 6$ cm from the rest configuration), to evaluate the influence of the lifting height on the predictions. Results are further grouped depending on the starting configuration (flat or pre-deformed). The AVE reported in Table 4.2 represents the error between each point in the acquired point cloud and the corresponding one in the deformed configuration, relative to the AVE at rest (median value 4.8 mm). Error at rest includes the contributions of registration error, inaccuracies in the computed correspondences and sensor noise. Table 4.2 also details the values of AVE *Same*, which allows to assess the magnitude of the error we are making even when using the optimal APs, thus solely caused by discrepancy between simulation and reality. Visual examples of the network predictions are provided in Fig. 4.8. Finally, we want to assess the robustness of the predictions at different grasping points, for the same configuration of attachments. To this end, we compute metric values considering the intersection of the predictions at different grasp-

Table 4.2: Evaluation metrics on the real experiments, expressed as median (25th-75 percentile). Last column reports the number of acquisitions which contributed to the statistics of the corresponding row.

	DSC	**TPR** [%]	**AVE** [mm]	**AVE *Same*** [mm]	**#samples**
Circle	0.28 (0.16-0.40)	68.7 (33.3-93.3)	3.2 (2.1-5.0)	3.5 (2.4-4.8)	101
Clover	0.40 (0.26-0.46)	62.8 (48.8-79.0)	3.1 (1.8-5.8)	3.0 (2.0-4.7)	79
Rectangle	0.43 (0.37-0.51)	86.7 (66.5-100.0)	3.7 (2.1-7.3)	4.2 (2.5-6.2)	85
Drop	0.54 (0.42-0.66)	75.6 (52.6-89.7)	3.4 (2.0-5.0)	3.6 (2.4-5.0)	80
Lift 2cm	0.41 (0.31-0.48)	72.6 (49.4-93.8)	3.9 (2.2-6.1)	3.8 (2.6-5.4)	85
Lift 4cm	0.44 (0.29-0.53)	70.6 (54.7-86.7)	5.0 (2.1-7.4)	4.8 (3.1-5.7)	44
Lift 6cm	0.44 (0.37-0.56)	76.3 (55.4-89.9)	3.9 (1.7-7.9)	4.5 (1.9-7.2)	15
Flat	0.41 (0.30-0.50)	82.1 (60.1-97.8)	2.8 (1.8-4.8)	3.1 (2.3-4.8)	198
Pre-deformed	0.40 (0.27-0.49)	65.1 (34.7-84.3)	4.1 (2.4-7.2)	3.8 (2.6-5.7)	147
Grasp	0.50 (0.23-0.56)	51.5 (35.4-60.0)	7.4 (4.5-9.8)	–	24

ing points, for the same lifting height (in our case 4 cm), i.e. fixing only those points which are predicted as fixed from all the grasping points (Table 4.2, last row). Fig. 4.9 provides some visual examples of these predictions. It is worth noting that real world experiments are associated to a visible surface below 50%, which is the most challenging condition.

4.4.1 Discussion

Simulated data

BA-Net predictions maximize both DSC and TPR, showing that the network has learnt to generalize to new unseen geometries and configurations of fixed points (Fig. 4.7). The median AVE achieved by simulations with predicted attachments is 1.1 mm. Since this value is obtained as an average error over the entire considered geometries, which includes both internal and surface points, it represents an overall precise matching between the ground truth and the deformed state obtained with predicted attachments. This precise matching is further supported by considering the large applied input displacement, with a median value of 38 mm. Although the median MVE might seem large, its value differs from the one obtained when running forward simulations with ground truth attachments (*Same* configuration) by less than 2 mm. The fact that MVE is not zero with the *Same* configuration indicates that there is a baseline error introduced by applying an input displacement instead of a force (Fig. 4.7c). Overall, simulation accuracy has significantly improved with respect to the cases where a naive initialization is given to the boundary conditions, i.e. when fixing either zero or all the points. Prediction accuracy is impacted by more challenging conditions, i.e. limited input information, as confirmed by metrics values when the visible surface is below 50% (*TestV*), that are slightly worse than the ones on the entire dataset.

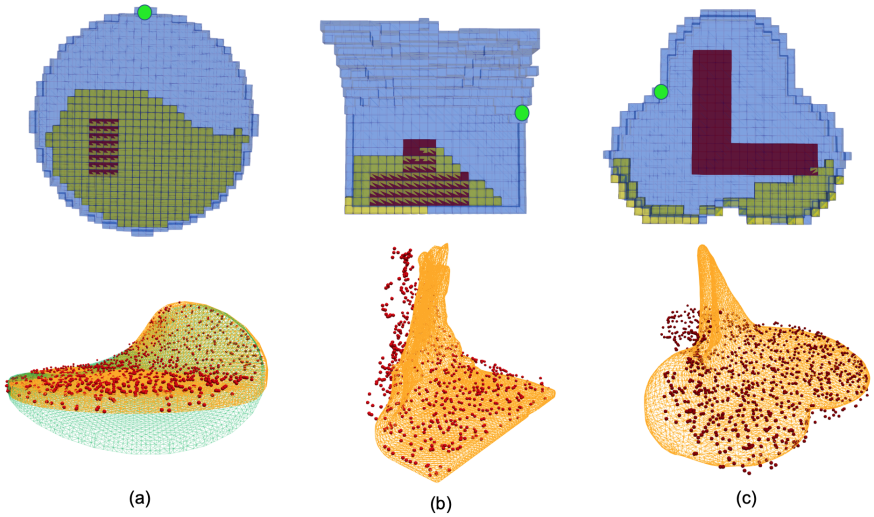


Figure 4.8: BA-Net predictions for three real cases (one per column). Upper row: voxelized initial PBM (sdf $<$ grid voxel size) (blue), ground truth APs (red) and predicted APs (yellow). The considered grasping point is indicated in green. Lower row: acquired point cloud (red) and deformed surface when using predicted APs (orange). (a) Sample at a lifting level of 2 cm. The green mesh overlaid on the bottom configuration represents the deformed surface when using ground truth attachments. (b) Sample at a lifting level of 4 cm, starting from an initially deformed configuration. (c) Sample at a lifting level of 3 cm. In this configuration, the PSM occludes the upper part of acquired point cloud.

Real world phantom data

Table 4.2 shows that values of the evaluation metrics are aligned for all the different experimental conditions. BA-Net is able to handle different geometries and material properties, with only slight differences. Worst results in terms of prediction accuracy are obtained for circle and clover. The reason for this is twofold. Firstly, they are the only samples for which we tested a configuration of attachments composed of two disjoint areas. Although samples with more than one attachment region are present in the training dataset (Fig. 4.7a), the network always predicts a single fixed region on the real data (Fig. 4.9b). The second reason for these suboptimal values is that the attachment region is often overestimated, as emerges by visual inspection of the results (Fig. 4.8a). However, this is not indicating bad prediction performance: BA-Net learns to model not only fixation points but also constraints imposed by the environment. In fact, the network predicts the whole area where the phantom is in contact with the RAB as attached, which constrains phantom motion but is not taken into account by the ground truth. This allows to achieve an overall good matching between simulation with predicted attachments and real deformed state, which actually outperforms the simulation result obtained when ground truth points are fixed (Fig. 4.8a). Although the best trade-off between the metrics assessing prediction accuracy is obtained for rectangle and drop, these shapes are associated to a higher AVE with respect to circle and clover. This is probably due to the fact that the considered constitutive law is not able to accurately describe the behavior of these two phantoms, fabricated with a different silicone rubber, which showed some time-dependent behavior (Fig. 4.8b). We expect that this error could be reduced by using a more accurate biomechanical model or injecting real samples in the dataset, but such fine tuning

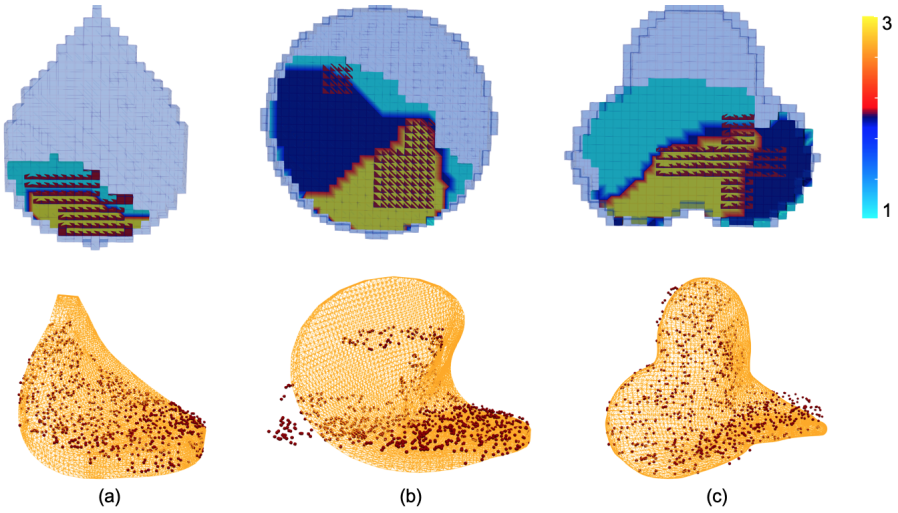


Figure 4.9: Network performance when considering the intersection of the predictions at different grasping points, for the same configuration of attachments, in three different real cases. Upper row: voxelized initial PBM (sdf < grid voxel size) (blue) and ground truth APs (red). Predicted APS are rendered according to a colormap which maps a region with yellow if all the predictions considered it as attached. Lower row: acquired point cloud (red) and the deformed surface when fixing points predicted as attached by all configurations (orange).

was out of the scope of this work.

Fig. 4.8c shows a failure case for BA-Net, which corresponds to the configuration associated with the leftmost grasping point. The heavy occlusion introduced by the PSM causes dramatic geometrical noise that perturbs the matching retrieved by [163], in particular due to ruined surface estimation. Even though BA-Net prediction is poor in this case, this is caused by limited surface visibility and not by failure of the method itself. The inaccurate matching could be resolved in the future by either injecting prior knowledge to the method (e.g. trusted landmarks or temporal constraints) or by introducing the second dVRK PSM, to limit occlusions.

BA-Net predictions are not influenced by the magnitude of the input displacement, as confirmed by the fact that higher input deformation introduces a limited gain in DSC and a slight reduction in TPR variability. The increase in the AVE values is due to the fact that bigger input displacements are more likely to introduce some instabilities in the simulations. We note that there is a slight difference in the performances depending on the starting state: even if the DSC is comparable between the two conditions, experiments starting from initially flat configurations are associated to higher percentage of correctly identified attachments. The difference in the AVE can be partially due to suboptimal 3D mesh when the configuration is initially deformed. While the geometry could be directly extracted from pre-operative data in real scenarios, for these experiments we had to warp the flat configuration based on the rest point cloud in the deformed state, which introduced some low quality regions that can have a negative impact on simulation accuracy.

BA-Net predictions are coherent when varying the grasping point (Fig. 4.9). However, when single states of deformation are viewed, the region of attachment points is often overestimated, likely because the network is unsure whether a region which does not move is fixed or not. Last row of Table 4.2 shows that the DSC tends to increase when using the

intersection of the predictions from multiple views. This indicates a better matching of the actual attachment region, even though it sometimes comes at the expenses of some missed regions (lower TPR). A reason for this behavior is provided in Fig. 4.9b and c, which shows that if just one of the predictions misses a region, it will be excluded from the intersection (leading to worse AVE as well). In future works, we plan to test other strategies for combining the predictions obtained from different grasping points, for example weighting the different contributions based on the likelyhood of the predictions.

In general, median values for the DSC coefficient seem quite low if compared to the ones obtained on simulated samples. However, metrics values must be interpreted with respect to the devised application. As highlighted in Section 4.1, the network could be exploited to support an autonomous surgical system in two ways: either by improving simulation accuracy in case of actions replanning or by providing a guess of the attachment region to guide the actions of the ARSS (for example during tissue retraction or dissection).

If we focus on the application of BA-Net for the improvement of simulation accuracy, we can see that simulations performed with the boundary conditions predicted by BA-Net lead to an AVE which is better than the one obtained when fixing ground truth points in most of the cases. This is due to the fact that simulations fixing ground truth points do not take into account the constraint provided by the RAB, while the network seems to learn to account for that. Simulations with ground truth attachments can be thought of as a real surgical scenario where we have some a priori knowledge about the attachment area, but no guess about the constraints provided by the surrounding anatomical environment. The reduction in AVE introduced by BA-Net tells us that the method has the potential to improve simulation accuracy with respect to the case when some a priori knowledge about the area of interest is available. Although obtained AVEs might seem large in absolute terms, we have to consider that it is obtained with sub-optimal simulations, characterized by quite coarse anatomical models and rough modelling assumptions. Using a higher resolution geometry and object-specific constitutive law would help to reduce such error. In the current work, our main focus was the assessment of the general prediction capabilities of the method and its ability to update biomechanical simulations, thus we relied on models that could guarantee a good trade-off between accuracy and computational performance.

The other possible application of BA-Net in surgery is the identification of APs to support autonomous tissue manipulation tasks. In this case, it is preferrable that most of the attached area is correctly identified, even at the cost of having added regions or inaccuracies in boundary delimitation. In this context, having high TPR values is more important than having a good DSC, because we want to minimize the amount of missed regions. Achieving a median TPR above 62% for all the real world phantom experiments tells us that most of the attached area is correctly identified. This is an interesting result if we consider that such accuracy is achieved by providing a single partial view of the deformed state of the tissue as input and without relying on any prior information and considering that the net is trained with only simulated samples. Even expert surgeons would find it challenging to precisely identify the attached area from a single manipulation, especially if the applied displacement is small. What experts would generally do is to move towards the expected area of attachments and perform further manipulations of the tissue, until they are confident enough about the location of the attached points. We expect that BA-Net would benefit from a similar approach, and future works will focus on improving the prediction by providing sequential frames as

input.

4.5 Experimental results on ex-vivo human kidney

Simulated pararenal tissue manipulation

The median AVE (25th-75 percentile) calculated on all the 3D model points between the deformed IOM and the corresponding reference sample is 1.4 $(0.7 - 3.2)$ mm (Fig. 4.10). Since ground truth APs are available for the simulated dataset, we assess prediction accuracy by computing the DSC, similarly to what we did for the synthetic phantom experiments. Median (25th-75 percentile) value for DSC is 0.51 $(0.40 - 0.60)$. Fig. 4.10b and c show that the network is challenged by the complex geometry and fails to accurately identify APs when they are distributed along the sharp edges of the mesh, especially when the amount of visible surface is very small and does not capture the region undergoing the highest deformation. However, when the visible surface captures the region with greatest deformation (Fig. 4.10b), BA-Net is able to provide a plausible prediction, which leads to a precise matching between the simulated and the reference configuration.

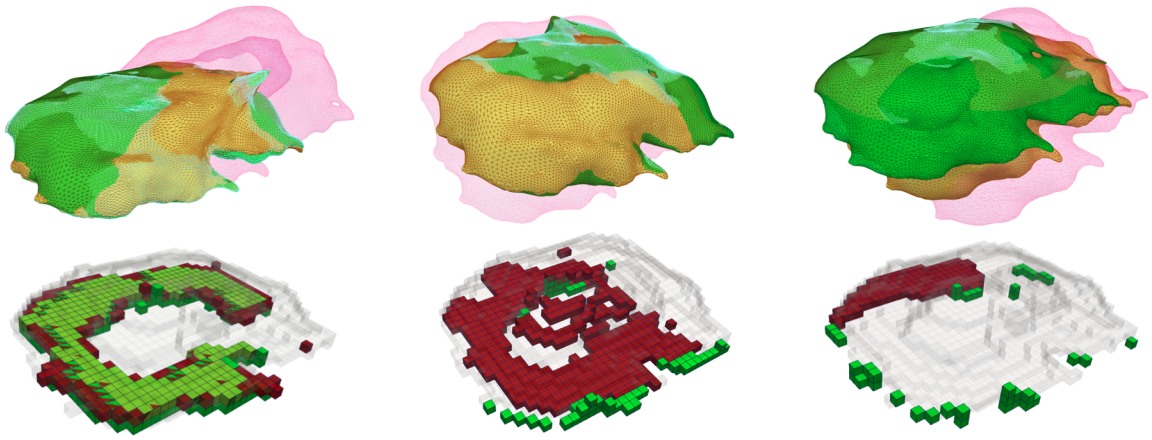


Figure 4.10: Results on the simulated pararenal tissue dataset considering input displacements above 25 mm. Upper row: ground truth deformed configuration (green) and simulated deformed configuration obtained when using predicted APs (yellow), with the considered visible IOS (light blue). Pink mesh represents the deformed configuration when APs are unknown, thus undefined (*Zero* configuration). Lower row: ground truth APs (green) and predicted APs (red) in grid space. (a) Good overlap between simulated and ground truth configurations (AVE=1.6 mm), due to good prediction accuracy (DSC=0.68). (b) Good overlap (AVE=1.7 mm) is possible even if prediction is not the same as the ground truth one (DSC=0.09). (c) High AVE (14.6 mm) is due to an inaccurate prediction (DSC=0.02), when visible surface does not provide enough information about tissue state.

Real pararenal tissue manipulation

Table 4.3 reports the average AVE over 10 runs of the whole pipeline, relative to the error at rest (average 3.62 mm, which includes contributions of segmentation and registration errors)

Table 4.3: AVE between ground truth point clouds and their corresponding points in the virtual environment when using predicted APs, at increasing deformation levels [mm]. Acquisitions are grouped by grasping point (A, B, C, D) and initial tissue state (1, 2, 3). Reported errors are relative to the error at rest and represent the average over 10 runs of the entire pipeline. Missing values are due to failures in instrument tracking. Last column reports the AVE obtained when no APs are defined in the virtual environment (Mean *Zero*).

Grasp	**State**	10 mm	20 mm	30 mm	40 mm	50 mm	**Mean**	**Std**	**Mean *Zero***
A	1	-	0.94	1.06	1.28	6.81	2.52	2.48	24.93
	2	-	-	1.42	2.42	2.75	2.20	0.65	31.99
B	1	2.96	6.75	-	-	-	4.86	1.90	13.02
	2	5.05	9.91	11.47	-	-	8.81	2.84	27.06
	3	3.27	11.71	9.12	3.92	-	7.01	3.60	31.53
C	1	-	1.65	2.49	-	-	2.07	0.42	18.66
	2	5.30	2.54	3.84	-	-	3.89	1.42	14.23
	3	2.59	4.61	6.93	10.76	-	6.22	3.04	18.79
D	1	1.07	-	-	-	-	1.07	0.00	7.72
	2	-	1.32	1.57	3.98	5.25	3.03	1.71	19.76
	3	1.51	2.28	3.57	-	-	2.45	0.85	7.29
Mean		3.11	4.64	4.61	4.47	4.94			
Std		1.50	3.90	3.52	3.32	1.80			

at increasing pulling levels. The reason why we report results over multiple runs is that we rely on an approximate nearest neighbour for ZoomOut to minimize the computational overhead, which might however introduce some differences between different runs. The average time required to update the synchronous PBM simulation is 1.44 ± 0.14 s (with an average of 0.039 s dedicated to pre-processing, 1.39 s to displacement estimation, and 0.013 s to BA-Net), tested on a workstation with an AMD Ryzen7 3700X CPU and NVIDIA RTX 2070 SUPER graphics card.

4.5.1 Discussion

Simulated pararenal tissue manipulation

Results on the simulated dataset show that when predicted APs are used to characterize the simulated environment, the simulated state differs from the corresponding reference of less than 2 mm on median, indicating an overall good matching (Fig.4.6a). Best alignment between the two configurations is achieved when the visible surface captures the area of greatest deformation. This fact does not introduce a limitation for the real clinical settings, where the intra-operative sensor is looking at the area of manipulation, which usually corresponds to the most deforming part.

Real pararenal tissue manipulation

Average AVEs obtained on real ex-vivo tissue manipulation remain below 5 mm in almost all the cases, which is aligned with the accuracy levels required for model-guided intra-operative

applications in the context of minimally-invasive surgery [14, 107, 15]. Furthermore, obtained AVEs are significantly lower than the ones achieved when APs are unknown, thus they cannot be defined and the simulation remains unconstrained (*Zero* configuration, last column in Table 4.3). In general, we notice that AVE increases with increasing deformation, in a more significant way with respect to what observed on real world phantom experiments (Section 4.4). This might be due to the fact that accurate simulation of real adipose tissues is more challenging than synthetic phantoms, probably requiring more sophisticated modeling choices (e.g., highly non-linear constitutive laws handling inhomogeneities). Moreover, the high deformations may cause poor surface estimation from [171], which assumes that the point cloud already represents the desired surface without noise or topological artifacts. In such case, ZoomOut promotes a disturbed correspondence (i.e., isometric to a wrong surface). To improve this step, investigating the point cloud denoising techniques seems a promising future direction. This further motivates the worse AVE values obtained in correspondence to grasping point B, whose point clouds are partially occluded by the surgical tool. If we consider the computational performances, we notice that our pipeline can provide a new estimate of the APs in less than $2\,\mathrm{s}$, with ZoomOut taking most of the time. This update rate allows to rely on a simulation environment which can reflect the changes in the surgical scenario with a very short latency.

Overall, these results suggest that the presented pipeline can be used to successfully update a PBM exploiting data coming directly from intra-operative sensors, while respecting both accuracy and time constraints compatible with standard minimally-invasive surgical applications.

4.6 Conclusion

In this Chapter, we have presented a complete pipeline that allows to update a patient-specific pre-operative model, starting from the raw data acquired during the intervention. Our pipeline relies on BA-Net, a deep neural network that provides an estimate of tissue attachment points starting from the pre-operative PBM and a single partial intra-operative view of the deformed tissues. Obtained results on both simulation and real world experiments have shown that the proposed strategy can improve a biomechanical simulation intra-operatively by updating the attachment points (i.e., simulation boundary conditions), reducing the difference bewteen the real and the simulated environments to less than $5\,\mathrm{mm}$ in all the tested cases. This accuracy level is aligned with the one required for model-guided intra-operative applications in the context of minimally-invasive surgery [14, 107, 15]. Furthermore, the presented pipeline provides a new estimate of tissue attachments within a very short delay (i.e., less than $2\,\mathrm{s}$). This allows to update the simulated model reflecting the changes in clinical settings with a latency compatible with standard surgical workflows, potentially making the method able to handle situations with dynamically changing APs, for example involving dissection, sutures removal or topological changes.

Conducted experiments have highlighted some common limitations, which will guide future research and development. First of all, quality of BA-Net predictions has shown to be impacted by the complex geometry of real tissues as well as inaccurate estimation of the displacement field introduced when dealing with real data, due to the presence of sensor noise. Future work will focus on making the synthetic training dataset more representative

of real world conditions, including strategies to generate more realistic pre-operative geometries (e.g., with surface irregularities) and simulating point cloud noise when extracting the IOS. We expect that this will help BA-Net to implicitly learn how to cope with the challenging realistic conditions.

An additional direction for future research relies on the improvement of BA-Net predictions by providing the network with a more informative input. Identifying APs from a single partial view of the deformed tissues would challenge even expert surgeons, especially if the applied displacement is small. Therefore, we plan to consider a sequence of intra-operative tissue states as input to the network, potentially including multiple grasping points. This strategy seems particularly promising to improve the robustness and coherence of the predictions, within an environment which is intrinsically evolving in time.

Furthermore, real world experiments have shown that the quality of the final result is influenced by the different sources of errors that are introduced throughout the various stages of the pipeline, from an imprecise initial rigid alignment, to the presence of sensor noise and inaccurate computation of corresponding points. In future works, we plan to improve the preprocessing stage, for instance by reconstructing the point cloud from the stereo-endoscope view [172]. Inaccurate surface matching can be addressed by either letting the network implicitly solve the surface correspondence problem as in [14], by providing salient points extracted from camera view to ZoomOut, or by improving surface estimation [173]. In particular, relying on DNN to directly solve for surface correspondences seems promising to further improve the time performances of the current implementation, where ZoomOut is responsible of the main computational overhead.

Contributions of this Chapter

The main contributions of this Chapter are the following:

1. We introduce Binary Attachment Network (BA-Net), a deep network that estimates attachment points of a deformable organ from a single partial view of the intra-operative surface, without any a-priori knowledge of their location.

2. We demonstrate that the method can update a PBM reaching clinically acceptable performances, both in terms of accuracy and intra-operative time constraints, on soft tissue manipulation of (1) synthetic phantoms with the da Vinci Research Kit and (2) ex-vivo pararenal tissue.

3. We publicly share the full pipeline for intra-operative update of a PBM starting from the raw sensor data as well as the collected dataset of soft tissue manipulation on synthetic phantoms at https://gitlab.com/altairLab/banet.

Publications linked to this Chapter

The content of this Chapter has been presented in the following publications:

1. Tagliabue E, Dall'Alba D, Pfeiffer M, Piccinelli M, Marin R, Castellani U, Speidel S, Fiorini P: *Data-driven Intra-operative Estimation of Anatomical Attachments for Autonomous Tissue Dissection*, IEEE Robotics and Automation Letters (2021);

2. Tagliabue E, Piccinelli M, Dall'Alba D, Verde J, Pfeiffer M, Marin R, Speidel S, Fiorini P, Cotin S: *Intra-operative Update of Boundary Conditions for Patient-specific Surgical Simulation*, International Conference on Medical Image Computing and Computer-Assisted Interventions (2021); 2021 Sept 27-30; Strasbourg (France).

