



HAL
open science

Identification and characterization of boundary conditions for patient specific biomechanical simulation

Sergei Nikolaev

► **To cite this version:**

Sergei Nikolaev. Identification and characterization of boundary conditions for patient specific biomechanical simulation. Modeling and Simulation. University of Strasbourg, 2021. English. NNT : . tel-03416443v1

HAL Id: tel-03416443

<https://inria.hal.science/tel-03416443v1>

Submitted on 5 Nov 2021 (v1), last revised 16 Nov 2022 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

ÉCOLE DOCTORALE _____ MSII_269_____

Laboratoire ICube

THÈSE