Part II

Simulation for autonomous surgery

Chapter 5

Learning a surgical task in simulation

5.1 Introduction

Automating a surgical intervention is an extremely complex task, due to the many different actions that are required to complete a procedure. For this reason, the research trend in autonomous surgery has broken down the problem and focused on the automation of individual tasks taking place during surgery. Following the definition of [174], tasks are sequences of activities which aim at a specific surgical objective and is recurrent in many different procedures, e.g. tissue dissection, suturing and knot tying. One of the challenges when attempting to automate surgical actions is that they all involve the interaction with the soft anatomical environment, thus requiring the development of control strategies that can cope with such dynamic behavior. For example, a common task in surgery is Tissue Retraction (TR), which requires the manipulation of soft tissues in order to expose a hidden region of interest. TR is performed during robot assisted nephrectomy procedures, where the surgeon has to interact with the highly deformable pararenal adipose tissue which covers the kidney in order to have access to the underlying tumor.

Some works have attempted to automate TR with standard motion control algorithms. For example, Nagy et al. have proposed to automate TR using soft computing methods [175]. In [176], authors have proposed a trajectory planner for TR based on coordinates extracted directly from the intra-operative image feed. Both these works have automated TR relying on pre-defined movement sequences without considering tissues dynamics, posing some doubts about its possible exploitation in realistic anatomical environments. In [177, 178], authors employ standard path planning methods to execute an optimal plan for the task generated within a biomechanical simulation in the pre-operative phase, in this way accounting for the deformation properties of the anatomy. However, all these approaches rely hand-crafted control policies, which make the execution of complex non-linear trajectories and behaviors challenging.

Most of the prior works concerning the automation of actions involving soft tissues manipulation rely on Learning from Demonstrations (LfD), where a task is learnt by imitating expert behavior [179, 180, 181, 182]. LfD is a preferred approach since it allows to learn proper interaction with deformable tissues without the need to explicitly model of their behavior. In addition, it does not require to explicitly design policies. However, the robustness of learned tasks to changes in the initial conditions or in the environment is strongly affected

by the amount and variety of expert demonstrations provided to the system [183]. Collecting a dataset with a vast repertoire of trajectories from multiple experts and varying initial conditions is impractical and often unfeasible in clinical settings. Moreover, with LfD the robot can only become as good as the human's demonstrations, but there is no additional information for improving the learnt behavior.

Deep Reinforcement Learning (DRL) has shown promising results in the automation of robotic tasks like object manipulation and pick-and-place tasks, without the need to design ad-hoc control strategies [184]. In DRL, the component which makes decisions about the action to take is called agent. A DRL agent reaches robust performance after it has explored a huge amount of possible policy options, which requires a large number of interactions with the environment, many of which fail with unsafe behavior [185]. Although safe learning methods have been proposed, their exploitation in the surgical robotic field remains limited, due to the impracticality to acquire many training trials from real surgical robotic systems [186]. For this reason, existing works based on DRL learn surgical tasks in simulated environments, to enable the many trial and error attempts required to train agents in safe and controlled settings [187, 188, 189, 190]. One of the main challenges when training in simulation is that the *reality gap* (i.e., the discrepancy between the simulated and real environments) must be kept as low as possible to enable successful deployment of the learnt policies in the real world, in what is called *sim-to-real* approach [191]. In the context of RMIS, this means that the simulation should account for both deformable properties of the anatomy and the interaction with the surgical tools. This is the main limitation of *dVRL*, a simulation framework to train DRL agents for surgical tasks proposed by Richter et al. [192], which is based on VREP and OpenAI Gym, and supports only rigid objects. Because of the reality gap, simulation-learnt behaviors have been transferred to the real surgical robotic system only for simplified geometries [188]. Only recently, Xu et al. have proposed *SurRoL*, a simulation-based platform for DRL that can simulate deformable objects and is interfaced with the dVRK [193]. *SurRoL* has shown promise for successful transfer of behaviors learnt in simulation to the real world.

Recent research has proposed to combine LfD and DRL with the aim of exploiting the strength of both approaches and overcome their respective drawbacks [194]. In particular, demonstrations can be used to guide the exploration done during learning, reducing the time required to find an improved control policy, which may depart from the demonstrated behavior. A promising strategy is represented by Generative Adversarial Imitation Learning (GAIL) [195], which has been successfully applied to endovascular manipulators but never used in RMIS [196].

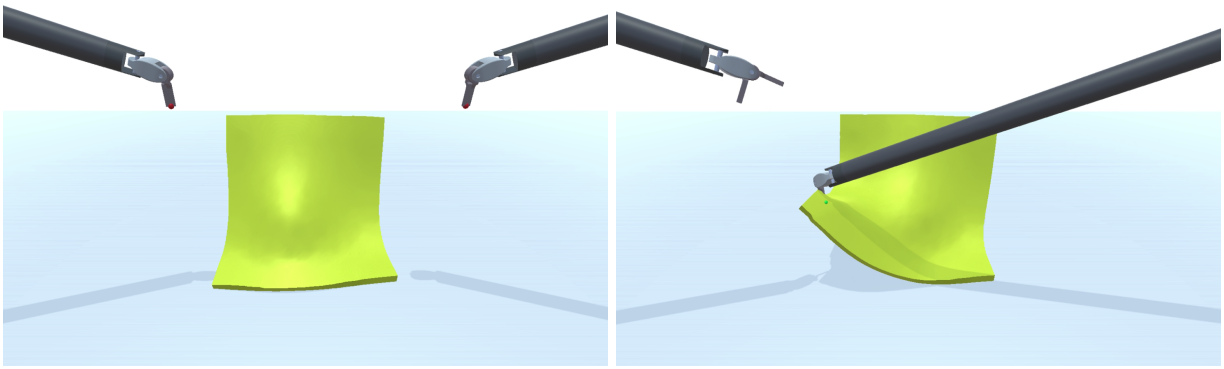
In this Chapter, we focus on the role that simulation can play to support the process of learning surgical tasks. By providing a realistic replica of the real scenario, simulation represents an exceptional resource for two reasons. First of all, it allows to perform the large amount of trial and error attempts required by learning-based methods in controlled settings, enabling the exploitation of state-of-the-art approaches without the need to access the real surgical systems, which is both impractical (sometimes unfeasible) and risky. Second, it provides an environment where to test the learnt behavior before its exploitation on the real robotic system, allowing to predict potentially unsafe or dangerous situations and prevent from their execution in the real world. Our main contribution is the development of a modular framework called *UnityFlexML* that provides an interface between a realistic simu-

lation of deformable anatomy, the surgical robotic system and learning-based methods. In particular, in the context of this Thesis, we consider DRL and LfD techniques. We show that the developed framework is suitable for both prototyping and testing two different learning approaches: a standard DRL method requiring multiple interactions with the environment and an approach that combines LfD and DRL, on the automation of a soft tissue retraction task. Finally, we demonstrate that the learnt policies can be successfully transferred to the real system, thanks to the high level of realism achieved by the simulation environment.

5.2 UnityFlexML: a framework to learn surgical tasks in simulation

UnityFlexML is a modular framework that allows to exploit learning-based methods to learn tasks in a simulated surgical environment which involve deformable objects (Fig. 5.1). The developed platform provides an interface between the real dVRK and a simulation of the real environment.

The framework is made publicly available at https://gitlab.com/altairLab/unityflexml.



a b

Figure 5.1: *UnityFlexML* framework. The simulated dVRK arms interact with deformable tissues, modelled using PBD method. Example scene (a) at rest, (b) during tissue manipulation.

5.2.1 Robot platform

UnityFlexML tackles automation of surgical tasks executed with the Patient Side Manipulator (PSM) arms of the dVRK. The dVRK is programmed using the Robot Operating System (ROS) [197]. In order to have a state space (i.e., the set of all possible configurations of the robotic agent) which is straightforward to observe in both the simulator and the real robot, the motion of the PSM end-effector (EE) is controlled in the Cartesian space, keeping the EE orientation constant (as in [192]). The framework can handle any possible surgical tool, provided that the corresponding kinematic model is loaded in simulation. Therefore, the PSM EE state is characterized by its position $\mathbf{p}_t$ and gripper state $g_t \in \{0, 1\}$ for open/close. Similarly to [192], we normalize the PSM positions with respect to the workspace, defined

by the PSM joint limits and the obstacles in the environment, to allow generalization of the learned policies to various joint configurations.

5.2.2 Simulation environment

The simulation is implemented within Unity3D, an engine whose main advantage is the high modularity, which allows users to easily customize the environment scene and exploit advanced features implemented in separate plugins [198]. Our framework relies on two main Unity3D plugins: the Machine Learning Agents Toolkit (ML-Agents), for training intelligent agents [199], and NVIDIA FleX, for soft object deformation [149]. Deformable bodies are simulated with the Position Based Dynamics (PBD) approach relying on the optimized implementation provided by NVIDIA FleX. This library has proved to be able to accurately model the deformable behavior of anatomical tissues with highly efficient computational performances (see Section 3.8) and it is thus particularly suitable to perform the huge number of trial and error attempts required for agent training with learning-based methods. Our choice to simulate anatomical deformations based on PBD instead of relying, for example, on the finite element method to obtain a physically accurate result (Section 2.3.1) is motivated by the superior performance of PBD in terms of computational efficiency and numerical stability. These two aspects are of major concern in our framework, which has to allow the simulated agent to perform multiple interactions with the environment in the lowest time to guarantee efficient training, thus also minimizing the risk of simulation instability.

From what concerns the simulation of the robotic part, we have implemented a closed form inverse kinematics of the PSMs to enable the Cartesian space control of the manipulator. Data exchange between Unity3D and ROS exploits UDP-based communication, using the method described in [200]. At each simulation step, the robotic system is allowed to perform a very small motion increment. As a consequence, we could assume the impact of the robot dynamic behavior to be negligible, thus we have not accounted for it in the simulation [192]. In *UnityFlexML*, grasping of an object is modelled as an atomic event triggered when the relative distance with the EE is less that 2 mm.

5.3 Background on learning-based methods

The aim of *UnityFlexML* is to provide an integrated simulation framework where learning-based methods can be trained. Before presenting how the framework can be used for this purpose, we introduce the main concepts in learning-based approaches.

5.3.1 Reinforcement learning

Reinforcement Learning (RL) denotes a learning paradigm where an agent learns behaviors to achieve its goal while interacting with the environment (Fig. 5.2). A general RL problem can be formulated as a Markov Decision Process (MDP) described by the tuple $\langle S, A, R, P \rangle$, where:

- S is the set of possible states of the environment;
- A is the set of possible actions of the agent;

- P is the transition probability distribution, with $P(s_{t+1}|s_t, a_t)$ being the probability of transitioning into state s_{t+1} when starting in state s_t under action a_t;
- R is the reward function, with $r_t = R(s_t, a_t, s_{t+1})$ being the immediate reward after transition from s_t to s_{t+1} with action a_t.

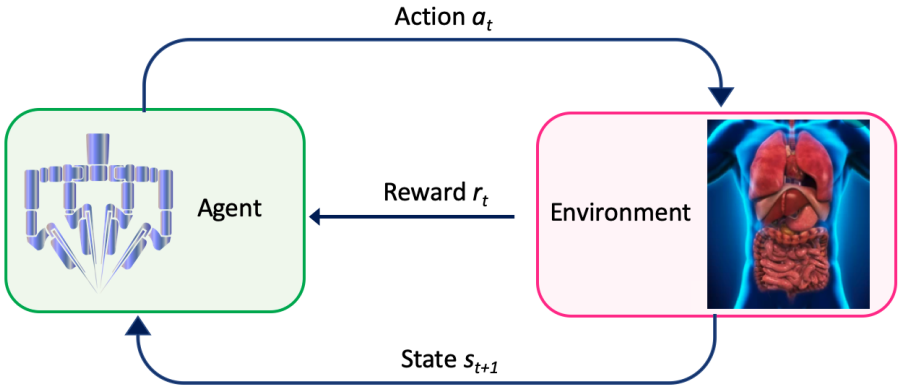


Figure 5.2: In RL, an agent takes an action a_t in an environment. In turns, the environment communicates a new state s_{t+1} and a reward signal r_t back to the agent.

At every step t of interaction with the environment, the agent receives an observation of the state s_t, and then decides on an action a_t to take. As a consequence of the action, the agent moves to the new state s_{t+1} and receives a reward signal r_t associated with the transition (s_t, a_t, s_{t+1}). The goal of the agent is to learn a behavior policy π, i.e. the rule used to decide which actions to take, that maximizes the expected cumulative reward E$[R(\tau)]$. $R(\tau)$ is commonly defined as:

$$R(\tau) = \sum_{t=0}^{\infty} \gamma^t r_t \tag{5.1}$$

which considers the sum of discounted rewards, where the discount factor $\gamma \in [0, 1)$ is introduced to weight events in the distant future less than events in the immediate future. $R(\tau)$ is sometimes defined over an episode, i.e. the sequence of states and actions over a fixed time horizon T. Due to the high dimensionality of the state space S in many practical problems, deep learning methods have been incorporated in the standard RL setting to represent the policy or other functions, leading to what is called Deep RL (DRL). The expected return obtained when starting from state s and successively acting with policy π is called value function $V_\pi(s)$. The optimal value function $V^*(s)$ is the one obtained when acting according to the optimal policy π^*, the one that achieves the optimal values in all the states. Instead, the action-value function $Q_\pi(s, a)$ gives the expected return when starting from state s, taking a first random action $a \in A$ and subsequently acting following policy π.

RL algorithms are classified in different ways depending on what they learn and how. A first categorization deals with the availability of a model of the environment, which allows the agent to consider a range of possible action plans and predict state transitions and rewards. Such model can be either given or learnt in a first stage. Several different approaches to model-based RL have been proposed [201]. Some of them compute optimal plans with respect to the model each time a state observation arrives, relying on pure planning techniques

like model-predictive control [202]. An alternative approach learns the policy that produces actions as similar as possible to an expert planning algorithm [203].

However, having a model of the environment is generally quite difficult, since the dynamics of the objects involved are not always known. To tackle this issue, model-free RL methods are popularly used [201]. While being easier to implement and tune, model-free RL is generally less sample efficient (i.e., it needs more experience to learn) than model-based RL. Model-free approaches can either optimize the policy directly (policy optimization methods) or learn the optimal action-value function (Q-learning methods). Policy optimization algorithms rely on data collected while acting according to the most recent version of the policy (on-policy optimization). On the other hand, Q-learning approaches indirectly optimize for agent performance relying on data collected at any point during training, regardless of the specific policy used when such data were obtained (off-policy optimization). This causes Q-learning to be less stable but generally more sample efficient than policy optimization, since it can reuse data more effectively.

Proximal policy optimization

Proximal Policy Optimization (PPO) is one of the most popular RL methods. It is a model-free on-policy RL algorithm which continuously alternates between collecting a new batch of observations and improving the policy [204]. In particular, PPO maximizes a surrogate objective function that estimates how much the expected return will change after parameters update. A common approach to train a PPO agent in DRL relies on two deep networks, one called *actor* and other called *critic*. The actor network receives state observations from the environment and learns what action to take under a given observed state. Whereas, the critic network receives the reward signal following the action from the environment and learns to evaluate if the action taken by the actor led the environment in a better or worse state. The critic sends its feedback to the actor, which exploits such information to improve the policy.

5.3.2 Learning from demonstrations

The most challenging step in standard RL consists in the design of the reward function. Manually shaping a good reward function to teach an agent to perform a task is immensely difficult, especially for complex tasks [205]. An approach that allows to learn without requiring a reward function is Learning from Demonstrations (LfD). LfD entails those methods that can learn from observed behaviors of other agents (generally human experts), specifically Imitation Learning (IL) and Inverse Reinforcement Learning (IRL).

In IL, the observed behavior is exploited to learn the optimal policy, without requiring interaction with the environment. For example, behavioral cloning learns to replicate the expert's policy from state-action pairs of expert trajectories in a supervised fashion [206]. However, it often suffers from poor generalization since this approach learns to imitate actions without any cue about the reasoning behind the task, so it is not able to understand if the actions are relevant for the final goal.

Whereas, the goal of IRL is not to use the demonstrated behavior to learn the policy, but to learn the reward function that could explain the expert behavior. The benefit of IRL over IL is that the learnt reward function is portable to other environments, thus leading to better

generalization capabilities. However, IRL algorithms are very expensive to run, since they require RL in an inner loop to extract the imitation policy from the learnt cost function.

Overall, both IL and IRL usually require a large number of expert trajectories, which are often not easy to get.

Generative adversarial imitiation learning

Generative Adversarial Imitation Learning (GAIL) represents a promising approach to learn human decision-making strategies from a limited set of expert demonstrations using deep neural networks [195]. In particular, GAIL combines LfD with a Generative Adversarial Network (GAN) to learn both the unknown policy and reward function. GANs consist of two networks: the *generator* and the *discriminator* [207]. The role of the generator is to generate exploration trajectories that should become more and more similar to expert trajectories as the training proceeds. The goal of GAIL is to train generators such that they learn to behave in the same way as the given experts. Meanwhile, discriminators serve as the reward functions for RL, which judge whether the behaviors look like the experts. Usually, the adopted policy generator relies on PPO. There are two reasons why PPO is used for GAIL: first, it uses smooth policy update for stable learning and second, PPO generates diversified trajectories that act as a wide sampling range for the discriminator in GAIL.

5.4 Learning tissue retraction within UnityFlexML

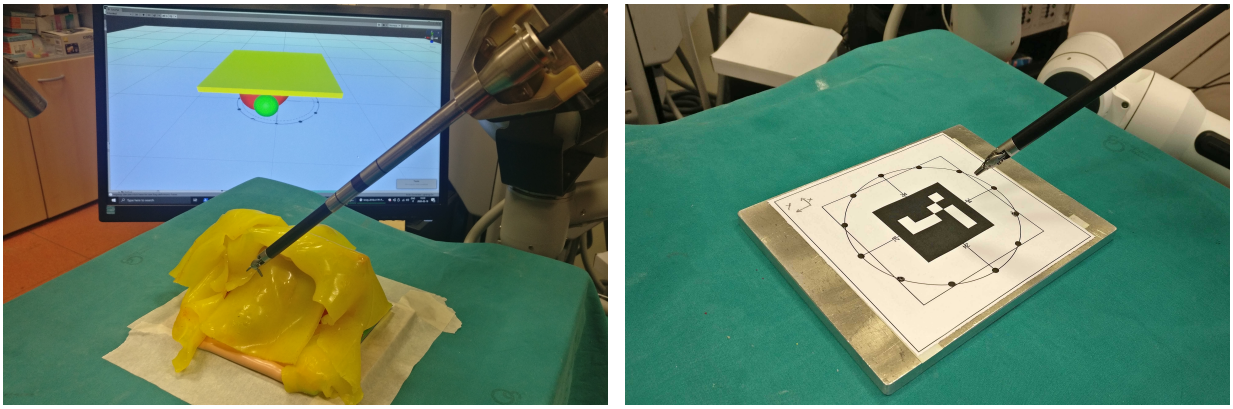
The presented *UnityFlexML* is employed to train agents in a simulated environment to learn a soft tissue retraction task. In particular, we aim at learning the sequence of actions that allow to accomplish the task and consists in reaching the adipose tissue covering the kidney, grasping it and lifting it in order to expose a tumor. In the following, after describing our experimental setup, we detail how we adapt the two considered approaches, i.e. a standard DRL approach and GAIL, to learn the desired task with our platform.

5.4.1 Experimental setup

Our real experimental setup consists of a synthetic kidney phantom covered with silicone fat tissue, shown in Fig. 5.3a. We restrict the portion of fat tissue our agent interacts with to a 90×90 mm square region which is rigidly anchored to the top part of the kidney. The silicone patch representing the fat is held in place through a custom designed rigid structure, which enables to uniquely define the position of fat and kidney between the simulated and the real environment. The initial configuration of the fat tissue is obtained by letting this square region fall on the kidney phantom under gravity load. In our experiments, we consider a scenario with a single PSM arm equipped with the Large Needle Driver. All the simulation experiments, including agents training and dVRK control, ran on a workstation equipped with an AMD Ryzen 3700X processor and NVIDIA TitanX GPU.

UnityFlexML environment

The 3D model of the kidney phantom used to inizialize the simulation environment as well as the position of the tumor $\mathbf{q}$ are extracted from segmentation of the Computerized Tomog-

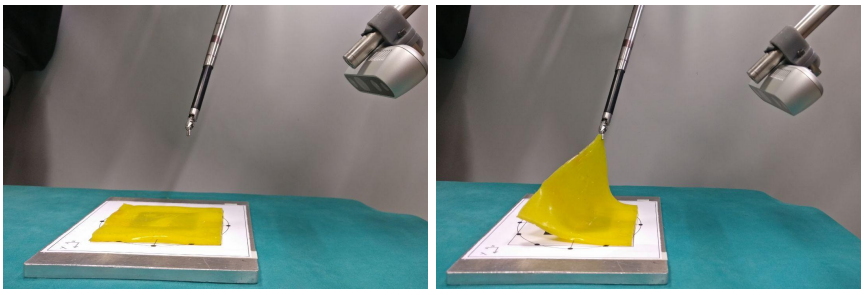


a b

Figure 5.3: (a) In our setup, a single arm of the dVRK interacts with silicone fat tissue covering a kidney phantom. The simulated scene controlling dVRK movements within *UnityFlexML* can be seen in the background. (b) The calibration board used to uniquely map all the components of our real experimental setup to the *UnityFlexML* environment.

raphy (CT) of the phantom.

In an effort to minimize reality gap, we perform a preliminary optimization procedure to find the PBD deformation parameters that describe the behavior of our synthetic adipose tissue, keeping the real and the simulated behaviors as close as possible. We employ the genetic algorithm scheme to optimize PBD parameters most impacting the deformable behavior of the fat tissue our robot interacts with, similarly to what has been described in Section 3.5.1. Optimization is performed on some preliminary experiments where a teleoperated PSM arm lifts the fat tissue, which starts from a planar configuration and is rigidly fixed on one side (Fig. 5.4). We define $N = 5$ different pinch points along fat contour and $L = 3$ different levels of lifting are defined for each pinch point. The point cloud representing ground truth positions of the fat tissue is acquired using an Intel RealSense D435 Depth camera (Intel Corporation, Santa Clara, USA), whose position is defined with respect to a custom calibration board which allows to rigidly align the simulated and the real environment (Fig. 5.3b).



a b

Figure 5.4: One of the experiments of the optimization process. The fat tissue is anchored to the calibration board (right side in Fig.). (a) Rest condition; (b) Deformed condition. Point cloud of the deformed tissue is acquired with the depth camera shown on the right.

Optimal values for the cluster spacing, cluster radius and cluster stiffness parameters are

estimated by minimizing the following error ϵ:

$$\epsilon = \frac{1}{N} \sum_{n=1}^{N} \sum_{l=1}^{L} \sum_{m=1}^{M} ||\mathbf{x_{PBD}}(l,n) - \mathbf{x_{PCL}}(l,n)||_2 \tag{5.2}$$

where $||.||_2$ represents the Euclidean distance between the position of the M particles defining the fat in simulation $\mathbf{x_{PBD}}$, at deformation level l and pinch point n, and the closest point of the corresponding point cloud $\mathbf{x_{PCL}}$. The acquired point cloud has been decimated to bring the number of points comparable to M. The diameter of the PBD particles is set to 3 mm (i.e., the width of our tissue sample), which allows to describe the dynamics of the fat tissue with a single layer of particles. The constraints and the range of allowed values for each parameter are set in the same way as described in Section 3.5.

Optimal values for the cluster spacing, radius and stiffness parameters generated with the optimization process are 0.127, 0.095 and 0.361 respectively, which lead to an average error between the simulated and the ground truth point clouds of approximately 3 mm, which is comparable with dimensions of PBD particles. These values are employed to describe the deformable behavior of the fat tissue in the simulation scene.

5.4.2 Learning methods

In our problem, the agent is represented by the end-effector (EE) of the da Vinci PSM, which interacts with the surrounding anatomical environment, whose initial state is assumed to be known from pre-operative data. Our task consists of moving the PSM arm from a pre-defined initial position $\mathbf{p}_0$ to a position close to the tumor $\mathbf{q}$, grasp the fat and lift it to a pre-defined final position $\mathbf{p}_T$. In order to make the learnt motion primitives robust to different initial configurations, the EE starts from a different position $\mathbf{p}_0$ after each episode (i.e., 2500 timesteps in our case) at training time. Whereas, $\mathbf{p}_T$ is considered fixed through all the training experiments. The considered state space leverages solely on kinematics information defining the current robot state and the environment, at time t:

$$\begin{aligned} s_t &= [\mathbf{p}_t, \mathbf{q}, \mathbf{p}_T, ||\mathbf{p}_t - \mathbf{q}||_2, ||\mathbf{p}_t - \mathbf{p}_T||_2, g_t] \\ a_t &= [\mathbf{\Delta}_t] \end{aligned} \tag{5.3}$$

where $||.||_2$ is the Euclidean distance. $\Delta_{t,i} = 0.5\alpha$, $\alpha \in \{0, -1, +1\}$ tells the agent if it has to remain still, move backward or forward by 0.5 mm in the i_{th} spatial dimension, while $g_t \in \{0, 1\}$ represents the gripper state (open/close).

The feasibility of using *UnityFlexML* to learn a surgical task is evaluated using two possible strategies: a standard DRL approach and GAIL.

DRL setting

The first considered method is a model-free on-policy DRL algorithm based on PPO, in its implementation provided by Unity3D ML-Agents. The architecture of the actor-critic networks of the PPO agent used is shown in Fig. 5.5. For the training phase, we design a reward function which varies with the gripper state, to encourage EE motion towards the tumor if the

tissue has not been grasped yet (i.e., gripper is still open), and tissue retraction if the tissue has been already grasped:

$$r_t = \begin{cases} ||\mathbf{p}_t - \mathbf{q}||_2 \cdot k - 0.5, & \text{if } g_t = 0 \\ ||\mathbf{p}_t - \mathbf{p}_T||_2 \cdot k, & \text{if } g_t = 1 \end{cases}$$

The scalar quantity of -0.5 is added to restrict the reward in the range $(-1.0, -0.5)$ before grasping and $(-0.5, 0)$ after grasping. The normalization constant k is introduced to allow re-scaling of the trajectories to different working spaces and is inversely proportional to the maximum distance the PSM can move. During training, the rewards are accumulated at each episode. Being a pure DRL method, this approach is entirely trained in simulation. In the following, we refer to this setting as “PPO".

GAIL setting

The second considered approach relies on the learning paradigm provided by GAIL, using a policy generator based on PPO. The network architecture is depicted in Fig. 5.5. The loss function used in this setting is a linear combination between DRL and GAIL losses $\alpha L_{DRL} + \beta L_{GAIL}$ where α and β represent weighting factors for the two loss functions. Our initial investigation on hyper-parameters tuning yielded best performance for $\alpha = 0.2$ and $\beta = 0.8$. Other values of α and β yielded slower convergence.

Training of a GAIL agent requires the collection of trajectory demonstrations. The acquired trajectories consist of repetitive fat lifting tasks performed by an expert user with the real dVRK, leveraging the communication pipeline provided by *UnityFlexML*. Although our simulation framework supports demonstration recordings using a keyboard or a joystick, the established communication pipeline between dVRK and *UnityFlexML* is crucial since it helps to acquire demonstrations directly with the real robotic system, thus without deviating from the surgical workflow. Overall, we collect a total of 35 task demonstrations. Each recorded demonstration consists of the set of kinematic observations that define the state space and the corresponding action at each timestep. Since the expert is well aware of the final objective of tumor exposure, the grasp position is near the tumor for all the demonstrations. Moreover, the expert user is instructed to diversify the trajectories by starting each demonstration from a different initial position above the fat surface.

5.4.3 Evaluation metrics

To assess the suitability of the framework as a tool for learning surgical tasks, we test the performance of the considered methods to learn the tissue retraction task when training within *UnityFlexML*. In particular, the high level of realism of the simulated environment created within *UnityFlexML* does not only allow us to train the methods with a sim-to-real approach but also provides a platform for testing the presented methods in realistic settings. As a consequence, the learnt behavior is tested both in a simulated environment, provided by *UnityFlexML*, and in the real one, in a sim-to-real fashion.

Performances of the presented algorithms are tested both in simulation and reality based on two different criteria: sample efficiency and optimality of the accomplished task. Sample efficiency denotes the amount of experience an algorithm needs to learn a behavior by

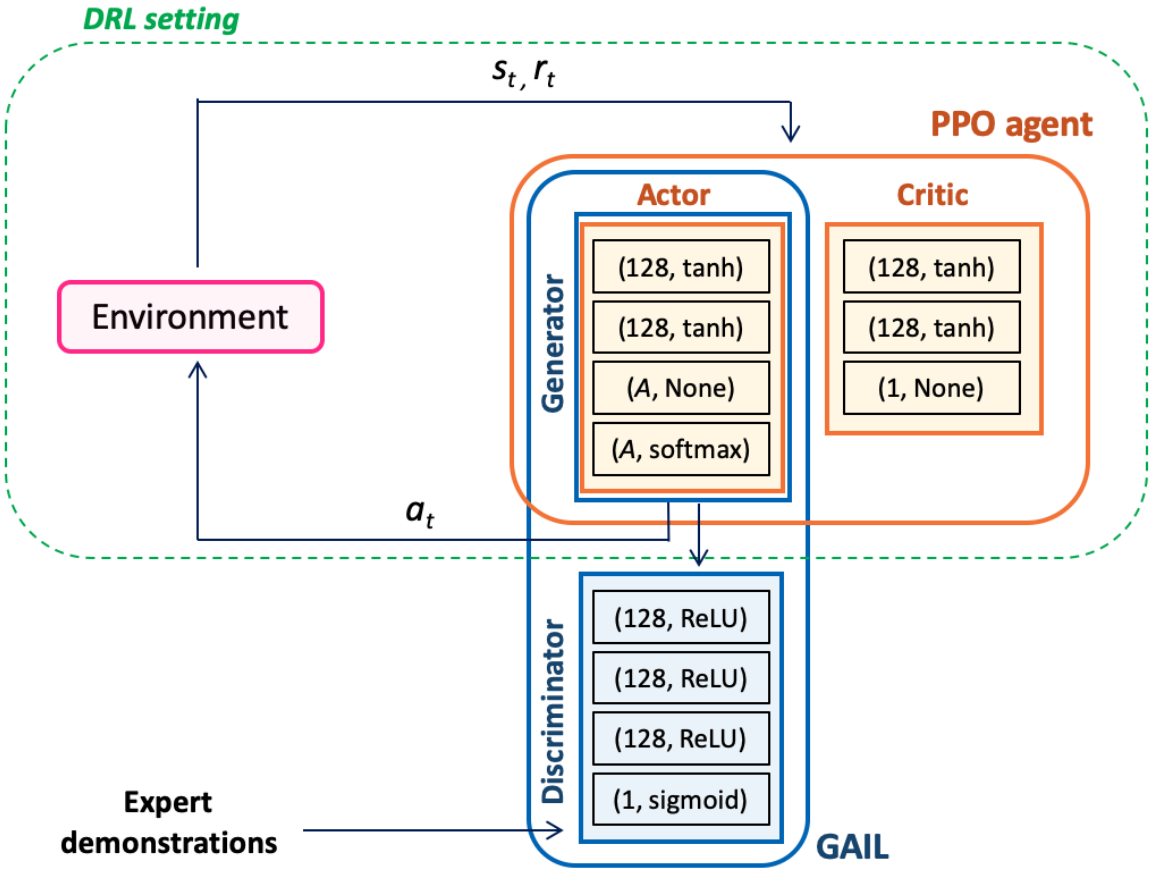


Figure 5.5: Network architecture of DRL and GAIL settings. The DRL setup relies on PPO, which consists of actor and critic network. The actor also acts as the policy generator in the GAIL setting. Generated trajectories and expert trajectories are passed to the discriminator. The discriminator learns a probability function which classifies the generator trajectory as expert or non-expert. Details of each network layers are reported inside each box in the format (hidden units, activation) respectively.

interacting with the environment. It is estimated as the number of time steps required by each algorithm to reach high reward values. Optimality of the learnt behavior represents the ability of each method to make the tumor visible upon task completion, and is assessed using a Tumour Exposure (TE) metric. We consider the image captured by an endoscope positioned in front of the kidney, both for the simulated and the real setup, where we select a circular region of interest around the tumor (exploiting the fact that its position is known). We then extract the visible portion by applying a mask with HSV bounds matching tumor color (Fig. 5.9). TE is computed as the percentage of tumor pixels which are visible within the region of interest, normalized in the range $[0, 1]$.

5.5 Autonomous tissue retraction in simulation

After training within *UnityFlexML*, the capability of the two presented methods to successfully accomplish the tissue retraction task is evaluated in simulation (Fig. 5.6). In the designed

experiment, the trained agents perform the task starting from 49 different positions uniformly sampled on a 7×7 regular grid above the portion of the fat tissue, in order to assess whether the behavior learnt by the agents is robust to different starting positions $\mathbf{p}_0$ of the EE. We evaluate TE each time the EE reaches $\mathbf{p}_T$.

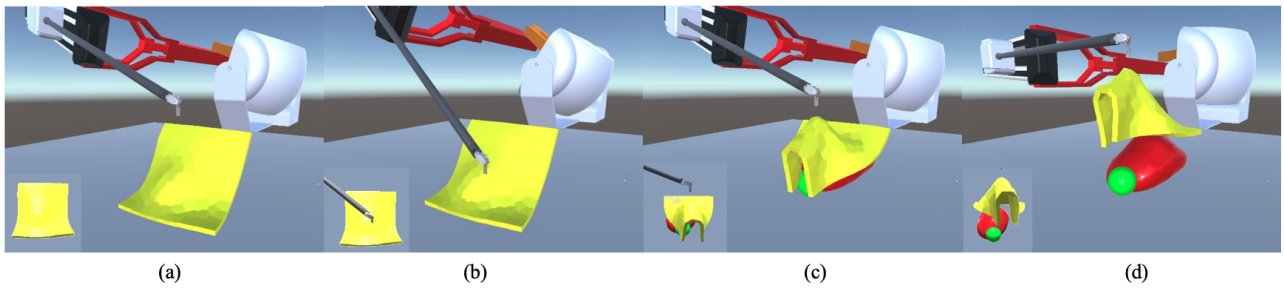


Figure 5.6: Sequence of action frames for task completion in simulation: (a) approach, (b) grasp, (c) retract, (d) expose. Perspective of the simulated camera is overlaid on the bottom left of each simulator frame.

5.5.1 Results and discussion

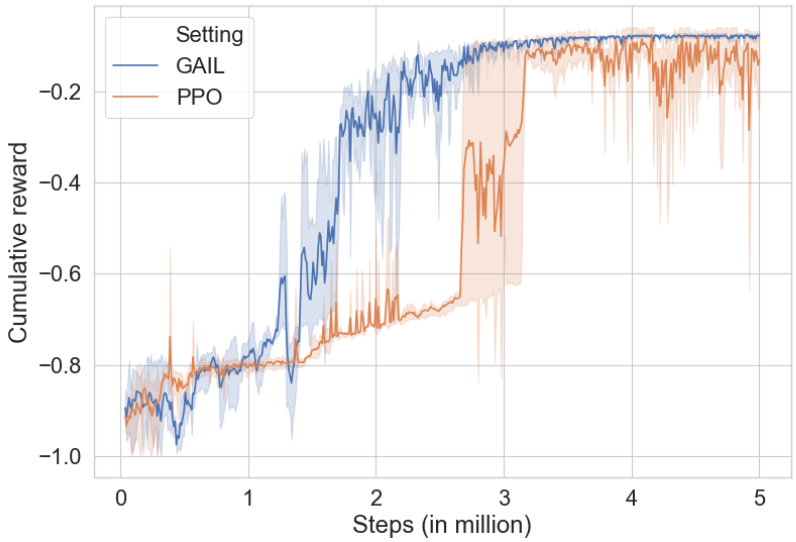


Figure 5.7: The obtained learning curve for PPO and GAIL. Cumulative reward is normalised in the range $[-1, 0]$. The shaded area spans the range of values obtained when training the agent starting from three different initialization seeds.

Fig. 5.7 shows the learning curves obtained with the considered learning strategies. Both GAIL and PPO learn to maximize the overall cumulative reward, but with a different pattern. GAIL learning curve shows a monotonous and smooth trend, since it gradually and continuously increases towards high-reward values, diverging from PPO around 1 million steps. Whereas, PPO shows a modular reward trend: it requires 2.5 million steps to learn the

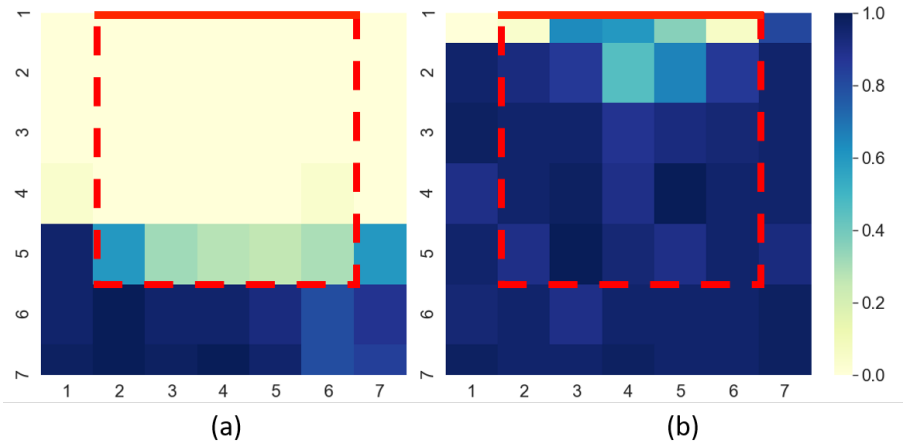


Figure 5.8: Simulation experiments: TE from the camera at different starting positions $\mathbf{p}_0$ for (a) PPO, (b) GAIL. The color of each subregion is related to the percentage of visible tumour area when $\mathbf{p}_0$ belongs to that subregion. The fat boundary from the top view is depicted in red dashed lines whereas the fat attachment is shown in the solid red line

approach behavior and interaction with the fat, and 1 million steps to learn the retract behavior. Overall, GAIL is more sample efficient than PPO since it requires fewer steps to learn the task. This experiment shows that incorporating human demonstrations makes learning sample efficient compared to baseline PPO, confirming that learning efficiency benefits from the incorporation of human knowledge.

The plot in Fig. 5.8 shows the TE from the simulated camera depending on the starting position of the PSM arm above and outside the boundary of the fat tissue. Whenever the agent starts from the distal part of the tissue (i.e., the one farther from the fixed region), agents trained with both PPO and GAIL learn to grasp the tissue and the tumor becomes visible from the camera (at least partially).

However, we note that PPO achieves little or no tumor exposure when the starting EE position is close to the fat attachment (Fig. 5.8a), even though the trained agent has correctly learnt how to perform the task (Fig. 5.7). It seems that, when $\mathbf{p}_0$ is initialized close to the fixed fat region, the agent is not able to move towards a reasonable grasping point, thus causing the tumor not to be exposed. This is probably due to the fact that we rely on a reward function which is suboptimal for the task. The designed reward function encourages the agent to approach the known position of the tumor, but abruptly changes as soon as the EE is in contact with the tissue regardless of its current grasping position. In this way, we might end up grasping at a location which is not optimal to reach the final objective of exposing the tumor. This suggests that manually tuning the reward function to encode complex task objective such as tumor exposure might be challenging, especially relying on kinematic data only. We expect that including TE-dependent term into the reward function would improve the learnt behavior towards the goal of the task. However, our preliminary evaluation with a reward including a TE-dependent term did not show significant improvements in the results. This might be due to the fact that TE is always zero before grasping, thus represents a sparse reward scenario. We plan to further investigate this in future works.

On the contrary, when human demonstrations are incorporated using GAIL, the learnt behavior allows to expose the tumour regardless of the starting position (Fig. 5.8b). It is worth

noticing that the strategy adopted by the user while acquiring demonstrations is to move and grasp towards points close to the tumor, having clear in mind that the aim is to maximize the exposure. The main difference between the behavior learnt by PPO and GAIL lies in the selection of the grasping point at varying starting position. In particular, when the starting position is above the attached area, GAIL grasps closer to the tumor obtaining a higher TE, since it learns to imitate the human operator, who moves towards the most appropriate points to maximize the exposure.

5.6 Sim-to-real autonomous tissue retraction

The behavior learnt with the considered approaches using *UnityFlexML* is transferred to the real robotic platform. Successful task execution on the real setup relies on two main factors. First of all, it depends on the level of realism of the simulated environment where the agent is trained, because the correct task can be learnt only if the reality gap is minimized. Secondly, it is influenced by the accuracy of the alignment between the simulated and the real environment, since all the movements of the dVRK arm in the real system are controlled through the simulated robot, including the grasping action.

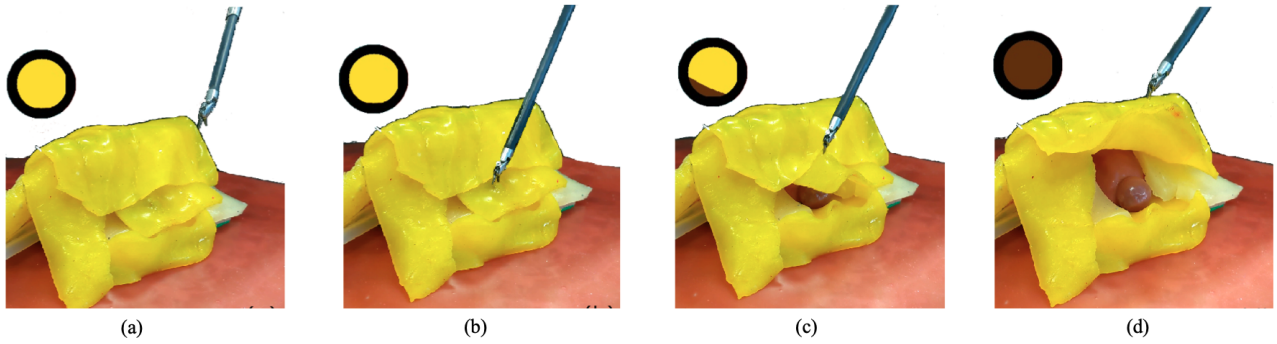


Figure 5.9: Sequence of action frames for task completion in real world setup, with the circular mask used to compute TE metric. (a) approach (TE=0%), (b) grasp (TE=0%), (c) retract (TE=∼15%), (d) tumour exposure (TE=100%). The real camera is placed in front of the phantom, in the same position as in the simulation (which does not correspond to the viewpoint of these pictures).

5.6.1 Results and discussion

The real and simulated setup are initially aligned with respect to the same reference frame, defined by the center of the calibration board (Fig 5.3b). The mean positioning error of the PSM arm is 1.7 mm.

We have been able to successfully replicate the learned behaviour from the simulated to the real environment without any appreciable inconsistency. The dVRK end-effector successfully gets in contact with the fat tissue for all the different initial positions, and it is always able to reach the target point. The tumor exposure percentage starting from various points above the real fat tissue is illustrated in Fig. 5.10. Considering the results obtained in simulation, we did not consider starting positions near the attachment when testing the behavior learnt

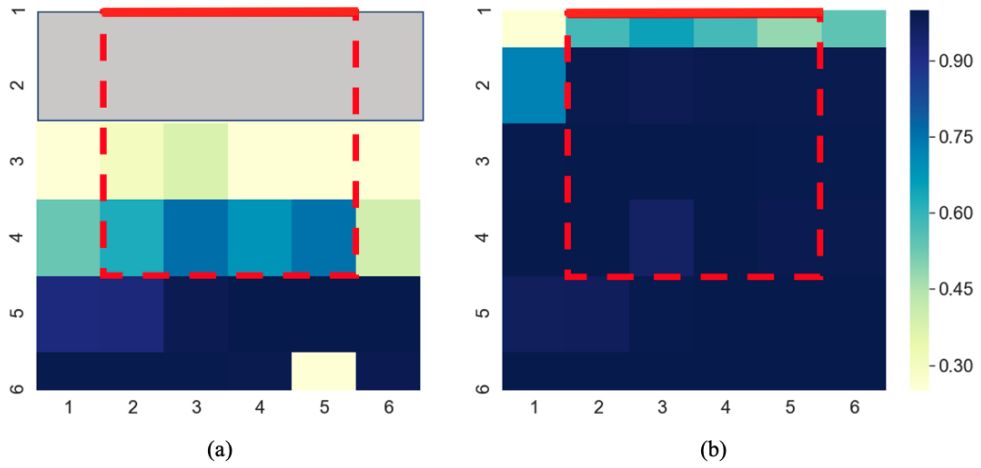


Figure 5.10: Real grasp experiments: TE from the camera when starting from different initial positions of the PSM, using (a) PPO (b) GAIL. The portion of fat tissue which is not considered for the experiments is coloured in gray.

with PPO (represented as the unattempted gray region in Fig. 5.10a) to avoid tissue tearing that might occur in case of grasping too close to the attached area.

If we consider the average TE over all trials from different starting points, PPO achieves an average TE of 0.38 while GAIL obtains an average TE of 0.90. When comparing the results obtained for GAIL and PPO, it emerges that GAIL is not only able to reach higher overall exposure, but it is also more robust to changes in the initial PSM position. In particular, tumor exposure is achieved also when starting from points that were unattempted for PPO (Fig. 5.10b), thus suggesting an overall improvement in performances, due to a more optimal learnt trajectory. This observation indicates that the initial PSM position has a great impact on the performances in the case of PPO, whereas GAIL is able to reach optimal performance regardless of the starting position, confirming results obtained in the simulated experiments. Overall, our results show that performance using demonstrations is robust and outperforms PPO both in simulation and real world.

5.7 Conclusion

In this Chapter, we have shown that simulation can support the process of learning a surgical task. The main advantage of using simulation to this purpose is that it allows to train all those learning approaches that require a high number of trial and error attempts within a personalized, safe and controlled environment. We have designed and implemented *UnityFlexML*, a modular framework which supports simulation of deformable objects, can be interfaced with learning algorithms and communicates with the real robotic system (in our case the dVRK). The platform has proved suitable for training learning-based approaches on a real surgical robotic application, i.e. fat tissue manipulation for the exposure of tumor during robot assisted nephrectomy procedure. The proposed framework does not only allow to train standard DRL approaches entirely in simulation, but it also provides an interface to acquire task executions by expert users on the real robotic system that can be used in LfD settings.

Thanks to a calibration procedure for optimizing the simulation parameters to reduce the reality gap, behaviors learnt in simulation have been successfully transferred to the real robotic system. A further advantage of having a simulation framework is that it can be employed as a verification environment where to test the learnt behavior before its execution on the real system. This has allowed us to avoid the execution of potentially unsafe configurations that emerged from tests in simulation (i.e., starting from positions close to the attached area). All these factors make the proposed simulation environment an essential component in development of autonomous agents for controlling surgical tools and manipulating soft tissues.

Although we have considered a simplified anatomical setup, the modularity of the framework will allow to easily increase the complexity of the simulation, for example including multiple organs and both robotic arms, to enable learning and testing of multiple procedural steps. We expect that this wil not introduce a drop in computational performance, due to reliance of the framework on the optimized NVIDIA FleX plugin to render soft object deformations.

Considering the exploited learning algorithms, future work will focus on the incorporation of visual cues, for example learning the task starting from the endoscopic images directly. Doing this, we will implicitly account for the current deformed tissue configuration. Sim-to-real transfer is likely to be more challenging in this case, since it will be essential to cope with the unavoidable difference between simulated and real endoscopic images. A possible strategy to cope with this issue will be to rely on methods for image-to-image translation like [208, 209], which will allow to transfer the style of real images to simulated ones.

Contributions of this Chapter

The main contributions of this Chapter are the following:

1. We introduce *UnityFlexML*, an open-source modular framework that provides an interface among a realistic simulation environment supporting deformable objects, the surgical robotic system and learning-based methods. *UnityFlexML* is available at https://gitlab.com/altairLab/unityflexml.
2. We show that the proposed framework has the required features to allow learning a surgical task (i.e., tissue retraction) both using a standard DRL method and a strategy combining DRL with LfD.
3. We demonstrate that the learnt policy translates directly to the surgical robotic system thanks to the da Vinci Research Kit (dVRK), without further training.

Publications linked to this Chapter

The content of this Chapter has been presented in the following publications:

1. Tagliabue E, Pore A, Dall'Alba D, Magnabosco E, Piccinelli M, Fiorini P: *Soft Tissue Simulation Environment to Learn Manipulation Tasks in Autonomous Robotic Surgery*, IEEE/RSJ International Conference on Intelligent Robots and Systems (2020); 2020 Oct 25-29; Las Vegas (USA);

2. Pore A, Tagliabue E, Piccinelli M, Dall'Alba D, Casals A, Fiorini P: *Learning from Demonstrations for Autonomous Soft-tissue Retraction*, 2021 International Symposium on Medical Robotics (2021); 2021 Nov 17-19; Atlanta (USA).

Chapter 6

A deliberative framework for autonomous robotic surgery

6.1 Introduction

One of the limiting factors for extensive development of autonomous robotic systems is the lack of an established theoretical framework that provides the required features to handle the different aspects and situations that an autonomous system has to face. This is due to the fact that identifying and implementing the required properties for autonomy is extremely challenging, especially when dealing with safety-critical scenarios as surgery, where errors can be deadly. In addition to being a high-risk scenario, surgery is characterized by a particularly complex environment consisting of deformable tissues, whose mechanical properties are only partially known in the pre-operative phase and challenging to measure and model [9]. The diversity and uncertainty of the environments that an ARSS has to face makes it impossible to foresee all possible courses of actions at design stage. For this reason, existing attempts to autonomous execution of surgical tasks have been demonstrated only on structured and controlled environments [7], making them still quite far from the applicability to real clinical scenarios.

To make a step forward towards autonomy in real surgical environments, in this Chapter we present a framework for autonomous robotic surgery that integrates the main properties that an ARSS should have. To cope with real anatomical environments, we have identified the following specific skills that an ARSS shall integrate:

- it should have *deliberative* capabilities [210], i.e. the capability of *monitoring* the environment through *sensors* while *acting*, in order to *reason* and *plan* a new strategy when the expected behavior does not match reality. Moreover, it must be able to adapt its prior model by *learning* from observations, to cope with the uncertainty of the environment and minimize critical events;

- it should be *informed* from pre-operative anatomical data and expert knowledge about task description and operative constraints, in order to devise a patient-specific strategy for the intervention;

- it should provide *interpretable* plans, which can be easily understood by a supervising

surgeon, to guarantee more reliability of the ARSS [211, 9, 212], as required from level 2 of autonomy [213].

In the design of a framework for autonomous surgery, the presence of a patient-specific simulation is of paramount importance for two reasons. First of all, a simulated environment initialized with a Patient-specific Biomechanical Model (PBM) allows to generate a plan of the intervention in the *pre-operative phase*, accounting for the specific needs of each subject and his/her clinical situation. Secondly, a simulation environment running in parallel to the actual execution in the *intra-operative phase* allows to both detect possible unexpected behavior via comparison with real sensor data, and provide additional critical information to the system, which cannot be obtained from the available sensors (e.g., an estimate of the interaction forces). Therefore, integration of a physics-based simulation into a framework for autonomous surgery seems a fundamental choice to enhance the functionalities of the ARSS.

In this Chapter, we introduce a DEliberative Framework for Robot-Assisted Surgery (DEFRAS) that integrates all the identified skills necessary to cope with uncertain anatomical environments. The framework has a modular structure, thus offering the opportunity to implement each deliberative function individually, specializing each module at algorithmic level depending on the desired task. After introducing DEFRAS modules and functionalities in general terms, we provide a more technical description of the specific functionalities implemented by each module to accomplish the Tissue Retraction (TR) task, the one considered for experimental validation of the framework. Finally, we demonstrate that DEFRAS is able to cope with deformable environments with uncertain parameters, accomplishing TR both in simulated and real experiments with the da Vinci Research Kit (dVRK).

6.2 Dealing with uncertainty in autonomous robotic surgery

Before providing the details of the proposed framework, in this Section we review recent frameworks for surgical robotic autonomy under uncertainty clarifying our contribution, and we motivate the use of deliberation and logic.

Uncertainty poses challenges to ARSSs relying only on prior models and/or task description [214], and can be related to multiple aspects of a surgical procedure. *Localization uncertainty* includes imprecise localization of the structures of interest or the instruments, due to limitations of available sensors, kinematic models or situation awareness algorithms, and can be addressed with improved mechanics [215] or sensors [216]. *Environmental uncertainty* encompasses the partial knowledge of the anatomical behavior and its unpredictability, which can be tackled with strategies for online model update [156, 217, 218]. *Procedural uncertainty* involves understanding of the current and next state in the surgical workflow, which can be improved through filtering of the predictions of the next action [219]. These approaches aim at reducing uncertainty, rather than integrating it in the planning process, so criticisms in real executions might still arise. Some works have tried to deal with uncertainty directly at tool motion planning level, exploiting model predictive control [219, 220]. However, also task-level uncertainty (e.g., wrong plans rising from poor awareness) shall be taken into account. To this aim, full motion parameters can be learnt, either with hidden Markov models [221] or with supervised refinement of finite state machines [181]. However, esti-

mation of task-level parameters with offline learning methods fails to generalize to possible unexperienced situations.

While existing works have attempted either to reduce uncertainty at sensing level or to deal with it at motion/task level, in this Chapter we address both aspects by introducing deliberative functions of autonomous agents, as described in [210]. In particular, we integrate specific modules for uncertainty *monitoring* and pre-operative model refinement through online *learning* within a framework for robot-assisted surgery that guarantees interpretability at task level with logic programming and supports online re-planning. Several paradigms for deliberation have been proposed in the field of artificial intelligence, focusing on different aspects of autonomy, e.g. planning (Belief-Desire-Intention model [222], and its implementations [223, 224]) and learning [225]. However, in order to guarantee adaptability to diverse complex tasks, we here focus on the integration and parallel development of different skills for autonomy, which can be improved individually depending on the task of interest. For this reason, our framework builds on the modular architecture and definitions of deliberative functions described in [210], which reports a large number of robotic applications.

In addition to deliberation, an ARSS must provide an interpretable behavior for a human supervisor [211, 212]. Interpretability for autonomous robots can be guaranteed by logic-based reasoning systems [226]. They encode prior concepts and specifications about a robotic domain in a knowledge base, using logic formalism which is easily readable by humans. A logic-based task planner does not encode pre-defined workflows (as finite state machines do); instead, it performs human-like logic inference from knowledge and data from sensors, in order to generate an appropriate workflow for the current situation. Hence, the logic-based approach formally guarantees that specifications encoded in prior knowledge (which may be, e.g., safety constraints in surgery) are satisfied during the execution. In the surgical context, logic formalization has been often implemented with ontologies [227, 228], which are computationally inefficient when knowledge update is needed, as required when operating in unstructured environments [229]. We propose to use non-monotonic logic programming [230], which provides an efficient and interpretable way to describe tasks, permits online revision of incomplete or dynamic information and allows to define safety constraints for planning the execution [231]. All these features are essential in safety-critical scenarios as surgery [232]. Non-monotonic logic programming has been applied in surgery only to automate a standard surgical training task in rigid domain [231].

6.3 DEFRAS: DEliberative Framework for Robot Assisted Surgery

In this Section, we present the modules of our DEliberative Framework for Robot Assisted Surgery (DEFRAS). The framework has been made publicly available at https://gitlab.com/altairLab/tissue_retraction.git (Fig. 6.1).

Task planning A task planning module computes the task plan, i.e. the sequence of actions to solve the task. Actions are elementary operations which correspond to a single motion trajectory or joint-level command, according to the granularity definitions described in [174]. The plan is computed based on available expert procedural knowledge, conveniently ex-

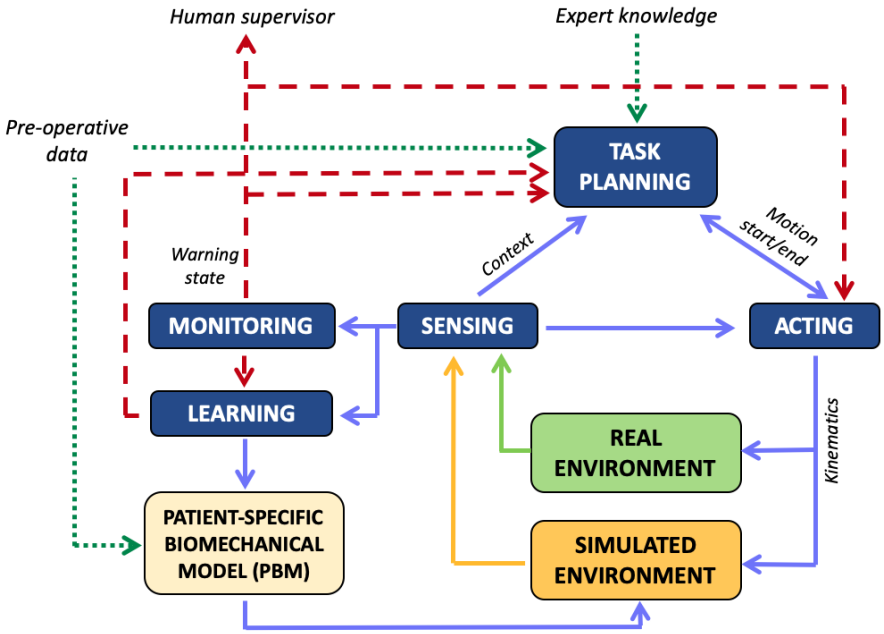


Figure 6.1: The architecture of DEFRAS. Deliberative modules are in blue boxes. Green dotted lines represent information exchanged only in the pre-operative phase. Red dashed lines trigger actions only when monitoring threshold/s is exceeded.

pressed in the declarative logic programming formalism of Answer Set Programming (ASP) [233], a non-monotonic logic paradigm which is more computationally efficient and expressive if compared to its main competitor Prolog [234].

An ASP program represents the domain of the task with a *description* $\mathcal{D}$ and a *history* $\mathcal{H}$. $\mathcal{D}$ comprises the alphabet of the task $\mathcal{A}$, defining relevant attributes of the domain, and axioms. Attributes may be *statics*, i.e., domain attributes whose values do not change over time; *fluents*, i.e., domain attributes whose values can be changed; and *actions*. Attributes may be *terms*; *atoms*, i.e. predicates of terms (e.g. $\texttt{Atom(T}_1\texttt{, ..., T}_n\texttt{)}$ is an atom with terms $\texttt{T}_{1,\dots,n}$ as arguments); and their logical negations. Values of terms are *constants* (either integers or strings). A term whose value is assigned is *ground*, and an atom is ground if its terms are ground. Axioms are logical relations between attributes. Causal rules $\texttt{A}_h$ $\texttt{:-}$ $\texttt{A}_{b1}\texttt{, ..., A}_{bn}$ define pre-conditions $\texttt{A}_{b1,\dots,bn}$ (*body* of the rule) for grounding the *head* $\texttt{A}_h$. Logical implications are implemented to define pre-conditions and effects of actions, and constraints (e.g., infeasibility of some actions).

Given the ASP program, an ASP solver returns both the plan and the set of anatomical features which generated it, thus ensuring task-level interpretability of the ARSS.

Sensing Throughout task execution, the sensing module analyses the state of the robot and the environment, to provide high-level environmental features for task planning. It also communicates target position for each action to the acting module, for motion planning. Moreover, it recognizes whether critical conditions occur, in which cases it triggers task re-planning or motion interruption to prevent risks.

Acting The acting module controls the execution of each action in the plan at low level. Different motion planning algorithms can be used, depending on the application. The appropriate motion law for each action can be selected from a pre-defined set, as obtained with analytical approaches such as roadmaps, potential fields or cell decomposition [235]. These methods work well when complete representation of the environment is available, as can be extracted, for example, from pre-operative data. However, complex surgical tasks, like tissue suturing or cutting, require enhanced dexterity and bi-manual coordination which might be impossible to encode within hand-crafted control policies. In this cases, trajectories may be generated by imitating expert human demonstrations [181, 182, 231]. Alternative approaches to motion planning can also be integrated, such as those relying on reinforcement learning, which has shown promise to solve challenging robotic planning tasks by trial-and-error, especially when combined with deep neural networks [184], or emerging approaches based on neural planning, e.g. [236].

Monitoring The monitoring module continuously estimates if the system internal model of the environment, given by a patient-specific simulation initialized with pre-operative information and running in parallel, is coherently representing the actual scenario. To this aim, a metric assessing the level of discrepancy between the real and the simulated environments is computed starting from sensing information. This metric represents the current level of uncertainty that the system has about the environment. Depending on this metric, different warning states are raised, associated with specific strategies at task-motion level which aim at increasing the reliability of the autonomous operation and accommodate safety.

Learning If the system is in a warning state, the learning module is in charge of updating the ARSS internal model of the environment starting from real observations. Being the system model represented by the simulated PBM, the learning module implements strategies to update PBM parameters from real data, to compensate for environmental uncertainty caused by imprecise parametrization.

Simulated environment A PBM of the anatomical environment is created starting from patient-specific geometry and mechanical properties extracted from pre-operative data. Such model initializes a simulated environment which is exploited to design a patient-specific task plan accounting for all the information available in the pre-operative phase. Furthermore, the simulation is run in parallel to the real execution in the intra-operative phase and it is continuously updated based on the current robot configuration received from the acting module. At each time step, the simulation represents what the system expects from its interaction with the environment based on the available knowledge, hence it allows to monitor the real world execution (Fig. 6.1). Moreover, the online simulation can provide complementary information to real sensor data.

Human supervisor The context and the plan are continuously monitored by a human supervisor, who can easily interpret the output of the autonomous system thanks to reliance on logic formalism. The presence of the human supervisor is required to guarantee the reliability of the ARSS, following the latest European regulations for high-risk autonomous systems [212].

6.4 Implementation of tissue retraction in DEFRAS

In this Section, we describe the Tissue Retraction (TR) task, which has been selected as representative task to validate DEFRAS, and we detail its implementation within our framework. The TR task allows to assess the ability of the framework to deal with uncertainty, since it involves interaction with the soft anatomical tissues, whose mechanical properties are generally only approximately known before the intervention. While an initial PBM can be generated from each patient pre-operative images and data, it may not accurately describe the actual behavior of the tissue, thus requiring intra-operative update.

Previous works for autonomous TR have proposed either to generate the plan by optimizing some metrics [177, 178, 176, 237, 238] or to learn the task from human demonstrations [189, 190]. However, none of them integrates methods to reason on the current environment and deal with unexpected situations which might occur in real surgery.

6.4.1 Tissue retraction task

Tissue retraction is a very common surgical task which consists in grasping and retracting soft tissues (e.g., adipose tissue) to expose a hidden Region Of Interest (ROI) (e.g., a tumor) to an intra-operative camera. The tissue is usually attached to surrounding anatomies in correspondence of Attachment Points (APs). In general, ROI exposure during TR is obtained primarily by pulling up the tissue after grasping. However, this action is not always successful due to kinematic limitations imposed by the anatomical environment and the insertion points of the instruments in the body, which define fixed pivots for the robotic arms. Furthermore, pulling may induce high forces which can damage the tissue. In these cases, alternative actions can be performed, such as moving the tissue in a parallel plane to the tissue surface, either folding it away from the intra-operative camera or possibly towards the ROI.

We create a PBM leveraging on tissue geometric and mechanical properties as well as locations of the ROI and APs obtained from pre-operative data (Fig. 6.1). Such model is used to initialize a simulation which is exploited to find a patient-specific task plan in the pre-operative phase (see Section 6.4.2 for details). For computational convenience, we subdivide the tissue in small regions, and consider only centers of these regions as possible grasping points. The pre-operative plan is then executed in the real environment with one of the Patient-Side Manipulators (PSMs) of the dVRK.

During execution, the biomechanical simulation is run in parallel, and the state of the real environment is continuously monitored with intra-operative sensors (Fig. 6.1). When mismatch is identified, the pre-operative plan may become sub-optimal or even infeasible, hence specific strategies to improve the plan shall be implemented and some simulation parameters may be updated. In particular, here we focus on the update of APs. Although the position of APs can be initialized based on anatomical knowledge or statistical information [50, 159], precise definition of APs in surgery can only be obtained interacting with the tissue, as we demonstrated in Chapter 4. The reason why we focus on APs is twofold. First of all, they constitute model boundary conditions, thus represent an important parameter influencing simulation accuracy. Second, knowledge of APs location is very important in TR to prevent the pulling of highly constrained areas, minimizing the risk of tissue damage.

In addition, the simulation also continuously provides an estimate of the applied force on

the tissue, compensating for the lack of force feedback in surgical robotic systems.

6.4.2 Task planning

The task planning module computes the best sequence of actions to solve the task, and it encodes task knowledge in the formalism of ASP [239]. Definitions of actions and relevant environmental features are encoded following the generic explanation of Section 6.4.1. Statics are `block` (from now on `B`), representing a generic region on the tissue with centroid $\mathtt{B}_C$; and `arm` (from now on `A`), representing the robotic arm (either `psm1` or `psm2`).

Possible actions (for each tool) in the considered task are *reaching a grasping point, opening gripper, grasping, pulling, releasing the tissue* and folding the tissue, either *moving planarly towards the ROI* or *moving away from the intra-operative camera*. Actions are expressed as predicates in plain English as *action(tool, object, property)*.

Fluents are relevant environmental conditions for the task. They include *locations of APs and ROI* with respect to grasping points; critical conditions on the *force exerted on the tissue*; kinematic states of the robot, e.g. whether it is *holding tissue* or the *gripper is open*, or condition of *maximum height* reached above the tissue; *reachability*[1] conditions to define which points can be grasped by which robotic arm; *distance* between blocks. All fluents are defined as *external* atoms, so they can be grounded from the sensing module. Table 6.1 reports actions and fluents used in the TR task, expressed as atoms with the ASP formalism.

Table 6.1: List of actions and fluents implemented for the TR task with the ASP formalism. $\mathtt{B}_C$ is a generic grasping point, `A` is the robotic arm, `t` is a discrete time step for temporal reasoning.

Actions `a`	**Description**
`reach(A,` $\mathtt{B}_C$`, t)`	`A` reaches grasping point $\mathtt{B}_C$
`grasp(A,` $\mathtt{B}_C$`, t)`	`A` grasps the tissue in $\mathtt{B}_C$
`release(A, t)`	`A` opens the gripper
`pull(A,` $\mathtt{B}_C$`, t)`	`A` pulls $\mathtt{B}_C$
`move_ROI(A,` $\mathtt{B}_C$`, t)`	`A` moves towards the ROI while holding $\mathtt{B}_C$
`move_away(A,` $\mathtt{B}_C$`, t)`	`A` moves away from the camera while holding $\mathtt{B}_C$
Fluents F	**Description**
`fixed(B, t)`	$\mathtt{B}_C$ is above APs
`above_ROI(B, t)`	$\mathtt{B}_C$ is above ROI
`at(A, B, t)`	`A` is above $\mathtt{B}_C$ on tissue
`max_force(t)`	critical force is overcome
`in_hand(A, B, t)`	`A` holds the tissue in $\mathtt{B}_C$
`closed_gripper(A, t)`	gripper of `A` is closed
`max_height(A, t)`	`A` has reached the maximum allowed height
`visible_ROI(t)`	ROI is visible
`reachable(A, B, t)`	$\mathtt{B}_C$ is reachable by `A`
`distance(B1, B2, X, t)`	`X` is the distance between $\mathtt{B}_1$-$\mathtt{B}_2$

Axioms express logical relations between atoms. For the TR task, we consider axioms representing causal laws, executability constraints, choice rules and optimization statements.

[1] With reference to Fig. 6.2a, a point on the tissue is reachable by a PSM if it lies on the same side as the PSM with respect to the $x - z$ plane in the common reference system.

Causal laws define effects of actions on fluents (e.g., A's gripper is closed after grasping, `closed_gripper(A, t) :- grasp(A, B_C, t-1)`).

Executability constraints specify the set of actions and fluents that cannot hold concurrently (e.g., it is not possible to pull the tissue if the force applied to the tissue exceeds constraints, `:- pull(A, B_C, t), max_height(A, t)`).

Choice rules specify the environmental pre-conditions which must be satisfied to execute actions. For our task, at most one single action can be executed at each time step. Then, we define a choice rule gathering pre-conditions for actions as `0 {action(t) : pre-cond(t)} 1`, which constrains the cardinality of the set of possible actions returned at each time step between 0 and 1.

Optimization statements define preference in the set of possible actions, in particular for grasping point selection in case the pre-operative plan is not successful and re-planning is needed. In this case, the best grasping point B_C is chosen as the one which has minimal distance from ROI (to ease its exposure) and maximal distance from APs (to reduce tissue damage). Logic-level optimization allows to easily define a commonsense criterion for evaluating candidate grasping points. However, in order to ensure computational tractability of ASP solving, the set of candidate grasping points must be finite. Hence, the tissue surface is discretized as an $N \times N$ grid, centered above the tissue to grasp. Possible grasping points B_C are defined at the centers of the grid cells.

We finally define the goal of the TR task as the constraint that the ROI must be eventually visible within a predefined percentage (`:- visible_ROI(t)`).

Pre-operative plan In the pre-operative phase, all the candidate grasping points are tried in simulated environment to find the optimal patient-specific plan. Considering the goals of our task, we select the plan that maximizes ROI exposure while minimizing tissue exerted force and interaction with it, in order to maximize the economy of motion and reduce the possibility of critical events. To achieve this, DEFRAS is launched after deactivating the monitoring and learning modules. We record plans originated picking all possible grasping points, and we select only plans reaching a minimum visibility of 80%. Among these, we select the 3 plans minimizing the tissue solicitation, i.e. the ones with lowest F_{max}, where F_{max} is the maximum recorded tissue force over the entire execution. Finally, we choose the optimal plan as the one with the lowest number of actions, thus minimizing interaction with tissue and obtaining the maximum economy of motion.

In case re-planning is needed in the intra-operative phase, a new grasping point is then selected according to the ASP optimization criteria described above.

6.4.3 Sensing

The sensing module continuously receives the state of the robot and the environment. The state of the robot S_R consists of the position of tools $p_{1,2}$ and opening angles of grippers $j_{1,2}$. The state of the environment S_E includes the point cloud of tissue surface PC_t from the real environment; visual information about the ROI from the intra-operative camera I; the position of the ROI p_{ROI}; the set of known APs PC_{AP}; and the set Σ of mechanical forces at each tissue point from the simulated environment. All the kinematic quantities are referred to the same reference frame after camera-robot calibration. The main tasks of the sensing modules are summarized in Alg. 3.

Algorithm 3 Sensing module

1: **Input**: S_E, S_R, current action `a`
2: **Output**: Set of context fluents F, target position g, motion rate r, replanning request `replan`, motion rate r
3: **while** not `task_ended` **do**
4: F = compute_fluents(S_E, S_R)
5: `replan`, r = check_failure(F)
6: g = compute_target(`a`)

Throughout task execution, the sensing module computes the environmental features described at the logic planning level as detailed in Alg. 4[2] and recognizes whether critical conditions occur (e.g., maximum height of PSMs or critical tissue force are reached). In these cases, the motion of PSMs is interrupted, and the current environmental situation is communicated to the planning module to adapt the flow of execution accordingly. In particular, the function $check\_failure$ verifies if the maximum arm height has been reached, or the maximum tissue force has exceeded a pre-defined threshold ϵ. In the former case, motion is interrupted and re-planning is triggered. In the latter case, DEFRAS first tries to reduce tissue solicitation by decreasing motion velocity [240]. If this does not bring the applied force below ϵ, motion is interrupted and re-planning is triggered.

Another important function of the sensing module is to compute target positions g for actions of PSMs, with the function $compute\_target$ in Alg. 3. The target is needed only for reaching the tissue, moving away from camera and moving towards ROI.

6.4.4 Acting

The acting module selects the appropriate motion law for each action from a pre-defined set, similarly to [231]. Some actions (*reaching tissue, moving away from camera* and *moving towards ROI*) are executed after also the target position g is received from the sensing module. Gripper actions are simple joint commands, while *pulling* is a motion of $10\,$mm along z axis. We consider only linear trajectories, keeping a fixed orientation at the end-effector for the sake of simplicity, as in [192]. This does not represent a limitation for our initial evaluation on a controlled setup. However, more sophisticated motion laws can be employed (Section 6.3). The acting module executes trajectories at the rate r specified by the sensing (Section 6.4.3) and monitoring modules (Section 6.4.5).

6.4.5 Monitoring

The monitoring module continuously receives PC_t and the point cloud of the simulated tissue PC_s from the sensing module, and computes a discrepancy metric μ, chosen as the median Euclidean distance of closest points between PC_t and PC_s. The distance is normalized with respect to its rest value (i.e., before grasping occurs) to compensate for offsets induced by registration error and sensor noise. Inspired from recent works as [241], the monitoring module considers 3 thresholds $\delta_1, \delta_2, \delta_3$ (empirically set to 2, 4 and $6\,$mm) to classify the warning state

[2]Time `t` is omitted in Alg. 4 because it is assigned at ASP level.

of the system (Alg. 5). In particular, when $\mu > \delta_1$ (Line 10) the motion velocity is reduced for caution, to minimize the risk of applying excessive forces [240] and to allow prompt capturing of eventual higher levels of discrepancy. If δ_2 is exceeded (Line 12), motion is interrupted and re-planning is asked at task planning level, meaning that probably the environmental conditions have significantly changed. In both these warning states, the learning module is triggered to update environmental knowledge exploiting data from sensors. Overcoming δ_3 (Line 14) results in interruption of DEFRAS and human intervention is required because either the current situation is too anomalous or the environmental model is inadequate.

Algorithm 4 compute_fluents

1: **Input**: $S_E = \{PC_t, PC_{AP}, p_{ROI}, I, \Sigma\}$, $S_R = \{j_{1,2}, p_{1,2}\}$
2: **Output**: Set of context fluents F
3: **Init**: $F = \varnothing$, S_B = discretize(PC_t) ▷ *S_B is the set of regions B on the tissue*
4: **for** B $\in S_B$ **do**
5: $j = argmin_{i=\{1,2\}}||y_i - y_{B_C}||_1$ ▷ *Reachability with respect to $x-z$ plane*
6: F.append(reachable(A$_j$, B))
7: **if** $\exists p \in PC_{AP} : \ p \in$ B **then** ▷ *B is a fixed block*
8: F.append(fixed(B))
9: **if** $p_{ROI} \in$ B **then** ▷ *ROI under block B*
10: F.append(above_ROI(B))
11: **for** B$_2 \in S_B$ **do** ▷ *Distance between centroids of blocks*
12: F.append(distance(B, B$_2$, X))
13: **if** compute_visibility(I) $>$ 80% **then** ▷ *ROI visibility*
14: F.append(visible_ROI)
15: **for** $i \in \{1, 2\}$ **do**
16: **if** $j_i <20^\circ$ **then** ▷ *Gripper state*
17: F.append(closed_gripper(A$_i$))
18: **for** B $\in S_B$ **do**
19: **if** $p_i \in$ B **then** ▷ *Arm location*
20: F.append(at(A$_i$, B))
21: **if** $j_i <20^\circ$ **then** ▷ *Tissue grasped*
22: F.append(in_hand(A$_i$, B))
23: **if** $p_i[2] >50\,\mathrm{mm}$ **then** ▷ *Max height reached*
24: F.append(max_height(A$_i$))
25: **if** $max\Sigma > \epsilon$ **then** ▷ *Max force reached*
26: F.append(max_force(A$_i$))
return F

6.4.6 Learning

As motivated in Section 6.4.1, in this work we focus on the intra-operative update of APs whenever required by the monitoring module. Update relies on BA-Net, the deep neural network which predicts APs starting from the tissue pre-operative geometry and the current

Algorithm 5 Monitoring module

Input: PC_t, PC_s, motion rate r, `grasped_tissue`
Output: Warning state s, motion rate r, trigger learning `learn`
Init: $d_{rest} = []$
while not `task_ended` **do**
    while not `grasped_tissue` **do**
        d_{rest}.append($||PC_t - PC_s||_2$)
    d_{rest} =mean(d_{rest})
    $d = ||PC_t - PC_s||_2$
    $\mu = d - d_{rest}$
    if $\delta_1 < \mu \leqslant \delta_2$ **then**
        return $s = 1; r = \frac{r}{2}$; `learn=True`
    else if $\delta_2 < \mu \leqslant \delta_3$ **then**
        return $s = 2; r = 0$; `learn=True`
    else if $\mu > \delta_3$ **then**
        return $s = 3; r = 0$; `learn=False`
    else
        return $s = 0; r = r$; `learn=False`

observed PC_t, presented in Chapter 4. This method has proved suitable for intra-operative model refinement, requiring a very low inference time and even improving biomechanical model accuracy. To improve the robustness to sensor noise, we consider the intersection among the latest 30 predictions provided by BA-Net as new estimate of APs. The new estimate of APs is used on the one hand by the task planning module to update its knowledge of the regions to avoid, in case of grasp re-planning. On the other hand, it is exploited to refine the PBM which drives the simulated environment, thus improving its overall reliability (in particular of the force estimate).

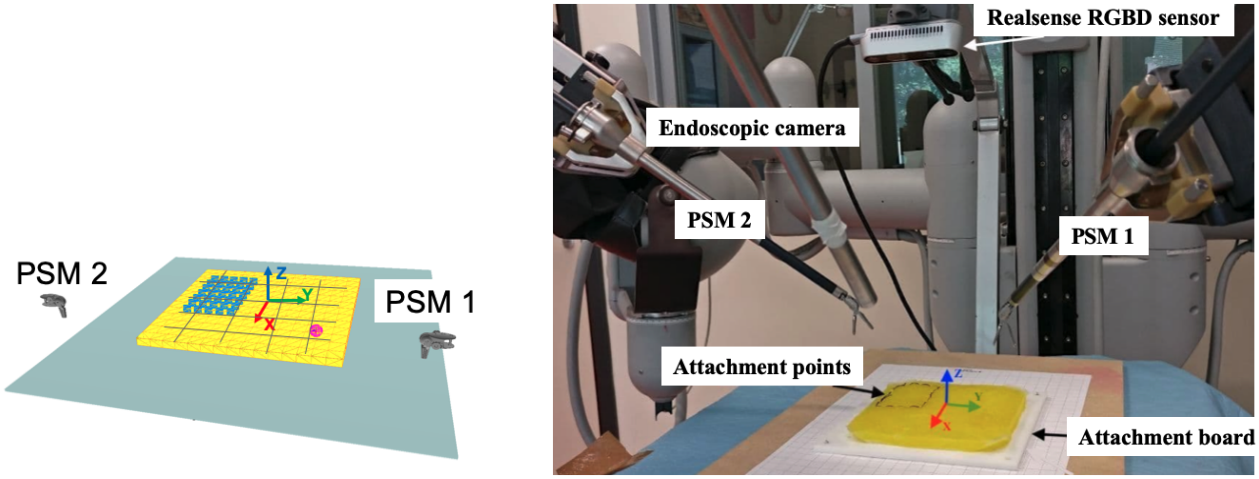
6.4.7 Simulated environment

To obtain a physically accurate result, we model the deformable behavior of anatomical tissues exploiting continuum mechanics laws, solved with the finite element (FE) method (see Section 2.3.1). The choice of a physics-based method allows to continuously estimate forces exerted on the tissue, compensating for the lack of force feedback in surgical robotic systems. Monitoring of the applied forces is crucial in our task to avoid unnecessary tissue damage. The deformable tissue is modelled as a linear elastic material with Young's modulus $3\,$kPa and Poisson ratio 0.45, a common modelling choice for adipose tissues [164], which are often responsible of hiding structures of interest during real surgery [242]. In order to handle large deformations while guaranteeing the computational efficiency required to run the simulation in parallel to the real execution, we rely on the corotational formulation of linear elasticity [48] provided by the SOFA framework [148]. In the considered simulation environment, grasping is implemented by constraining the tissue points closest to the simulated PSM (distance below a small threshold, $5\,$mm in our task) to follow PSM motion, similarly to [177]. Grasping is activated or released based on commands received from the acting

module. Current implementation does not consider the possibility of slippage or breaking contacts between the tool and the tissue, that might happen during real retraction.

6.5 Autonomous tissue retraction with DEFRAS

Experiments are conducted both in simulated and real environments to assess the capability of DEFRAS to successfully accomplish the TR task, coping with inaccurate parametrization of the available model of the environment from pre-operative information.



a b

Figure 6.2: Experimental setup for autonomous tissue retraction with DEFRAS. The simulated and real scene are calibrated with respect to a common reference frame at the tissue center. (a) Simulated setup. The magenta sphere represents the ROI, while blue dots are the attachment points. Gray lines delineate the discretization for selection of grasping points. (b) Real setup. The attachment board allows to define tissue attachment points. A Realsense RGBD sensor captures the point cloud of the tissue throughout the task.

6.5.1 Experimental setup

We consider the problem of retracting a thin layer of soft tissue ($120 \times 120 \times 5$ mm) to expose a ROI to the endoscopic camera, which is placed at the main surgical viewpoint (Fig. 6.2). The task is executed on the dVRK, whose PSMs are equipped with ProGrasp® surgical grasping tools (Fig. 6.2b). The ROI is considered exposed if at least 80% of its surface is visible from the endoscope. For monitoring purposes, we continuously capture the tissue point cloud PC_t with an additional Intel® Realsense D435 RGBD sensor, placed opposite to the endoscope to maximize tissue visibility during manipulation, as done for the experiments presented in Sections 4.4 and 4.5. The RGBD sensor, endoscopic camera and the PSMs are calibrated with respect to a common reference frame placed at the center of the tissue (Fig. 6.2), using the methodology described in [169]. The simulated environment is aligned to the same reference frame and a simulated RGBD sensor, placed in the same position as the real one, provides the point cloud of the simulated tissue PC_s.

FRAS: Framework for Robot-Assisted Surgery

At this point, it is worth introducing a base version of the framework, called FRAS, where:

- grasping point is always chosen following the heuristic used by DEFRAS in case of intra-operative re-planning, as described in Section 6.4.2. This means that FRAS does not rely on any pre-operative plan;
- neither the monitoring or the learning modules are activated.

FRAS is used to assess the values of some metrics that are independent from the monitoring and learning modules. For example, FRAS is used to identify a reasonable value for the force threshold ϵ (see Section 6.4.3). We consider 17 pre-operative configurations of tissue, each one with different locations for APs and ROI. For each pre-operative configuration we launch FRAS only in simulation, ignoring also force measurements. We compute plans for all possible grasping points under these conditions, and we record the estimated forces from the simulation. We obtain a median force $F_m = 0.15\,\mathrm{N}$ among all 425 executions. F_m is chosen to represent the typical force on the tissue in the considered setup, hence we set $\epsilon = 2F_m = 0.3\,\mathrm{N}$.

Furthermore, FRAS is used as a comparison to assess the role of monitoring and learning modules, as well as the utility of the pre-operative plan.

6.5.2 Validation of DEFRAS in simulation

The capability of DEFRAS to cope with uncertain environmental parameters has been initially assessed in simulation. Experiments in simulation allow to evaluate the performance of the framework excluding possible inaccuracies arising from the presence of real sensor noise. We design 8 pre-operative scenarios, each characterized by a different configuration of APs (red outlines in Fig. 6.3) extracted as random surface patches following the same approach presented in Section 4.2.1. For each pre-operative scenario, we generate optimal plans following the strategy described in Section 6.4.2.

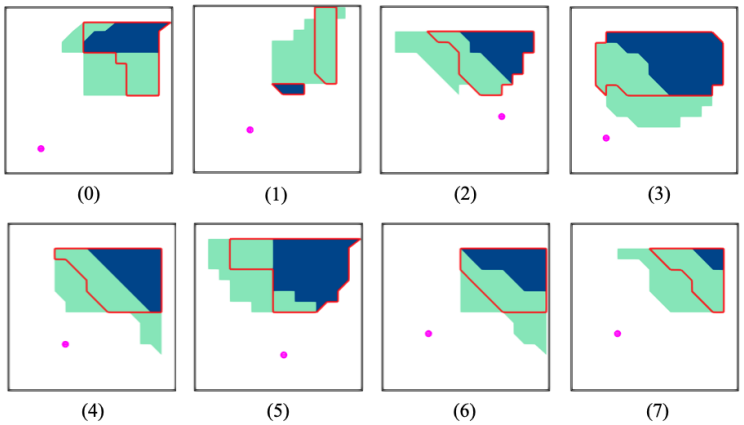
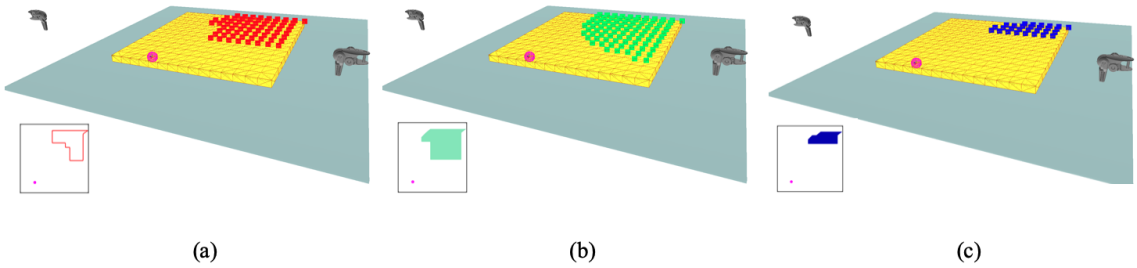


Figure 6.3: Configurations of APs used for the experiments in simulation. The red outline delimits the pre-operative configuration used to generate the plan; considered actual simulations are in blue (⊖) and green (⊕); magenta sphere represents the ROI.



(a) (b) (c)

Figure 6.4: Simulation scenes relative to the configurations of APs for case 0 in Fig. 6.3. (a) The pre-operative configuration; (b) The $\oplus$ actual configuration; (c) The $\ominus$ configuration.

Each pre-operative plan is executed by DEFRAS on two validation scenarios: the former with fewer APs (in our case, between 20-60%, indicated with symbol $\ominus$) and the latter with more APs (in our case, between 140-160%, indicated with $\oplus$) with respect to the corresponding pre-operative configuration. These experiments emulate the realistic situation where initial knowledge of APs is not precise, thus the pre-operative plan might need adjustment due to mismatch between the current and the simulated environments. Fig. 6.4 provides a close-up on the configurations of APs considered for case 0 in Fig. 6.3. To run these experiments, we rely on two parallel simulations: the former (*pre-operative simulation*) represents the simulated environment in the scheme of Fig. 6.1, estimating the tissue forces and initialized with pre-operative APs (red outlines in Fig. 6.3); the latter (*actual simulation*) represents the validation scenario with either fewer or more APs than the pre-operative simulation, with simulated dVRK and cameras (blue and green areas in Fig. 6.3). Visibility in the actual simulation is computed by counting the number of visible ROI points from the simulated endoscope, assessed via ray-casting technique. The monitoring module continuously compares the tissue point clouds from the two simulations, triggering specific interventions according to Alg. 5. To assess the performance of the monitoring module, we compute the success rate of APs update with BA-Net as ratio between the number of times the update results in $\mu \leqslant \delta_1$ in Alg. 5 (no warning from monitoring) and the number of times BA-Net is triggered because $\mu > \delta_1$ (warning state).

Furthermore, we compare the performance of DEFRAS with respect to FRAS, in order to show the advantages of the pre-operative plan and of the monitoring and learning modules in a deliberative architecture for autonomous robotic surgery. Therefore, we let FRAS perform autonomous TR on the actual simulations.

Re-planning efficiency In addition to validating DEFRAS capability to handle uncertain environmental conditions, we evaluate the computational performance of task reasoner as the *re-planning time*, defined as the time required to compute a new plan given a context interpretation from the sensing module. In particular, we assess how the re-planning time is impacted by the resolution of the grid that defines candidate grasping points (see Section 6.4.2), while also varying the percentage of APs. In fact, both the number of possible grasping points and the percentage $f_{\%}$ of APs influence the size of the search space.

To this purpose, we run 100 simulations initialized with different random environmental states (i.e., p_{ROI} and PC_{AP}) and we execute the task with FRAS, i.e. relying on the described

strategy for grasping point selection in case of re-planning and without the monitoring and learning modules. In this evaluation, we consider different resolutions of the grid defining candidate grasping points $N \in \{5, ..., 10\}$. For each value of N, we run 25 simulations for each value $f_{\%} \in \{10\%, 30\%, 50\%, 70\%\}$. This results in 600 total simulations, each initialized with a different random p_{ROI} and PC_AP (according to the value of $f_{\%}$).

6.5.3 Validation of DEFRAS on the dVRK

The performance of DEFRAS is evaluated on autonomous TR of silicone tissue with the real dVRK (Fig. 6.2b). Following the same approach of Section 4.3.1, APs in the real environment are defined by stitching the tissue on an attachment board with regularly spaced holes every $10\,$mm. The point cloud of the tissue in real environment is acquired after color-based segmentation and decimation of the raw point cloud from Realsense with the same pre-processing pipeline presented in Section 4.2, and compared with the point cloud from the simulated environment for monitoring. Visibility in the real environment is computed detecting the ROI (with known size and position) from the endoscopic camera images with color-based segmentation and considering the percentage of visible pixels.

We consider three real world scenarios (A, B and C), obtained by stitching the tissue to the attachment board with 3 different configurations of APs (light blue in Fig. 6.5). For each scenario, we consider 3 pre-operative models, each characterized by a different pre-operative knowledge of APs: (1) fewer APs (between 20-60%, indicated with $\ominus$), (2) more APs (between 140-160%, indicated with $\oplus$), and (3) same APs (indicated with $\circleddash$) than/as the real scenario (Fig. 6.5). For each of them, we generate the pre-operative plan in simulation, and test DEFRAS on the corresponding real configuration with the dVRK.

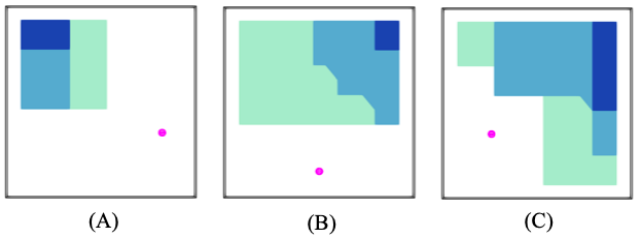


Figure 6.5: Configurations of APs used for the real world experiments. The light blue region in each subimage represents the configuration of APs in the real setup ($\circleddash$). Blue and green regions indicate pre-operative configurations with fewer APs ($\ominus$) and more APs ($\oplus$) than the real configuration. Magenta sphere represents the ROI.

Moreover, we analyze the performance of our framework with respect to two state-of-the-art implementations of autonomous TR on real setups. In particular, we compare both to the results obtained with the learning-based approaches presented in previous Section 5.6.1 and to the work by Attanasio et al. [176], which relies on image-guided control to reveal a ROI behind tissue flap. Both the considered approaches evaluate task success as percentage of ROI exposure upon task completion.

6.6 Experimental results of autonomous TR in simulation

Table 6.2 reports the details of the optimal plans generated for each pre-operative configuration in the experiments in simulation (red outlines in Fig. 6.3). Each of these plans is executed with DEFRAS on a scenario with fewer and more APs than those of the corresponding pre-operative configuration. Details of such executions are presented in the upper part of Table 6.3. The lower part of Table 6.3 reports the results obtained when executing the TR task on the same scenarios using FRAS.

Re-planning times at varying grid resolutions and percentage of grasping points are reported in Table 6.4.

Table 6.2: Pre-operative plans for the 8 scenarios in our simulation experiments. Table reports the final reached visibility, the number of actions, the number of times $F > \epsilon$ and F_{max}. Last two columns report median and interquartile (IQR) range per row.

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	Median	IQR
Visibility [%]	100	100	100	95	80	100	93	100	100	94-100
Actions	4	6	6	5	5	5	5	5	5	5-5
$F > \epsilon$	0	0	1	0	1	1	1	1	1	0-1
F_{max} [N]	0.3	0.3	0.4	0.3	0.4	0.3	0.6	0.5	0.4	0.3-0.4

Table 6.3: Simulation experiments. $\ominus$ indicates the scenario with fewer APs, while $\oplus$ the one with more APs than those of the pre-operative configuration. The upper Table reports results obtained with DEFRAS, while the lower Table reports results obtained with FRAS. “BA-Net" row reports the success rate of APs update with BA-Net.

DEFRAS

	0		**1**		**2**		**3**		**4**		**5**		**6**		**7**			
	$\ominus$	$\oplus$	$\ominus$	$\oplus$	$\ominus$	$\oplus$	$\ominus$	$\oplus$	$\ominus$	$\oplus$	$\ominus$	$\oplus$	$\ominus$	$\oplus$	$\ominus$	$\oplus$	Median	IQR
Visibility [%]	100	100	100	100	100	0	9	95	100	100	0	100	100	100	100	100	100	98-100
Actions	4	11	6	6	6	30	38	12	5	26	29	25	5	30	5	5	8	5-26
$F > \epsilon$	0	26	11	3	1	44	240	27	0	46	99	83	8	61	1	3	18	2-49
F_{max} [N]	0.2	1.3	0.4	0.4	0.4	1.4	0.9	1.8	0.3	1.7	3.3	3.3	0.4	1.2	0.3	0.4	0.6	0.4-1.5
BA-Net	1/1	1/1	-	0/1	2/2	1/1	6/6	-	1/1	2/2	11/12	-	1/1	4/4	2/2	-	-	-

FRAS

Visibility [%]	100	90	100	0	0	0	90	0	100	100	100	0	97	93	93	93	93	0-100
Actions	5	6	15	23	22	22	7	22	17	19	7	22	16	16	9	15	16	8-22
$F > \epsilon$	0	0	29	33	32	28	11	40	21	29	0	28	27	18	20	12	24	11-29
F_{max} [N]	0.3	0.3	1.0	1.0	1.1	1.0	0.4	1.2	0.9	0.9	0.3	1.1	0.9	0.9	1.2	0.9	0.9	0.8-1.0

Table 6.4: Re-planning time (average $\pm$ standard deviation) with different tissue grid size N and percentage of APs $f_{\%}$.

N	**Re-planning time** [s]			
	$\mathbf{f_{\%} = 10\%}$	$\mathbf{f_{\%} = 30\%}$	$\mathbf{f_{\%} = 50\%}$	$\mathbf{f_{\%} = 70\%}$
5	0.25 ± 0.02	0.24 ± 0.02	0.23 ± 0.01	0.24 ± 0.01
6	0.68 ± 0.02	0.65 ± 0.02	0.63 ± 0.01	0.62 ± 0.01
7	1.50 ± 0.02	1.48 ± 0.03	1.41 ± 0.02	1.42 ± 0.01
8	3.06 ± 0.04	2.97 ± 0.06	2.90 ± 0.04	2.88 ± 0.02
9	5.88 ± 0.08	5.70 ± 0.15	5.51 ± 0.03	5.51 ± 0.05
10	10.75 ± 0.20	10.21 ± 0.28	9.92 ± 0.09	9.87 ± 0.08

6.6.1 Discussion

The pre-operative plans generated for each configuration using our criterion (final paragraph of Section 6.4.2) successfully expose the ROI with a maximum of 6 actions, with the force exceeding the safety threshold ϵ at most once (Table 6.2).

Reliance on such plans leads to successful completion of the TR task (i.e., the ROI is exposed) in 13 out of 16 cases (Table 6.3), which means DEFRAS can accomplish the task even when the pre-operative information (i.e., knowledge of APs) is imprecise. Moreover, the results show that, even when starting from pre-operative plans which minimize the tissue force, the force threshold can be exceeded during task execution, hence force feedback is needed to prevent potentially dangerous tissue solicitation.

Furthermore, the tissue force reaches higher values ($> 1\,\mathrm{N}$) mostly in scenarios with more APs than the pre-operative configuration ($\oplus$ in Table). This is motivated by the fact that optimal plans might select a grasping point close to APs that are unknown pre-operatively. In this case, monitoring detects a difference between the pre-operative simulation and the actual simulation, and triggers APs update in the learning module. This leads to prediction of APs very close to the current grasping point, thus the estimated force increases. The results of our framework also show that when pre-operative APs overestimate actual APs ($\ominus$ in Table), the task is accomplished with fewer actions (except for fails) with respect to the case when APs are underestimated ($\oplus$ in Table). Therefore, it seems that overestimating APs in the pre-operative phase leads to more optimized task executions, reducing both interaction with the tissue and potentially unsafe tissue solicitation.

If we compare the results obtained when executing the TR task using DEFRAS with those obtained using FRAS (lower part of Table 6.3), it emerges that the presence of the pre-operative plan in DEFRAS is fundamental to optimize executions in terms of tissue solicitation (fewer actions, lower number of times the force threshold is exceeded and lower maximum force, considering median values). Comparison between the two frameworks also evidences the advantages of monitoring and learning modules, introduced by our deliberative framework. Whenever the mismatch between the pre-operative simulation and the actual simulation causes the monitoring module to trigger update of APs, BA-Net is able to provide an updated estimate of APs that brings the monitoring metric below threshold, except cases $1\oplus$

(task success, ROI is exposed) and $5\ominus$ (the task fails). Overall, the integration of the proposed deliberative modules results in higher median visibility (100% vs. 93%) and lower number of failed tasks (3 fails vs. 5) with respect to FRAS. We notice that FRAS is able to solve the task when DEFRAS fails in configurations $3\ominus$ and $5\ominus$. However, in both cases the monitoring module triggers APs update and the force threshold is overcome more times than all other configurations. This suggests that the pre-operative knowledge of APs is particularly far from reality in these cases. As a consequence, the estimate of the force from the pre-operative simulation provided by the FRAS cannot be considered reliable (being the simulation different from reality), potentially leading to unsafe executions by FRAS.

Re-planning efficiency Analyzing the computational performace of our framework in case of re-planning, our results show that the task reasoner is able to compute a new grasping point and a plan within $1.5\,\mathrm{s}$ when using a 7×7 grid discretization for grasping point definition, demonstrating high efficiency. This guarantees efficient adaptation to the quickly changing anatomical environment, introducing only a minor delay which is compatible with intra-operative times. From Table 6.4 it emerges that the most influencing parameter is N, while $f_{\%}$ does not significantly affect the average planning time. Reliance on the lowest resolution grid (5×5) for the definition of candidate grasping points for all the validation experiments presented in this Chapter allows us to maximize computational efficiency, both in pre-operative plan generation phase and intra-operative re-planning. Using a higher resolution grid would imply more time to generate the optimal task plan due to a larger search space. Although this would not represent a major limitation in the pre-operative phase, where there are no strict time constraints for plan definition, in case intra-operative re-planning is needed, it is important to rely on an efficient heuristic for the selection of the new grasping point, which ensures that plan generation does not introduce major delays in the execution.

Definition of the set of candidate grasping points as uniformly spaced above the tissue of interest represents a naive strategy which has been selected for our initial evaluation. Despite its simplicity, the same strategy does not only apply to the considered setup, but can be employed for more complex setups characterized by irregular geometries. It is worth highlighting that different grasping point selection strategies can be implemented and easily integrated within DEFRAS, e.g. based on path planning as in [178].

6.7 Experimental results of autonomous TR on the dVRK

Relevant variables describing the executions of the TR task on the experiments on the real setup are reported in Table 6.5.

6.7.1 Discussion

Obtained results on the real world experiments confirm that whenever pre-operative APs underestimate real APs, the monitoring and learning modules are triggered more frequently. The update of APs leads to successful task completion in all cases. Moreover, when the simulated and real APs match (cases labelled with $\ominus$), the number of actions is usually minimum (except for configuration B, with 8 actions for $\ominus$ and 7 for $\oplus$). This suggests that overall the optimal pre-operative plan leads to better performances when APs are perfectly known.

Table 6.5: Real world experiments using DEFRAS. For each real world configuration (A, B, C), we report details of TR executions of 3 different pre-operative plans, obtained when: ($\ominus$) underestimating real APs, ($\oplus$) overestimating real APs, ($\circleddash$) having perfect knowledge of the real APs.

	A			**B**			**C**				
	$\ominus$	$\oplus$	$\circleddash$	$\ominus$	$\oplus$	$\circleddash$	$\ominus$	$\oplus$	$\circleddash$	Median	IQR
Visibility [%]	99	95	100	100	100	71	99	83	96	99	95-100
Actions	13	15	9	24	7	8	14	7	7	9	7-14
$F > \epsilon$	11	20	8	0	8	20	8	8	6	8	8-11
F_{max} [N]	1.2	1.1	0.5	0.2	0.5	1.1	1.1	0.7	0.5	0.7	0.5-1.1
BA-Net	1/1	1/1	1/1	13/14	-	-	1/1	-	-	-	-

When qualitatively analyzing executions for configuration A, it emerges that both for $\ominus$ and $\oplus$ re-planning is required: in the first case due to high force after grasping too close to real APs, detected by the monitoring module; in the second case because of invisible ROI after optimal plan execution. Re-planning proves able to successfully expose the ROI in both cases. It is also interesting to consider TR execution when using the plan generated on the configuration with perfect knowledge of real APs ($\circleddash$). In this case, the monitoring module detects a mismatch between the real and the simulated scenarios, probably due to noise and uncertainty introduced by real cameras and the calibration procedure. DEFRAS proves able to cope with such situation and successfully complete the task.

The average visibility achieved by DEFRAS is 94%, which is aligned with the 90% visibility obtained when using GAIL (Section 5.6.1), and it improves the one reported in [176] (i.e., 83%). It is worth remarking that our framework is able to model uncertainties and handle re-planning, but these functions are absent in the approaches considered for comparison.

6.8 Conclusions

In this Chapter, we have presented a modular architecture which provides the fundamental building blocks and requisites for autonomy in robotic surgery. Our framework, called DEFRAS, introduces deliberative functionalities in order to deal with the uncertainty of anatomical environments. In particular, it is capable of monitoring the real execution and learning from real observations. The modular structure of the framework offers the opportunity to specialize each deliberative function individually, depending on the desired task. Moreover, a logic-based task planner guarantees interpretability of the surgical plan, thus allowing for human supervision, which is needed for reliable execution. A groundbreaking aspect of DEFRAS is the presence of a biomechanical simulation, which plays a crucial role in both the pre-operative and the intra-operative phase. Before the intervention, the simulation is exploited to design the task plan accounting for each patient's characteristics. Whereas, during task execution, it allows both to monitor the actual surgical environment and detect the presence of unexpected situations, and to provide additional information about the surgical scene.

Experiments on a representative surgical task, i.e. soft tissue retraction, have shown that our framework has the needed features to handle deformable environments with uncertain parameters, successully accomplishing the task. In particular, DEFRAS has proved able to learn and refine available but imprecise pre-operative knowledge about the location of attachment points, relying on constant monitoring of sensors and simulation data. In this way, any pre-operative plan can be adapted, and tissue force information from the simulation remains reliable for safe task completion. Compared with a framework without the deliberative functions of monitoring and learning, results show that the percentage of task success increases, especially when pre-operative knowledge of attachment points overestimates real ones. Moreover, reliance on a pre-operative patient-specific plan brings benefits in terms of minimal manipulation of the tissue, hence reduced risk of tissue damage.

Although the implementation of each module described in this Chapter is specific to accomplish the tissue retraction task, the general DEFRAS architecture is designed to provide the needed functionalities for autonomous execution of any surgical task. The modularity of DEFRAS will allow us to improve individual deliberative capabilities in order to deal with more realistic and challenging surgical scenarios in future works. Motion adaptation as in [231] and learning from demonstrations as in [243] can be easily implemented to replicate expert surgeons' dexterity. Furthermore, we plan to improve task knowledge description both pre-operatively from the analysis of surgical annotations [244] and intra-operatively by observing the surgeon in most critical situations, exploiting logic learning as in [245, 246]. Finally, we will investigate the possibility to introduce additional sensors on the instruments in order to enhance the sensing capabilities of the framework [247] and cope with specific problems of real surgical settings, e.g. point cloud occlusion. This will also allow to refine other patient-specific parameters and identify more robust task-specific thresholds.

Overall, this framework represents one of the first attempts to a complete deliberative architecture that integrates all the functionalities required to deal with the dynamic and uncertain surgical environments. Therefore, it contributes to make a step forward towards autonomy in surgical robotics.

Contributions of this Chapter

The main contributions of this Chapter are the following:

1. We present DEFRAS, a DEliberative Framework for Robot-Assisted Surgery that integrates all skills necessary to cope with uncertain anatomical environments, while guaranteeing successful task execution and respect of safety constraints. The framework is made pubicly available at https://gitlab.com/altairLab/tissue_retraction.git.

2. We show that a framework for autonomous surgery requires the integration of a biomechanical simulation for different reasons, from the design a personalized intervention plan in a pre-operative phase, to the continuous monitoring of the actual surgical environment and to deal with specific problems of real surgery, e.g. lack of force feedback, in the intra-operative phase.

3. We demonstrate that DEFRAS is able to cope with deformable environments with uncertain parameters, accomplishing a representative surgical task, i.e. tissue retraction, both in simulated and real experiments with the da Vinci Research Kit.

Publications linked to this Chapter

The content of this Chapter has been presented in the following publications:

1. Meli D, Tagliabue E, Dall'Alba D, Fiorini P: *Autonomous tissue retraction with a biomechanically informed logic based framework*, 2021 International Symposium on Medical Robotics (2021); 2021 Nov 17-19; Atlanta (USA);

2. Tagliabue E, Meli D, Dall'Alba D, Fiorini P: *Deliberation in autonomous robotic surgery: a framework for handling anatomical uncertainty*, IEEE International Conference on Robotics and Automation (2022); 2022 May 23-27; Philadelphia (USA);.

Chapter 7

Conclusion

The aim of this thesis has been to propose the different ways in which simulation can support to an autonomous surgical robotic system and enhance its performance.

We have shown that physics-based simulations (i.e., those describing anatomical behavior based on the laws of continuum mechanics) can be used to generate large amounts of data that can be employed to train machine learning methods. Reliance on synthetic data allows to compensate for the impracticality to acquire such a dataset in the medical field. In this Thesis, we have provided two examples of successful exploitation of dataset generated from simulations. First of all, they have been used to train a convolutional neural network which acts as a patient-specific model that predicts anatomical deformations given an input displacement, called U-Mesh. The performance of the U-Mesh has been evaluated on the prediction of the large non-linear deformations of breast tissues during ultrasound scanning. In this challenging context, the U-Mesh has proved able not only to reach accuracy levels aligned with those of the method used to generate the dataset, but also to provide extremely fast predictions without suffering from numerical stability, thus overcoming the main limitations of other existing simulation approaches. As a second example, we have proposed a method that updates the parametrization of a simulated model intra-operatively and is trained with synthetic data only. The developed strategy, called BA-Net, provides an up-to-date estimate of the location of tissues attachment points, whose position cannot be precisely known from pre-operative data and thus requires intra-operative identification. By relying on a deep network, BA-Net is able to update a biomechanical model in less than $2\,\mathrm{s}$ (considering the whole pipeline from sensor data acquisition to model update), reaching clinically acceptable performances both in terms of simulation accuracy and intra-operative time constraints, during manipulation of both synthetic and ex-vivo adipose tissue.

Simulation can assist the design, test and validation of an ARSS. To this end, we have shown that surgical tasks can be learnt in simulation leveraging on strategies that require multiple interactions with the environment (e.g., deep reinforcement learning). In this case, relying on a simulation environment is extremely helpful to avoid the limitations connected with data acquisition on real surgical robotic systems. We have demonstrated that, after testing in simulation, learnt tasks can be successfully transferred to the real system relying on UnityFlexML, our developed open-source framework that interfaces a stable and efficient simulation environment with the real da Vinci Research Kit. Finally, we have presented DEFRAS, a framework for autonomous surgery which includes deliberative functions in order to

cope with the dynamic and uncertain anatomical environment. DEFRAS integrates a biomechanical simulation, whose presence has proved essential to support the autonomous system both in the pre-operative phase for surgical task planning, and in the intra-operative phase for monitoring and enhancement of the real execution. Among its different functionalities, DEFRAS incorporates BA-Net to enable online update of the simulated model starting from real observations, leading to successful task execution even in case of inaccurate initial model parametrization.

By addressing the different benefits that simulation can bring to an ARSS, proposing possible solutions to tackle the main issues of existing approaches and publicly sharing the developed methods, we encourage a more widespread use of simulation in autonomous surgery.

7.1 Future research directions

Throughout this Thesis, we have demonstrated that simulation can support an ARSS both enhancing task execution and in the development of methodologies to address some specific challenges of surgery (e.g., intra-operative model update or task learning). In particular, in the last Chapter (Chapter 6) we have shown how a simulated environment and the developed methodology for intra-operative update of attachment points (BA-Net) can be successfully integrated within a framework for autonomous surgery (DEFRAS). Their role has proved fundamental for the autonomous execution of tissue retraction on a deformable phantom (i.e., a single task on a controlled setup), which has been selected for our preliminary evaluation. The goal of future works will be to consider a more realistic clinical scenario, starting from an anatomical phantom and then moving to ex-vivo settings. In terms of task, we will tackle the autonomous execution of other parts of a surgical procedure (e.g., partial nephrectomy), with the aim of automating the entire intervention. This will imply to address other surgical tasks like dissection, cutting and suturing. In turns, it will require the implementation of advanced strategies to cope with the specific challenges arising due to the new actions involved, in terms of modeling, control and simulation. We expect that simulation will play a crucial role to properly deal with such a challenging context, supporting the development, validation and task execution phases. To tackle such a complex scenario, we will leverage on the promising results achieved in this work and address their limitations. In particular, the main direction for future research will be the improvement of the methodologies developed in this Thesis, strengthening their robustness and generalization capabilities, with the final goal of supporting the autonomous execution of increasingly complex surgical tasks and their integration within an extended version of DEFRAS.

First of all, moving to a more realistic setup and task will introduce some challenges in the design of a simulated environment with the required features, i.e. able to achieve high accuracy, while guaranteeing computational performance compatible with the execution workflow. For the validation scenario considered in this Thesis (i.e., tissue retraction of a silicone phantom), such performance requirements could be met with the FE method running on CPU, relying on the corotational formulation [48]. However, we expect that more optimized solution methods and/or formulations will be needed as soon as more complex interactions are involved, e.g. contacts with surrounding organs and topological modifications arising in case of cutting and suturing, which would increase the computational burden of the simulation. However, it is worth highlighting that the role of the simulated environment in DEFRAS

is to complement real sensor data while monitoring the execution. Although it is desirable to have a simulation running in real-time for such purpose, it will still be acceptable if the simulation can provide feedback at a lower rate with respect to the main execution loop. The FE method remains the preferable approach for the simulated environment due to its reliance on real tissue mechanical properties, which facilitates model parametrization both in the pre-operative and in the intra-operative phase. As a consequence, we will investigate the integration of optimized FE implementations that take advantage of the processing power of GPUs to improve computational performance, such as the one proposed in [60].

However, it is possible that the speedup obtained even with optimized FE implementations will not be sufficient to achieve acceptable performance. Therefore, another promising research direction is to rely on a simulated environment driven by a neural network, likewise the U-Mesh presented in Chapter 3, which has obtained excellent computational performance, together with high accuracy and stability. However, the current method presents some limitations that need to be addressed. At present, the U-Mesh represents a patient-specific model that can handle a single anatomy. Generalization to a new geometry is already possible, but requires re-training of the network with a new geometry-specific dataset. Therefore, as an initial step we will make the network able to deal with any input geometry, e.g., encoding shape information in the input, following the same approach already proposed in Chapter 4. Two major aspects need to be addressed to assess the feasibility of exploiting a neural network as simulation method within DEFRAS. First, we will investigate the possibility of accounting for imprecise modelling assumptions in the training data generation phase. In fact, we have demonstrated that pre-operative data and knowledge are not sufficient to create an accurate patient-specific model, thus requiring the design of strategies for intra-operative model update. Differently from FE-based simulations, updating parameters will not be straightforward when the simulation relies on a neural network. A possible option would be to find a strategy to condition the network output based on observed data or new parameters. For example, model parameters might be given as input to the method, as in [106] where boundary conditions are provided as input features. An alternative would be to include samples with variable parametrization within the training dataset (e.g. constitutive models, mechanical properties and boundary conditions) such that the network would implicitly learn to compensate for inaccurate parameters. The second aspect that requires investigation is the possibility to model complex interactions and topological modifications. At the moment, the U-Mesh requires a fixed and regular grid as input. Topological modifications might be addressed either by encoding ad-hoc features in the input grid (e.g. information about connectivity between neighbouring voxels) or by considering different kinds of network architectures (e.g. Long Short-Term Memory networks [248] or GANs [207]).

Targeting a more complex environment and task will inevitably introduce challenges in finding the proper parametrization of the models, which makes the presence of strategies for intra-operative model update even more important. Therefore, it will be essential to improve the robustness of the proposed pipeline for model update, in its entirety. As a first point, we will focus on improving the pre-processing step, which at the moment relies on geometric and color-based segmentation and decimation of the tissue point cloud. This simple pre-processing pipeline is highly influenced by lighting conditions, the presence of occlusions and moving tissues, even in controlled scenarios as the ones considered in this work. In future works, we will try to reduce the effects of noise at the source, e.g. relying on edge maps

[249, 250]. Moreover, we plan to integrate methods for robust tracking of tissue displacement based on automatic feature points extraction [251, 252], and improve robustness of the scene segmentation exploiting for example semantic information [253]. It is worth stressing out that improvements in the perception pipeline of the environment will not only impact the accuracy of the model update strategy, but it will be fundamental to enhance the sensing capabilities of DEFRAS. For example, the robustness of the discrepancy metric computed by the monitoring module will be impacted by the availability of more accurate data from real sensors.

The developed strategy for intra-operative update of APs (i.e., BA-Net) has also room for improvements. In particular, the accuracy of the method is not only impacted by the quality of the input point cloud, which can be addressed with an improved pre-processing (as described in the previous paragraph), but also by the quality of the displacement estimation, which depends on the computation of corresponding points between the organ geometry and the acquired point cloud. This is an important aspect that we plan to tackle in the future by letting the network implicitly solve for surface correspondences as in [14], thus directly providing the point cloud as input. A possible alternative to skip the surface matching problem would be to rely on other network architectures, like PointNet [254], that take as input the point clouds directly. This approach has already shown promise to solve several tasks in the computer vision field, as surveyed in [255]. Furthermore, we plan to improve the robustness of the current method by investigating possible ways to enforce temporal consistency of the predictions, thus accounting for the temporal evolution of surgery. To this purpose, we will develop a novel DNN model that considers multiple sequential measurements (i.e. tissue surface 3D reconstructions) from the intra-operative sensors. We will consider an improved version of the currently employed U-Net, equipped with Long Short-Term Memory and Attention Gate Cells, which have demonstrated promising capabilities to extract the correlation between consecutive measurements [256], which might lead to a boost in prediction accuracy. As an additional improvement, we will enhance the adhesion modeling by replacing the current binary constraints used to model APs with a more complex model, for instance by modelling the attachments with groups of springs with variable stiffness, depending on the extent of the constraint. This more realistic model will be exploited both to generate the synthetic training dataset and to test the method.

Addressing a more clinically realistic scenario will require us to improve our current experimental setup. First of all, we will test autonomous execution on phantoms generated from real anatomical atlases [257], and then move to ex-vivo (both animal and human) experiments. Afterwards, we will investigate how to make our setup compatible with the limited workspace available in RMIS settings. The main factor that will prevent direct transfer of our current setup to real clinical settings is the presence of an external RGBD sensor (in addition to the endoscopic camera), which is used to obtain depth information and maximize tissue visibility for monitoring purposes. To tackle this issue, we plan to integrate an advanced stereo-endoscope sensor with robust depth sensing capabilities provided by state-of-the-art Micro Electro-Mechanical Systems directly on the dVRK endoscopic arm, using an actuated endoscope similarly to [258]. This setup would have a minimum sensing distance of few centimeters, making the system suitable for an RMIS scenario.

In future works, we will also investigate how to enhance the awareness of the framework, hence we will consider the introduction of advanced sensing technologies on the instruments.

We plan to integrate Electrical Bioimpedance (EBI) sensors on the dVRK arms to make the system able to identify the different tissues during interaction with them [259]. This technology can also be helpful to verify if grasping has been successful, compensating for possible inaccuracies in the registration process. In addition, we will investigate the possibility to equip the dVRK arms with strain gauges to obtain an estimate of the interaction force starting from instrument deflection, e.g. exploiting the method proposed in [260]. Information on the exerted forces from real sensors opens promising research directions towards the development of new strategies devoted to the refinement of patient-specific mechanical parameters [261].

Finally, autonomous execution of more complex surgical tasks will require to extend the individual deliberative capabilities of DEFRAS to deal with such challenging scenarios. We will start with improving task knowledge description both pre-operatively from the analysis of surgical annotations [244] and intra-operatively by observing the surgeon's reactions to critical situations [245, 246]. Moreover, performing complex tasks would require enhanced dexterity that can be achieved implementing either motion adaptation strategies as [231] or learning from demonstrations as [243]. Learning from demonstrations approaches can be implemented leveraging on UnityFlexML, the framework presented in Chapter 5, where we have shown that they can even be coupled with deep reinforcement learning. The simulation environment provided by UnityFlexML, which relies on the position based dynamics method, is particularly promising to achieve efficient simulations involving multiple organs and topology modifications, with a sufficient level of realism. However, since the design of handcrafted reward functions for complex tasks would be not straightforward, in future works we will test image-based learning strategies and integrate image-to-image translation as [262, 208, 263] for style transfer from simulated to real images.

With this Thesis, we have shown that the availability of a patient-specific anatomical simulation can contribute to the development of an ARSS in many different ways. As a consequence, simulation needs to be integrated as a key component in next generation surgical robotic systems, which will have increasingly autonomous capabilities. Despite the promising results already obtained, the methods proposed in this Thesis offer margin for improvement, especially to make them closer to applicability to real clinical conditions, thus opening space for new exciting research directions.

Bibliography

[1] W. Kilby, J. Dooley, G. Kuduvalli, S. Sayeh, and C. Maurer Jr, "The CyberKnife® robotic radiosurgery system in 2010," *Technology in cancer research & treatment*, vol. 9, no. 5, pp. 433–452, 2010.

[2] J. L. Pons, "Rehabilitation exoskeletal robotics," *IEEE Engineering in Medicine and Biology Magazine*, vol. 29, no. 3, pp. 57–63, 2010.

[3] "MAKO," https://www.stryker.com/us/en/portfolios/orthopaedics/joint-replacement/mako-robotic-arm-assisted-surgery.html, accessed: 2021-09-30.

[4] "ROSA® knee system," https://www.zimmerbiomet.com/en/products-and-solutions/specialties/knee/rosa--knee-system.html, accessed: 2021-09-30.

[5] "Mazor X Stealth," https://www.medtronic.com/it-it/operatori-sanitari/therapies-procedures/spinal-orthopaedic/spine-robotics.html, accessed: 2021-09-30.

[6] "Intuitive Surgical Inc." https://www.intuitive.com/en-us, accessed: 2021-09-30.

[7] C. D'Ettorre, A. Mariani, A. Stilli, F. R. y Baena, P. Valdastri, A. Deguet, P. Kazanzides, R. H. Taylor, G. S. Fischer, S. P. DiMaio, *et al.*, "Accelerating surgical robotics research: A review of 10 years with the da vinci research kit," *IEEE Robotics & Automation Magazine*, 2021.

[8] P. Kazanzides, Z. Chen, A. Deguet, G. S. Fischer, R. H. Taylor, and S. P. DiMaio, "An open-source research kit for the da vinci® surgical system," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 6434–6439.

[9] A. Attanasio, B. Scaglioni, E. De Momi, P. Fiorini, and P. Valdastri, "Autonomy in surgical robotics," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 4, pp. 651–679, 2021.

[10] H. Choi, C. Crump, C. Duriez, A. Elmquist, G. Hager, D. Han, F. Hearl, J. Hodgins, A. Jain, F. Leve, *et al.*, "On the use of simulation in robotics: Opportunities, challenges, and suggestions for moving forward," *Proceedings of the National Academy of Sciences*, vol. 118, no. 1, 2021.

[11] N. Haouchine, W. Kuang, S. Cotin, and M. Yip, "Vision-based force feedback estimation for robot-assisted surgery using instrument-constrained biomechanical three-dimensional maps," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2160–2165, 2018.

[12] J. V. Hajnal and D. L. Hill, *Medical image registration*. CRC press, 2001.

[13] N. Haouchine, J. Dequidt, I. Peterlik, E. Kerrien, M.-O. Berger, and S. Cotin, "Image-guided simulation of heterogeneous tissue deformation for augmented reality during hepatic surgery," in *Mixed and Augmented Reality (ISMAR), 2013 IEEE International Symposium on*. IEEE, 2013, pp. 199–208.

[14] M. Pfeiffer, C. Riediger, S. Leger, J.-P. Kühn, D. Seppelt, R.-T. Hoffmann, J. Weitz, and S. Speidel, "Non-rigid volume to surface registration using a data-driven biomechanical model," pp. 724–734, 2020.

[15] A. Mendizabal, E. Tagliabue, T. Hoellinger, J.-N. Brunet, S. Nikolaev, and S. Cotin, "Data-driven simulation for augmented surgery," in *Developments and Novel Approaches in Biomechanics and Metamaterials*, July 2020, vol. 132, pp. 71–96.

[16] N. Abolhassani, R. Patel, and M. Moallem, "Needle insertion into soft tissue: A survey," *Medical engineering & physics*, vol. 29, no. 4, pp. 413–431, 2007.

[17] Z. Zuo, W. Y. Qian, X. Liao, and P.-A. Heng, "Position based catheterization and angiography simulation," in *2018 IEEE 6th International Conference on Serious Games and Applications for Health (SeGAH)*. IEEE, 2018, pp. 1–7.

[18] M. Hong, J. W. Rozenblit, and A. J. Hamilton, "Simulation-based surgical training systems in laparoscopic surgery: a current review," *Virtual Reality*, vol. 25, no. 2, pp. 491–510, 2021.

[19] C. Basdogan, M. Sedef, M. Harders, and S. Wesarg, "Vr-based simulators for training in minimally invasive surgery," *IEEE Computer Graphics and Applications*, vol. 27, no. 2, pp. 54–66, 2007.

[20] J. Zhang, Y. Zhong, and C. Gu, "Deformable models for surgical simulation: A survey," *IEEE reviews in biomedical engineering*, vol. 11, pp. 143–164, 2018.

[21] U. Meier, O. López, C. Monserrat, M. C. Juan, and M. Alcaniz, "Real-time deformable models for surgery simulation: a survey," *Computer methods and programs in biomedicine*, vol. 77, no. 3, pp. 183–197, 2005.

[22] M. J. Ackerman, "The visible human project: a resource for education." *Academic medicine: journal of the Association of American Medical Colleges*, vol. 74, no. 6, pp. 667–670, 1999.

[23] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-net: Fully convolutional neural networks for volumetric medical image segmentation," in *2016 fourth international conference on 3D vision (3DV)*. IEEE, 2016, pp. 565–571.

[24] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3d surface construction algorithm," *ACM siggraph computer graphics*, vol. 21, no. 4, pp. 163–169, 1987.

[25] N. Golias and R. Dutton, "Delaunay triangulation and 3d adaptive mesh generation," *Finite elements in analysis and design*, vol. 25, no. 3-4, pp. 331–341, 1997.

[26] M. C. Krygier, T. LaBonte, C. Martinez, C. Norris, K. Sharma, L. N. Collins, P. P. Mukherjee, and S. A. Roberts, "Quantifying the unknown impact of segmentation uncertainty on image-based simulations," *Nature communications*, vol. 12, no. 1, pp. 1–11, 2021.

[27] Y.-C. Fung *et al.*, *Mechanical properties of living tissues*. Springer, 1993, vol. 547.

[28] R. M. Sigrist, J. Liau, A. El Kaffas, M. C. Chammas, and J. K. Willmann, "Ultrasound elastography: review of techniques and clinical applications," *Theranostics*, vol. 7, no. 5, p. 1303, 2017.

[29] Y. K. Mariappan, K. J. Glaser, and R. L. Ehman, "Magnetic resonance elastography: a review," *Clinical anatomy*, vol. 23, no. 5, pp. 497–511, 2010.

[30] H. Delingette, "Toward realistic soft-tissue modeling in medical simulation," *Proceedings of the IEEE*, vol. 86, no. 3, pp. 512–523, 1998.

[31] N.-H. Kim, *Introduction to nonlinear finite element analysis*. Springer Science & Business Media, 2014.

[32] M. Freutel, H. Schmidt, L. Dürselen, A. Ignatius, and F. Galbusera, "Finite element modeling of soft tissues: material models, tissue interaction and challenges," *Clinical Biomechanics*, vol. 29, no. 4, pp. 363–372, 2014.

[33] R. W. Ogden, *Non-linear elastic deformations*. Courier Corporation, 1997.

[34] ——, "Large deformation isotropic elasticity–on the correlation of theory and experiment for incompressible rubberlike solids," *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, vol. 326, no. 1567, pp. 565–584, 1972.

[35] G. Bedrosian, "Shape functions and integration formulas for three-dimensional finite element analysis," *International journal for numerical methods in engineering*, vol. 35, no. 1, pp. 95–108, 1992.

[36] O. Schenk and K. Gärtner, "Solving unsymmetric sparse systems of linear equations with pardiso," *Future Generation Computer Systems*, vol. 20, no. 3, pp. 475–487, 2004.

[37] J. R. Shewchuk *et al.*, "An introduction to the conjugate gradient method without the agonizing pain," 1994.

[38] M. Bro-Nielsen, "Finite element modeling in surgery simulation," *Proceedings of the IEEE*, vol. 86, no. 3, pp. 490–503, 1998.

[39] K. Brock, M. Sharpe, L. Dawson, S. Kim, and D. Jaffray, "Accuracy of finite element model-based multi-organ deformable image registration," *Medical physics*, vol. 32, no. 6Part1, pp. 1647–1659, 2005.

[40] J. A. Schnabel, C. Tanner, A. D. Castellano-Smith, A. Degenhard, M. O. Leach, D. R. Hose, D. L. Hill, and D. J. Hawkes, "Validation of nonrigid image registration using finite-element methods: application to breast mr images," *IEEE transactions on medical imaging*, vol. 22, no. 2, pp. 238–247, 2003.

[41] F. Morin, H. Courtecuisse, I. Reinertsen, F. Le Lann, O. Palombi, Y. Payan, and M. Chabanas, "Brain-shift compensation using intraoperative ultrasound and constraint-based biomechanical simulation," *Medical image analysis*, vol. 40, pp. 133–153, 2017.

[42] M. Bro-Nielsen and S. Cotin, "Real-time volumetric deformable models for surgery simulation using finite elements and condensation," in *Computer graphics forum*, vol. 15, no. 3. Wiley Online Library, 1996, pp. 57–66.

[43] S. Cotin, H. Delingette, and N. Ayache, "Real-time elastic deformations of soft tissues for surgery simulation," *IEEE transactions on Visualization and Computer Graphics*, vol. 5, no. 1, pp. 62–73, 1999.

[44] K. Miller, G. Joldes, D. Lance, and A. Wittek, "Total lagrangian explicit dynamics finite element algorithm for computing soft tissue deformation," *Communications in numerical methods in engineering*, vol. 23, no. 2, pp. 121–134, 2007.

[45] R. R. Garlapati, A. Roy, G. R. Joldes, A. Wittek, A. Mostayed, B. Doyle, S. K. Warfield, R. Kikinis, N. Knuckey, S. Bunt, *et al.*, "More accurate neuronavigation data provided by biomechanical modeling instead of rigid registration," *Journal of neurosurgery*, vol. 120, no. 6, pp. 1477–1483, 2014.

[46] J. Zhang, "A direct jacobian total lagrangian explicit dynamics finite element algorithm for real-time simulation of hyperelastic materials," *International Journal for Numerical Methods in Engineering*, vol. 122, no. 20, pp. 5744–5772, 2021.

[47] S. Marchesseau, T. Heimann, S. Chatelin, R. Willinger, and H. Delingette, "Multiplicative jacobian energy decomposition method for fast porous visco-hyperelastic soft tissue model," in *MICCAI*. Springer, 2010, pp. 235–242.

[48] M. Müller, J. Dorsey, L. McMillan, R. Jagnow, and B. Cutler, "Stable real-time deformations," in *Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, 2002, pp. 49–54.

[49] C. A. Felippa and B. Haugen, "A unified formulation of small-strain corotational finite elements: I. theory," *Computer Methods in Applied Mechanics and Engineering*, vol. 194, no. 21-24, pp. 2285–2335, 2005.

[50] R. Plantefève, I. Peterlik, N. Haouchine, and S. Cotin, "Patient-specific biomechanical modeling for guidance during minimally-invasive hepatic surgery," *Annals of biomedical engineering*, vol. 44, no. 1, pp. 139–153, 2016.

[51] I. Peterlík, H. Courtecuisse, R. Rohling, P. Abolmaesumi, C. Nguan, S. Cotin, and S. Salcudean, "Fast elastic registration of soft tissues under large deformations," *Medical image analysis*, vol. 45, pp. 24–40, 2018.

[52] N. Golse, A. Petit, M. Lewin, E. Vibert, and S. Cotin, "Augmented reality during open liver surgery using a markerless non-rigid registration system," *Journal of Gastrointestinal Surgery*, vol. 25, no. 3, pp. 662–671, 2021.

[53] N. Haouchine, P. Juvekar, W. M. Wells III, S. Cotin, A. Golby, and S. Frisken, "Deformation aware augmented reality for craniotomy using 3d/2d non-rigid registration of cortical vessels," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2020, pp. 735–744.

[54] P. Baksic, H. Courtecuisse, and B. Bayle, "Shared control strategy for needle insertion into deformable tissue using inverse finite element simulation," in *ICRA 2021-International Conference on Robotics and Automation*, 2021.

[55] N. Hamzé, I. Peterlík, S. Cotin, and C. Essert, "Preoperative trajectory planning for percutaneous procedures in deformable environments," *Computerized Medical Imaging and Graphics*, vol. 47, pp. 16–28, 2016.

[56] P. Moreira, I. Peterlik, M. Herink, C. Duriez, S. Cotin, and S. Misra, "Modelling prostate deformation: Sofa versus experiments," *Prostate*, vol. 17, no. 83.0, pp. 1–0, 2013.

[57] O. Goksel, K. Sapchuk, and S. E. Salcudean, "Haptic simulator for prostate brachytherapy with simulated needle and probe interaction," *IEEE transactions on haptics*, vol. 4, no. 3, pp. 188–198, 2011.

[58] J. Allard, H. Courtecuisse, and F. Faure, "Implicit fem solver on gpu for interactive deformation simulation," in *GPU computing gems Jade Edition*. Elsevier, 2012, pp. 281–294.

[59] S. F. Johnsen, Z. A. Taylor, M. J. Clarkson, J. Hipwell, M. Modat, B. Eiben, L. Han, Y. Hu, T. Mertzanidou, D. J. Hawkes, *et al.*, "Niftysim: A gpu-based nonlinear finite element package for simulation of soft tissue biomechanics," *International journal of computer assisted radiology and surgery*, vol. 10, no. 7, pp. 1077–1095, 2015.

[60] H. Courtecuisse, J. Allard, P. Kerfriden, S. P. Bordas, S. Cotin, and C. Duriez, "Real-time simulation of contact and cutting of heterogeneous soft-tissues," *Medical image analysis*, vol. 18, no. 2, pp. 394–410, 2014.

[61] J. H. Hipwell, V. Vavourakis, L. Han, T. Mertzanidou, B. Eiben, and D. J. Hawkes, "A review of biomechanically informed breast image registration," *Physics in Medicine & Biology*, vol. 61, no. 2, p. R1, 2016. [Online]. Available: http://stacks.iop.org/0031-9155/61/i=2/a=R1

[62] E. Cueto and F. Chinesta, "Real time simulation for computational surgery: a review," *Advanced Modeling and Simulation in Engineering Sciences*, vol. 1, no. 1, pp. 1–18, 2014.

[63] S. Niroomandi, I. Alfaro, E. Cueto, and F. Chinesta, "Real-time deformable models of non-linear tissues by model reduction techniques," *Computer methods and programs in biomedicine*, vol. 91, no. 3, pp. 223–231, 2008.

[64] ——, "Accounting for large deformations in real-time simulations of soft tissues based on reduced-order models," *Computer Methods and Programs in Biomedicine*, vol. 105, no. 1, pp. 1–12, 2012.

[65] ——, "Model order reduction for hyperelastic materials," *International Journal for Numerical Methods in Engineering*, vol. 81, no. 9, pp. 1180–1206, 2010.

[66] S. Bhattacharjee and K. Matouš, "A nonlinear manifold-based reduced order model for multiscale analysis of heterogeneous hyperelastic materials," *Journal of Computational Physics*, vol. 313, pp. 635–653, 2016.

[67] S. Niroomandi, D. González, I. Alfaro, F. Bordeu, A. Leygue, E. Cueto, and F. Chinesta, "Real-time simulation of biological soft tissues: a pgd approach," *International journal for numerical methods in biomedical engineering*, vol. 29, no. 5, pp. 586–600, 2013.

[68] C. Monserrat, U. Meier, M. Alcaniz, F. Chinesta, and M. C. Juan, "A new approach for the real-time simulation of tissue deformations in surgery simulation," *Computer Methods and Programs in Biomedicine*, vol. 64, no. 2, pp. 77–85, 2001.

[69] P. Wang, A. Becker, I. Jones, A. Glover, S. Benford, C. Greenhalgh, and M. Vloeberghs, "Virtual reality simulation of surgery with haptic feedback based on the boundary element method," *Computers & structures*, vol. 85, no. 7-8, pp. 331–339, 2007.

[70] M. Doblaré, E. Cueto, B. Calvo, M. Martínez, J. Garcia, and J. Cegonino, "On the employ of meshless methods in biomechanics," *Computer methods in applied mechanics and engineering*, vol. 194, no. 6-8, pp. 801–821, 2005.

[71] A. Horton, A. Wittek, G. R. Joldes, and K. Miller, "A meshless total lagrangian explicit dynamics algorithm for surgical simulation," *International journal for numerical methods in biomedical engineering*, vol. 26, no. 8, pp. 977–998, 2010.

[72] G. Zhang, A. Wittek, G. Joldes, X. Jin, and K. Miller, "A three-dimensional nonlinear meshfree algorithm for simulating mechanical responses of soft tissue," *Engineering Analysis with Boundary Elements*, vol. 42, pp. 60–66, 2014.

[73] A. Palyanov, S. Khayrulin, and S. D. Larson, "Application of smoothed particle hydrodynamics to modeling mechanisms of biological tissue," *Advances in Engineering Software*, vol. 98, pp. 1–11, 2016.

[74] K. Miller, A. Horton, G. Joldes, and A. Wittek, "Beyond finite elements: a comprehensive, patient-specific neurosurgical simulation utilizing a meshless method," *Journal of biomechanics*, vol. 45, no. 15, pp. 2698–2701, 2012.

[75] S. F. Gibson and B. Mirtich, "A survey of deformable modeling in computer graphics," Citeseer, Tech. Rep., 1997.

[76] D. Zerbato, S. Galvan, and P. Fiorini, "Calibration of mass spring models for organ simulations," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2007, pp. 370–375.

[77] B. Lloyd, G. Székely, and M. Harders, "Identification of spring parameters for deformable object simulation," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 5, pp. 1081–1094, 2007.

[78] M. N. Omar and Y. Zhong, "A review of mass spring method improvements for modeling soft tissue deformation," *Human-Centered Technology for a Better Tomorrow*, pp. 203–215, 2022.

[79] H. Dehghani Ashkezari, A. Mirbagheri, S. Behzadipour, and F. Farahmand, "A mass-spring-damper model for real time simulation of the frictional grasping interactions between surgical tools and large organs," *Scientia Iranica*, vol. 22, no. 5, pp. 1833–1841, 2015.

[80] M. Harders, R. Hutter, A. Rutz, P. Niederer, and G. Székely, "Comparing a simplified fem approach with the mass-spring model for surgery simulation," *Studies in health technology and informatics*, pp. 103–109, 2003.

[81] U. Kühnapfel, H. K. Cakmak, and H. Maaß, "Endoscopic surgery training using virtual reality and deformable tissue simulation," *Computers & graphics*, vol. 24, no. 5, pp. 671–682, 2000.

[82] Y. Zhang, Z. Yuan, Y. Ding, J. Zhao, Z. Duan, and M. Sun, "Real time simulation of tissue cutting based on gpu and cuda for surgical training," in *2010 International Conference on Biomedical Engineering and Computer Science*. IEEE, 2010, pp. 1–4.

[83] M. Müller, B. Heidelberger, M. Teschner, and M. Gross, "Meshless deformations based on shape matching," in *ACM transactions on graphics (TOG)*, vol. 24, no. 3. ACM, 2005, pp. 471–478.

[84] J. Bender, M. Müller, and M. Macklin, "A survey on position based dynamics, 2017," *EUROGRAPHICS 2017 Tutorials*, 2017.

[85] I. Berndt, R. Torchelsen, and A. Maciel, "Efficient surgical cutting with position-based dynamics," *IEEE computer graphics and applications*, vol. 38, no. 3, pp. 24–31, 2017.

[86] J. Pan, J. Bai, X. Zhao, A. Hao, and H. Qin, "Real-time haptic manipulation and cutting of hybrid soft tissue models by extended position-based dynamics," *Computer Animation and Virtual Worlds*, vol. 26, no. 3-4, pp. 321–335, 2015.

[87] Y. Wang, Y. Xiong, K. Xu, K. Tan, and G. Guo, "A mass-spring model for surface mesh deformation based on shape matching." in *GRAPHITE*, vol. 6, 2006, pp. 375–380.

[88] L. Xu, Y. Lu, and Q. Liu, "Integrating viscoelastic mass spring dampers into position-based dynamics to simulate soft tissue deformation in real time," *Royal Society open science*, vol. 5, no. 2, p. 171587, 2018.

[89] B. Kubiak, N. Pietroni, F. Ganovelli, and M. Fratarcangeli, "A robust method for real-time thread simulation," in *Proceedings of the 2007 ACM symposium on Virtual reality software and technology*. ACM, 2007, pp. 85–88.

[90] M. Camara, E. Mayer, A. Darzi, and P. Pratt, "Simulation of patient-specific deformable ultrasound imaging in real time," in *Imaging for Patient-Customized Simulations and Systems for Point-of-Care Ultrasound*. Springer, 2017, pp. 11–18.

[91] ——, "Soft tissue deformation for surgical simulation: a position-based dynamics approach," *International journal of computer assisted radiology and surgery*, vol. 11, no. 6, pp. 919–928, 2016.

[92] A. Segato, C. Di Vece, S. Zucchelli, M. Di Marzo, T. Wendler, M. F. Azampour, S. Galvan, R. Secoli, and E. De Momi, "Position-based dynamics simulator of brain deformations for path planning and intra-operative control in keyhole neurosurgery," *IEEE Robotics and Automation Letters*, 2021.

[93] S. Dawda, M. Camara, P. Pratt, J. Vale, A. Darzi, and E. Mayer, "Patient-specific simulation of pneumoperitoneum for laparoscopic surgical planning," *Journal of medical systems*, vol. 43, no. 10, pp. 1–9, 2019.

[94] S. F. Gibson, "3d chainmail: a fast algorithm for deforming volumetric objects," in *Proceedings of the 1997 symposium on Interactive 3D graphics*, 1997, pp. 149–ff.

[95] A. R. Aguilera, A. León, G. Arroyo, and J. M. Mantas, "Sp-chainmail: a gpu-based sparse parallel chainmail algorithm for deforming medical volumes," *The Journal of Supercomputing*, vol. 71, pp. 3482–3499, 2015.

[96] D. Fortmeier, A. Mastmeyer, and H. Handels, "Image-based palpation simulation with soft tissue deformations using chainmail on the gpu," in *Bildverarbeitung für die Medizin 2013*. Springer, 2013, pp. 140–145.

[97] S. F. Frisken-Gibson, "Using linked volumes to model object collisions, deformation, cutting, carving, and joining," *IEEE transactions on visualization and computer graphics*, vol. 5, no. 4, pp. 333–348, 1999.

[98] X. Wang and A. Fenster, "A virtual reality based 3d real-time interactive brachytherapy simulation of needle insertion and seed implantation," in *2004 2nd IEEE International Symposium on Biomedical Imaging: Nano to Macro (IEEE Cat No. 04EX821)*. IEEE, 2004, pp. 280–283.

[99] D.-I. A. Neubauer, "A chainmail algorithm for direct volume deformation in virtual endoscopy applications," Ph.D. dissertation, PhD thesis, 2005.

[100] S. Gibson, J. Samosky, A. Mor, C. Fyock, E. Grimson, T. Kanade, R. Kikinis, H. Lauer, N. McKenzie, S. Nakajima, *et al.*, "Simulating arthroscopic knee surgery using volumetric object representations, real-time volume rendering and haptic feedback," in *CVRMed-MRCAS'97*. Springer, 1997, pp. 367–378.

[101] A. Dhillon and G. K. Verma, "Convolutional neural network: a review of models, methodologies and applications to object detection," *Progress in Artificial Intelligence*, vol. 9, no. 2, pp. 85–112, 2020.

[102] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[103] K. Morooka, X. Chen, R. Kurazume, S. Uchida, K. Hara, Y. Iwashita, and M. Hashizume, "Real-time nonlinear fem with neural network for simulating soft organ model deformation," in *MICCAI*. Springer, 2008, pp. 742–749.

[104] M. Tonutti, G. Gras, and G.-Z. Yang, "A machine learning approach for real-time modelling of tissue deformation in image-guided neurosurgery," *Artificial intelligence in medicine*, vol. 80, pp. 39–47, 2017.

[105] K. J. Rechowicz and F. D. McKenzie, "Development and validation methodology of the nuss procedure surgical planner," *Simulation*, vol. 89, no. 12, pp. 1474–1488, 2013.

[106] M. Pfeiffer, C. Riediger, J. Weitz, and S. Speidel, "Learning soft tissue behavior of organs for surgical navigation with convolutional neural networks," *International Journal of Computer Assisted Radiology and Surgery*, pp. 1–9, 2019.

[107] J.-N. Brunet, A. Mendizabal, A. Petit, N. Golse, E. Vibert, and S. Cotin, "Physics-based deep neural network for augmented reality during liver surgery," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2019, pp. 137–145.

[108] D. Lorente, F. Martínez-Martínez, M. J. Rupérez, M. Lago, M. Martínez-Sober, P. Escandell-Montero, J. M. Martínez-Martínez, S. Martínez-Sanchis, A. J. Serrano-López, C. Monserrat, *et al.*, "A framework for modelling the biomechanical behaviour of the human liver during breathing in real time using machine learning," *Expert Systems with Applications*, vol. 71, pp. 342–357, 2017.

[109] R. Phellan, B. Hachem, J. Clin, J.-M. Mac-Thiong, and L. Duong, "Real-time biomechanics using the finite element method and machine learning: Review and perspective," *Medical Physics*, vol. 48, no. 1, pp. 7–18, 2021.

[110] F. Martínez-Martínez, M. J. Rupérez-Moreno, M. Martínez-Sober, *et al.*, "A finite element-based machine learning approach for modeling the mechanical behavior of the breast tissues under compression in real-time," *Computers in biology and medicine*, vol. 90, pp. 116–124, 2017.

[111] U. Yamamoto, M. Nakao, M. Ohzeki, and T. Matsuda, "Deformation estimation of an elastic object by partial observation using a neural network," *arXiv preprint arXiv:1711.10157*, 2017.

[112] M. Zanotel, I. Bednarova, V. Londero, A. Linda, M. Lorenzon, R. Girometti, and C. Zuiani, "Automated breast ultrasound: basic principles and emerging clinical applications," *La radiologia medica*, vol. 123, no. 1, pp. 1–12, 2018.

[113] M. Z. Mahmoud, M. Aslam, M. Alsaadi, M. A. Fagiri, and B. Alonazi, "Evolution of robot-assisted ultrasound-guided breast biopsy systems," *Journal of radiation research and applied sciences*, vol. 11, no. 1, pp. 89–97, 2018.

[114] J. Carriere, J. Fong, T. Meyer, R. Sloboda, S. Husain, N. Usmani, and M. Tavakoli, "An admittance-controlled robotic assistant for semi-autonomous breast ultrasound scanning," in *2019 International Symposium on Medical Robotics (ISMR)*. IEEE, 2019, pp. 1–7.

[115] "MURAB project," https://www.murabproject.eu, accessed: 2021-09-04.

[116] M. K. Welleweerd, D. Pantelis, A. G. de Groot, F. J. Siepel, and S. Stramigioli, "Robot-assisted ultrasound-guided biopsy on mr-detected breast lesions," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 2965–2971.

[117] A. V. Nikolaev, L. de Jong, G. Weijers, V. Groenhuis, R. M. Mann, F. J. Siepel, B. M. Maris, S. Stramigioli, H. H. Hansen, and C. L. de Korte, "Quantitative evaluation of an automated cone-based breast ultrasound scanner for mri–3d us image fusion," *IEEE transactions on medical imaging*, vol. 40, no. 4, pp. 1229–1239, 2021.

[118] T. S. Pheiffer, R. C. Thompson, D. C. Rucker, A. L. Simpson, and M. I. Miga, "Model-based correction of tissue compression for tracked ultrasound in soft tissue image-guided surgery," *Ultrasound in medicine & biology*, vol. 40, no. 4, pp. 788–803, 2014.

[119] M. R. Burcher, L. Han, and J. A. Noble, "Deformation correction in ultrasound images using contact force measurements," in *Proceedings IEEE Workshop on Mathematical Methods in Biomedical Image Analysis (MMBIA 2001)*. IEEE, 2001, pp. 63–70.

[120] J. Dahmani, Y. Petit, and C. Laporte, "Model-based correction of ultrasound image deformations due to probe pressure," in *Medical Imaging 2017: Image Processing*, vol. 10133. International Society for Optics and Photonics, 2017, p. 101331D.

[121] R. Guo, G. Lu, B. Qin, and B. Fei, "Ultrasound imaging technologies for breast cancer detection and management: A review," *Ultrasound in medicine & biology*, 2017.

[122] B. Flack, M. Makhinya, and O. Goksel, "Model-based compensation of tissue deformation during data acquisition for interpolative ultrasound simulation," in *2016 IEEE 13th International Symposium on Biomedical Imaging (ISBI)*, April 2016, pp. 502–505.

[123] Z. Jiang, Y. Zhou, Y. Bi, M. Zhou, T. Wendler, and N. Navab, "Deformation-aware robotic 3d ultrasound," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7675–7682, 2021.

[124] K. Petrinec, E. Savitsky, and D. Terzopoulos, "Patient-specific interactive simulation of compression ultrasonography," in *2014 IEEE 27th International Symposium on Computer-Based Medical Systems*, May 2014, pp. 113–118.

[125] A. M. Priester, S. Natarajan, and M. O. Culjat, "Robotic ultrasound systems in medicine," *IEEE transactions on ultrasonics, ferroelectrics, and frequency control*, vol. 60, no. 3, pp. 507–523, 2013.

[126] Q. Huang, J. Lan, and X. Li, "Robotic arm based automatic ultrasound scanning for three-dimensional imaging," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 2, pp. 1173–1182, 2018.

[127] S. Virga, R. Göbl, M. Baust, N. Navab, and C. Hennersperger, "Use the force: deformation correction in robotic 3d ultrasound," *International journal of computer assisted radiology and surgery*, vol. 13, no. 5, pp. 619–627, 2018.

[128] E. O'Flynn, A. Wilson, and M. Michell, "Image-guided breast biopsy: state-of-the-art," *Clinical radiology*, vol. 65, no. 4, pp. 259–270, 2010.

[129] V. Groenhuis, E. Tagliabue, M. K. Welleweerd, F. J. Siepel, J. M. Osorio, B. M. Maris, U. Zimmermann, P. Fiorini, and S. Stramigioli, "Deformation compensation in robotically-assisted breast biopsy," in *Proc. 11th Int. Conf. Inf. Process. Comput. Assist. Interv.(IPCAI)*, 2020, p. 4.

[130] L. Han, J. Hipwell, T. Mertzanidou, T. Carter, M. Modat, S. Ourselin, and D. Hawkes, "A hybrid fem-based method for aligning prone and supine images for image guided breast surgery," in *2011 IEEE international symposium on biomedical imaging: from nano to macro*. IEEE, 2011, pp. 1239–1242.

[131] P. Pathmanathan, D. Gavaghan, J. Whiteley, M. Brady, M. Nash, P. Nielsen, and V. Rajagopal, "Predicting tumour location by simulating large deformations of the breast using a 3d finite element model and nonlinear elasticity," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2004, pp. 217–224.

[132] F. S. Azar, D. N. Metaxas, and M. D. Schnall, "A deformable finite element model of the breast for predicting mechanical deformations under external perturbations," *Academic Radiology*, vol. 8, no. 10, pp. 965–975, 2001.

[133] G. Dufaye, A. Cherouat, J.-M. Bachmann, and H. Borouchaki, "Advanced finite element modelling for the prediction of 3d breast deformation," *European Journal of Computational Mechanics/Revue Européenne de Mécanique Numérique*, vol. 22, no. 2-4, pp. 170–182, 2013.

[134] M. Eder, S. Raith, J. Jalali, A. Volf, M. Settles, H.-G. Machens, and L. Kovacs, "Comparison of different material models to simulate 3-d breast deformations using finite element analysis," *Annals of biomedical engineering*, vol. 42, no. 4, pp. 843–857, 2014.

[135] L. Han, J. H. Hipwell, B. Eiben, *et al.*, "A nonlinear biomechanical model based registration method for aligning prone and supine mr breast images," *IEEE transactions on medical imaging*, vol. 33, no. 3, pp. 682–694, 2013.

[136] P. A. Yushkevich, J. Piven, H. Cody Hazlett, R. Gimpel Smith, S. Ho, J. C. Gee, and G. Gerig, "User-guided 3D active contour segmentation of anatomical structures: Significantly improved efficiency and reliability," *Neuroimage*, vol. 31, no. 3, pp. 1116–1128, 2006.

[137] P. Cignoni, M. Callieri, M. Corsini, M. Dellepiane, F. Ganovelli, and G. Ranzuglia, "MeshLab: an Open-Source Mesh Processing Tool," in *Eurographics Italian Chapter Conference*, V. Scarano, R. D. Chiara, and U. Erra, Eds. The Eurographics Association, 2008.

[138] A. Lasso, T. Heffter, A. Rankin, C. Pinter, T. Ungi, and G. Fichtinger, "Plus: open-source toolkit for ultrasound-guided intervention systems," *IEEE Transactions on Biomedical Engineering*, vol. 61, no. 10, pp. 2527–2537, 2014.

[139] A. Fedorov, R. Beichel, J. Kalpathy-Cramer, J. Finet, J.-C. Fillion-Robin, S. Pujol, C. Bauer, D. Jennings, F. Fennessy, M. Sonka, J. Buatti, S. Aylward, J. Miller, S. Pieper, and R. Kikinis, "3d slicer as an image computing platform for the quantitative imaging network," *Magnetic resonance imaging*, vol. 30, no. 9, pp. 1323–1341, 2012.

[140] F. Visentin, V. Groenhuis, B. Maris, D. Dall'Alba, F. Siepel, S. Stramigioli, and P. Fiorini, "Iterative simulations to estimate the elastic properties from a series of mri images followed by mri-us validation," *Medical & biological engineering & computing*, pp. 1–12, 2018.

[141] S.-Y. Selmi, E. Promayon, J. Sarrazin, and J. Troccaz, "3d interactive ultrasound image deformation for realistic prostate biopsy simulation," in *Biomedical Simulation*, F. Bello and S. Cotin, Eds. Cham: Springer International Publishing, 2014, pp. 122–130.

[142] D. Baraff, "Non-penetrating rigid body simulation," *State of the art reports*, 1993.

[143] ——, "Analytical methods for dynamic simulation of non-penetrating rigid bodies," in *Proceedings of the 16th annual conference on Computer graphics and interactive techniques*, 1989, pp. 223–232.

[144] C. Duriez, C. Andriot, and A. Kheddar, "Signorini's contact model for deformable objects in haptic simulations," in *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, vol. 4. IEEE, 2004, pp. 3232–3237.

[145] Y. Saad, *Iterative methods for sparse linear systems*. SIAM, 2003.

[146] C. Duriez, C. Guébert, M. Marchal, S. Cotin, and L. Grisoni, "Interactive simulation of flexible needle insertions based on constraint models," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2009, pp. 291–299.

[147] K. Miller and J. Lu, "On the prospect of patient-specific biomechanics without patient-specific properties of tissues," *Journal of the mechanical behavior of biomedical materials*, vol. 27, pp. 154–166, 2013.

[148] F. Faure, C. Duriez, H. Delingette, J. Allard, B. Gilles, S. Marchesseau, H. Talbot, H. Courtecuisse, G. Bousquet, I. Peterlik, *et al.*, "Sofa: A multi-model framework for interactive physical simulation," in *Soft tissue biomechanical modeling for computer assisted surgery*. Springer, 2012, pp. 283–321.

[149] "NVIDIA gameworks. Nvidia FleX," https://developer.nvidia.com/flex, 2018, [Accessed: 2018-12-20].

[150] R. Amini, J. Z. Kartchner, L. A. Stolz, D. Biffar, A. J. Hamilton, and S. Adhikari, "A novel and inexpensive ballistic gel phantom for ultrasound training," *World journal of emergency medicine*, vol. 6, no. 3, p. 225, 2015.

[151] M. Mitchell, *An introduction to genetic algorithms*. MIT press, 1998.

[152] C. Audet and J. E. Dennis Jr, "Analysis of generalized pattern searches," *SIAM Journal on optimization*, vol. 13, no. 3, pp. 889–903, 2002.

[153] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *Medical Image Computing and Computer Assisted Intervention*, pp. 234–241, 2015.

[154] C. Rodero, P. Real, P. Zuñeda, C. Monteagudo, M. Lozano, and I. García-Fernández, "Characterisation of position based dynamics for elastic materials," in *Proceedings of the XXVI Spanish Computer Graphics Conference*. Eurographics Association, 2016, pp. 49–57.

[155] F. Liu, Z. Li, Y. Han, J. Lu, F. Richter, and M. C. Yip, "Real-to-sim registration of deformable soft tissue with position-based dynamics for surgical robot autonomy," *arXiv preprint arXiv:2011.00800*, 2020.

[156] J. Y. Wu, P. Kazanzides, and M. Unberath, "Leveraging vision and kinematics data to improve realism of biomechanic soft tissue simulation for robotic surgery," *Int. journ. of computer assisted radiology and surgery*, vol. 15, no. 5, pp. 811–818, 2020.

[157] F. Galbusera, A. Cina, M. Panico, D. Albano, and C. Messina, "Image-based biomechanical models of the musculoskeletal system," *European radiology experimental*, vol. 4, no. 1, pp. 1–13, 2020.

[158] I. Peterlik, N. Haouchine, L. Ručka, and S. Cotin, "Image-driven stochastic identification of boundary conditions for predictive simulation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2017, pp. 548–556.

[159] S. Nikolaev and S. Cotin, "Estimation of boundary conditions for patient-specific liver simulation during augmented surgery," *International Journal of Computer Assisted Radiology and Surgery*, 2020.

[160] I. Peterlik, H. Courtecuisse, C. Duriez, and S. Cotin, "Model-based identification of anatomical boundary conditions in living tissues," in *International Conference on Information Processing in Computer-Assisted Interventions*. Springer, 2014, pp. 196–205.

[161] Z. A. Taylor, M. Cheng, and S. Ourselin, "High-speed nonlinear finite element analysis for surgical simulation using graphics processing units," *IEEE transactions on medical imaging*, vol. 27, no. 5, pp. 650–663, 2008.

[162] M. Ovsjanikov, M. Ben-Chen, J. Solomon, A. Butscher, and L. Guibas, "Functional maps: a flexible representation of maps between shapes," *ACM Transactions on Graphics (TOG)*, vol. 31, no. 4, pp. 1–11, 2012.

[163] S. Melzi, J. Ren, E. Rodolà, A. Sharma, P. Wonka, and M. Ovsjanikov, "Zoomout: spectral upsampling for efficient shape correspondence," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 6, p. 155, 2019.

[164] N. Alkhouli, J. Mansfield, E. Green, J. Bell, B. Knight, N. Liversedge, J. C. Tham, R. Welbourn, A. C. Shore, K. Kos, *et al.*, "The mechanical properties of human adipose tissues and their relationships to the structure and composition of the extracellular matrix," *American Journal of Physiology-Endocrinology and Metabolism*, vol. 305, no. 12, pp. E1427–E1435, 2013.

[165] A. Mendizabal, E. Tagliabue, J.-N. Brunet, D. Dall'Alba, P. Fiorini, and S. Cotin, "Physics-based deep neural network for real-time lesion tracking in ultrasound-guided breast biopsy," *Computational Biomechanics for Medicine Workshop MICCAI*, 2019.

[166] A. A. Taha and A. Hanbury, "Metrics for evaluating 3d medical image segmentation: analysis, selection, and tool," *BMC medical imaging*, vol. 15, no. 1, p. 29, 2015.

[167] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *International Conference on Learning Representations*, 2019. [Online]. Available: https://openreview.net/forum?id=Bkg6RiCqY7

[168] L. N. Smith, "Cyclical learning rates for training neural networks," in *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2017, pp. 464–472.

[169] A. Roberti, N. Piccinelli, D. Meli, R. Muradore, and P. Fiorini, "Improving rigid 3-d calibration for robotic surgery," *IEEE Transactions on Medical Robotics and Bionics*, vol. 2, no. 4, pp. 569–573, 2020.

[170] B. Benligiray, C. Topal, and C. Akinlar, "Stag: A stable fiducial marker system," *Image and Vision Computing*, vol. 89, pp. 158–169, 2019.

[171] N. Sharp and K. Crane, "A laplacian for nonmanifold triangle meshes," in *Computer Graphics Forum*, vol. 39. Wiley Online Library, 2020, pp. 69–80.

[172] M. Allan, J. Mcleod, C. C. Wang, J. C. Rosenthal, K. X. Fu, T. Zeffiro, W. Xia, Z. Zhanshi, H. Luo, X. Zhang, *et al.*, "Stereo correspondence and reconstruction of endoscopic data challenge," *arXiv preprint arXiv:2101.01133*, 2021.

[173] F. Williams, T. Schneider, C. Silva, D. Zorin, J. Bruna, and D. Panozzo, "Deep geometric prior for surface reconstruction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[174] F. Lalys and P. Jannin, "Surgical process modeling: a review," *International J CARS*, vol. 9, no. 3, pp. 495–511, 2014.

[175] T. D. Nagy, M. Takács, I. J. Rudas, and T. Haidegger, "Surgical subtask automation-soft tissue retraction," in *2018 IEEE 16th World Symposium on Applied Machine Intelligence and Informatics (SAMI)*. IEEE, 2018, pp. 55–60.

[176] A. Attanasio, B. Scaglioni, M. Leonetti, A. F. Frangi, W. Cross, C. S. Biyani, and P. Valdastri, "Autonomous tissue retraction in robotic assisted minimally invasive surgery–a feasibility study," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6528–6535, 2020.

[177] S. Patil and R. Alterovitz, "Toward automated tissue retraction in robot-assisted surgery," in *2010 IEEE Int. Conf. Robotics and Automation*. IEEE, 2010, pp. 2088–2094.

[178] R. Jansen, K. Hauser, N. Chentanez, *et al.*, "Surgical retraction of non-uniform deformable layers of tissue: 2d robot grasping and path planning," in *2009 IEEE/RSJ Int. Conf. Intell. Robots and Systems*. IEEE, 2009, pp. 4092–4097.

[179] J. Schulman, A. Gupta, S. Venkatesan, M. Tayson-Frederick, and P. Abbeel, "A case study of trajectory transfer through non-rigid registration for a simplified suturing scenario," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 4111–4117.

[180] T. Osa, K. Harada, N. Sugita, and M. Mitsuishi, "Trajectory planning under different initial conditions for surgical task automation by learning from demonstration," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 6507–6513.

[181] A. Murali, S. Sen, B. Kehoe, A. Garg, S. McFarland, S. Patil, W. D. Boyd, S. Lim, P. Abbeel, and K. Goldberg, "Learning by observation for surgical subtasks: Multilateral cutting of 3d viscoelastic and 2d orthotropic tissue phantoms," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 1202–1209.

[182] H. Su, A. Mariani, S. E. Ovur, A. Menciassi, G. Ferrigno, and E. De Momi, "Toward teaching by demonstration for robot-assisted minimally invasive surgery," *IEEE Transactions on Automation Science and Engineering*, 2021.

[183] A. Hussein, M. M. Gaber, E. Elyan, and C. Jayne, "Imitation learning: A survey of learning methods," *ACM Computing Surveys (CSUR)*, vol. 50, no. 2, pp. 1–35, 2017.

[184] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1238–1274, 2013.

[185] J. Garcıa and F. Fernández, "A comprehensive survey on safe reinforcement learning," *Journal of Machine Learning Research*, vol. 16, no. 1, pp. 1437–1480, 2015.

[186] R. Cheng, G. Orosz, R. M. Murray, and J. W. Burdick, "End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 3387–3395.

[187] N. D. Nguyen, T. Nguyen, S. Nahavandi, A. Bhatti, and G. Guest, "Manipulating soft tissues by deep reinforcement learning for autonomous robotic surgery," in *2019 IEEE International Systems Conference (SysCon)*. IEEE, 2019, pp. 1–7.

[188] B. Thananjeyan, A. Garg, S. Krishnan, C. Chen, L. Miller, and K. Goldberg, "Multilateral surgical pattern cutting in 2d orthotropic gauze with deep reinforcement learning policies for tensioning," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 2371–2378.

[189] C. Shin, P. W. Ferguson, S. A. Pedram, *et al.*, "Autonomous tissue manipulation via surgical robot using learning based model predictive control," in *2019 Int. Conf. Robotics and Automation (ICRA)*. IEEE, 2019, pp. 3875–3881.

[190] S. A. Pedram, P. W. Ferguson, C. Shin, A. Mehta, E. P. Dutson, F. Alambeigi, *et al.*, "Toward synergic learning for autonomous manipulation of deformable tissues via surgical robots: An approximate q- approach," in *2020 8th IEEE RAS/EMBS Int. Conf. for Biomedical Robotics and Biomechatronics (BioRob)*. IEEE, 2019, pp. 878–884.

[191] A. Boeing and T. Bräunl, "Leveraging multiple simulators for crossing the reality gap," in *2012 12th International Conference on Control Automation Robotics & Vision (ICARCV)*. IEEE, 2012, pp. 1113–1119.

[192] F. Richter, R. K. Orosco, and M. C. Yip, "Open-sourced reinforcement learning environments for surgical robotics," *arXiv preprint arXiv:1903.02090*, 2019.

[193] J. Xu, B. Li, B. Lu, Y.-H. Liu, Q. Dou, and P.-A. Heng, "Surrol: An open-source reinforcement learning centered and dvrk compatible platform for surgical robot learning," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021.

[194] A. Nair, B. McGrew, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Overcoming exploration in reinforcement learning with demonstrations," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 6292–6299.

[195] J. Ho and S. Ermon, "Generative adversarial imitation learning," in *Advances in neural information processing systems*, 2016, pp. 4565–4573.

[196] W. Chi, G. Dagnino, T. M. Kwok, A. Nguyen, D. Kundrat, M. E. Abdelaziz, C. Riga, C. Bicknell, and G.-Z. Yang, "Collaborative robot-assisted endovascular catheterization with generative adversarial imitation learning," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 2414–2420.

[197] "ROS - Robot Operating System," https://www.ros.org/, accessed: 2021-12-10.

[198] "Unity Technologies," https://unity.com/, accessed: 2021-11-03.

[199] A. Juliani, V.-P. Berges, E. Vckay, Y. Gao, H. Henry, M. Mattar, and D. Lange, "Unity: A general platform for intelligent agents," *arXiv preprint arXiv:1809.02627*, 2018.

[200] L. Qian, A. Deguet, and P. Kazanzides, "dVRK-XR: Mixed Reality Extension for da Vinci Research Kit," in *Hamlyn Symposium on Medical Robotics*, 2019.

[201] A. S. Polydoros and L. Nalpantidis, "Survey of model-based reinforcement learning: Applications on robotics," *Journal of Intelligent & Robotic Systems*, vol. 86, no. 2, pp. 153–173, 2017.

[202] A. Nagabandi, G. Kahn, R. S. Fearing, and S. Levine, "Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 7559–7566.

[203] J. Schrittwieser, I. Antonoglou, T. Hubert, K. Simonyan, L. Sifre, S. Schmitt, A. Guez, E. Lockhart, D. Hassabis, T. Graepel, *et al.*, "Mastering atari, go, chess and shogi by planning with a learned model," *Nature*, vol. 588, no. 7839, pp. 604–609, 2020.

[204] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[205] M. F. Dixon, I. Halperin, and P. Bilokon, "Inverse reinforcement learning and imitation learning," in *Machine Learning in Finance*. Springer, 2020, pp. 419–517.

[206] M. Bain and C. Sammut, "A framework for behavioural cloning." in *Machine Intelligence 15*, 1995, pp. 103–129.

[207] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, "Generative adversarial networks: An overview," *IEEE Signal Processing Magazine*, vol. 35, no. 1, pp. 53–65, 2018.

[208] M. Pfeiffer, I. Funke, M. R. Robu, S. Bodenstedt, L. Strenger, S. Engelhardt, T. Roß, M. J. Clarkson, K. Gurusamy, B. R. Davidson, *et al.*, "Generating large labeled data sets for laparoscopic image processing tasks using unpaired image-to-image translation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2019, pp. 119–127.

[209] D. Rivoir, M. Pfeiffer, R. Docea, F. Kolbinger, C. Riediger, J. Weitz, and S. Speidel, "Long-term temporally consistent unpaired video translation from simulated surgical 3d data," *arXiv preprint arXiv:2103.17204*, 2021.

[210] F. Ingrand and M. Ghallab, "Deliberation for autonomous robots: A survey," *Artificial Intelligence*, vol. 247, pp. 10–44, 2017.

[211] M.-C. Fiazza and P. Fiorini, "Design for interpretability: Meeting the certification challenge for surgical robots," in *2021 Int. Conf. Intelligence and Safety for Robotics (ISR)*. IEEE, 2021.

[212] "Regulation of the european parliament and of the council laying down harmonised rules on artificial intelligence (artificial intelligence act) and amending certain union legislative acts," accessed: 2021-11-08. [Online]. Available: https://eur-lex.europa.eu/resource.html?uri=cellar:e0649735-a372-11eb-9585-01aa75ed71a1.0001.02/DOC_1&format=PDF

[213] G.-Z. Yang, J. Cambias, K. Cleary, E. Daimler, J. Drake, P. E. Dupont, N. Hata, P. Kazanzides, S. Martel, R. V. Patel, *et al.*, "Medical robotics—regulatory, ethical, and legal considerations for increasing levels of autonomy," *Science Robotics*, vol. 2, no. 4, p. 8638, 2017.

[214] T. D. Nagy and T. Haidegger, "A dvrk-based framework for surgical subtask automation," *Acta Polytechnica Hungarica*, pp. 61–78, 2019.

[215] S. Sen *et al.*, "Automating multi-throw multilateral surgical suturing with a mechanical needle guide and sequential convex optimization," in *2016 IEEE Int. Conf. Robotics and Automation (ICRA)*. IEEE, 2016, pp. 4178–4185.

[216] A. Shademan, R. S. Decker, J. D. Opfermann, S. Leonard, A. Krieger, and P. C. Kim, "Supervised autonomous robotic soft tissue surgery," *Science translational medicine*, vol. 8, no. 337, 2016.

[217] F. Liu *et al.*, "Real-to-sim registration of deformable soft tissue with position-based dynamics for surgical robot autonomy," in *2021 IEEE Int. Conf. on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 12 328–12 334.

[218] F. Alambeigi, Z. Wang, R. Hegeman, Y.-H. Liu, and M. Armand, "A robust data-driven approach for online learning and manipulation of unmodeled 3-d heterogeneous compliant objects," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 4140–4147, 2018.

[219] G. De Rossi *et al.*, "Cognitive robotic architecture for semi-autonomous execution of manipulation tasks in a surgical environment," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 7827–7833.

[220] B. Kehoe *et al.*, "Autonomous multilateral debridement with the raven surgical robot," in *2014 IEEE Int. Conf. Robotics and Automation (ICRA)*. IEEE, 2014, pp. 1432–1439.

[221] P. Berthet-Rayne *et al.*, "Hubot: A three state human-robot collaborative framework for bimanual surgical tasks based on learned models," *2016 IEEE Int. Conf. Robotics and Automation (ICRA)*, pp. 715–722, 2016.

[222] M. Bratman, *Intention, Plans, and Practical Reason*. Cambridge: Cambridge, MA: Harvard University Press, 1987.

[223] M. P. Georgeff and A. L. Lansky, "Reactive reasoning and planning." in *AAAI*, vol. 87, 1987, pp. 677–682.

[224] M. E. Bratman *et al.*, "Plans and resource-bounded practical reasoning," *Computational intelligence*, vol. 4, no. 3, pp. 349–355, 1988.

[225] E. Bolisani and C. Bratianu, "Knowledge strategy planning: an integrated approach to manage uncertainty, turbulence, and dynamics," *Journal of Knowledge Management*, 2017.

[226] M. Tenorth and M. Beetz, "Knowrob – a knowledge processing infrastructure for cognition-enabled robots," *International Journal of Robotics Research (IJRR)*, vol. 32, no. 5, pp. 566–590, 2013.

[227] H. Nakawala, R. Bianchi, L. E. Pescatori, O. De Cobelli, G. Ferrigno, and E. De Momi, ""Deep-Onto" network for surgical workflow and context recognition," *International journal of computer assisted radiology and surgery*, vol. 14, no. 4, pp. 685–696, 2019.

[228] D. Katić, A.-L. Wekerle, F. Gärtner, H. Kenngott, B. P. Müller-Stich, R. Dillmann, and S. Speidel, "Ontology-based prediction of surgical events in laparoscopic surgery," in *Medical Imaging 2013: Image-Guided Procedures, Robotic Interventions, and Modeling*, vol. 8671. International Society for Optics and Photonics, 2013, p. 86711A.

[229] M. Ginesi, D. Meli, H. Nakawala, A. Roberti, and P. Fiorini, "A knowledge-based framework for task automation in surgery," in *2019 19th Int. Conf. Advanced Robotics (ICAR)*. IEEE, 2019, pp. 37–42.

[230] Y. Dimopoulos, B. Nebel, and J. Koehler, "Encoding planning problems in nonmonotonic logic programs," in *European Conference on Planning*. Springer, 1997, pp. 169–181.

[231] M. Ginesi, D. Meli, A. Roberti, N. Sansonetto, and P. Fiorini, "Autonomous task planning and situation awareness in robotic surgery," in *2020 IEEE Int. Conf. Intell. Robots and Systems (IROS)*. IEEE, 2020, pp. 3144–3150.

[232] R. Muradore, P. Fiorini, G. Akgun, D. E. Barkana, M. Bonfe, F. Boriero, *et al.*, "Development of a cognitive robotic system for simple surgical tasks," *International Journal of Advanced Robotic Systems*, vol. 12, no. 4, p. 37, 2015.

[233] V. Lifschitz, "Answer set programming and plan generation," *Artificial Intelligence*, vol. 138, no. 1-2, pp. 39–54, 2002.

[234] A. Colmerauer, "An introduction to prolog iii," in *Computational Logic*. Springer, 1990, pp. 37–79.

[235] J.-C. Latombe, *Robot motion planning*. Springer Science & Business Media, 2012, vol. 124.

[236] A. H. Qureshi, Y. Miao, A. Simeonov, and M. C. Yip, "Motion planning networks: Bridging the gap between learning-based and classical motion planners," *IEEE Transactions on Robotics*, vol. 37, no. 1, pp. 48–66, 2020.

[237] E. Tagliabue, A. Pore, D. Dall'Alba, E. Magnabosco, M. Piccinelli, and P. Fiorini, "Soft tissue simulation environment to learn manipulation tasks in autonomous robotic surgery," in *2020 IEEE International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020.

[238] A. Pore, D. Corsi, E. Marchesini, D. Dall'Alba, A. Casals, F. Alessandro, and P. Fiorini, "Safe reinforcement learning using formal verification for tissue retraction in autonomous robotic-assisted surgery," in *2021 IEEE/RSJ Int. Conf. Intell. Robots and Systems (IROS)*. IEEE, 2021, pp. 0–6.

[239] F. Calimeri, W. Faber, M. Gebser, G. Ianni, R. Kaminski, T. Krennwallner, N. Leone, M. Maratea, F. Ricca, and T. Schaub, "Asp-core-2 input language format," *Theory and Practice of Logic Programming*, vol. 20, no. 2, pp. 294–309, 2020.

[240] J. Wang, Q.-Y. Yu, W. Li, B.-R. Wang, and Z.-R. Zhou, "Influence of clamping stress and duration on the trauma of liver tissue during surgery operation," *Clinical Biomechanics*, vol. 43, pp. 58–66, 2017.

[241] M. Machin, J. Guiochet, H. Waeselynck, J.-P. Blanquart, M. Roy, and L. Masson, "Smof: A safety monitoring framework for autonomous systems," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 48, no. 5, pp. 702–715, 2016.

[242] S. B. Bhayani, "da vinci robotic partial nephrectomy for renal cell carcinoma: an atlas of the four-arm technique," *Journal of robotic surgery*, vol. 1, no. 4, pp. 279–285, 2008.

[243] M. Ginesi, N. Sansonetto, and P. Fiorini, "Overcoming some drawbacks of dynamic movement primitives," *Robotics and Autonomous Systems*, vol. 144, p. 103844, 2021.

[244] M. Bombieri, M. Rospocher, D. Dall'Alba, and P. Fiorini, "Automatic detection of procedural knowledge in robotic-assisted surgical texts," *International Journal of Computer Assisted Radiology and Surgery*, pp. 1–9, 2021.

[245] D. Meli, M. Sridharan, and P. Fiorini, "Inductive learning of answer set programs for autonomous surgical task planning," *Machine Learning*, pp. 1–25, 2021.

[246] D. Meli, P. Fiorini, and M. Sridharan, "Towards inductive learning of surgical task knowledge: A preliminary case study of the peg transfer task," *Procedia Computer Science*, vol. 176, pp. 440–449, 2020.

[247] Z. Cheng *et al.*, "Design and integration of electrical bio-impedance sensing in surgical robotic tools for tissue identification and display," *Frontiers in Robotics and AI*, vol. 6, p. 55, 2019.

[248] B. Lindemann, T. Müller, H. Vietz, N. Jazdi, and M. Weyrich, "A survey on long short-term memory networks for time series prediction," *Procedia CIRP*, vol. 99, pp. 650–655, 2021.

[249] R. C. Gonzales and R. E. Woods, "Digital image processing," 2002.

[250] S. Xie and Z. Tu, "Holistically-nested edge detection," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1395–1403.

[251] Y. Li, F. Richter, J. Lu, E. K. Funk, R. K. Orosco, J. Zhu, and M. C. Yip, "Super: A surgical perception framework for endoscopic tissue manipulation with surgical robotics," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2294–2301, 2020.

[252] J. Lu, A. Jayakumari, F. Richter, Y. Li, and M. C. Yip, "Super deep: A surgical perception framework for robotic tissue manipulation using deep learning for feature extraction," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 4783–4789.

[253] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[254] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 652–660.

[255] W. Liu, J. Sun, W. Li, T. Hu, and P. Wang, "Deep learning on point clouds and its application: A survey," *Sensors*, vol. 19, no. 19, p. 4188, 2019.

[256] A. Attanasio, C. Alberti, B. Scaglioni, N. Marahrens, A. F. Frangi, M. Leonetti, C. S. Biyani, E. De Momi, and P. Valdastri, "A comparative study of spatio-temporal u-nets for tissue segmentation in surgical robotics," *IEEE Transactions on Medical Robotics and Bionics*, vol. 3, no. 1, pp. 53–63, 2021.

[257] A. Leporini, E. Oleari, C. Landolfo, A. Sanna, A. Larcher, G. Gandaglia, N. Fossati, F. Muttin, U. Capitanio, F. Montorsi, *et al.*, "Technical and functional validation of a teleoperated multirobots platform for minimally invasive surgery," *IEEE Transactions on Medical Robotics and Bionics*, vol. 2, no. 2, pp. 148–156, 2020.

[258] A. Diodato, M. Brancadoro, G. De Rossi, H. Abidi, D. Dall'Alba, R. Muradore, G. Ciuti, P. Fiorini, A. Menciassi, and M. Cianchetti, "Soft robotic manipulator for improving dexterity in minimally invasive surgery," *Surgical innovation*, vol. 25, no. 1, pp. 69–76, 2018.

[259] Z. Cheng, K. L. Schwaner, D. Dall'Alba, P. Fiorini, and T. R. Savarimuthu, "An electrical bioimpedance scanning system for subsurface tissue detection in robot assisted minimally invasive surgery," *IEEE Transactions on Biomedical Engineering*, 2021.

[260] A. Calanca, E. Dimo, R. Vicario, P. Fiorini, M. Serpelloni, and G. Legnani, "Introducing series elastic links for affordable torque-controlled robots," *IEEE Robotics and Automation Letters*, vol. 4, no. 1, pp. 137–144, 2018.

[261] A. Sengupta, R. Lagneau, A. Krupa, E. Marchand, and M. Marchal, "Simultaneous tracking and elasticity parameter estimation of deformable objects," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 10 038–10 044.

[262] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1125–1134.

[263] K. Rao, C. Harris, A. Irpan, S. Levine, J. Ibarz, and M. Khansari, "Rl-cyclegan: Reinforcement learning aware simulation-to-real," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 157–11 166.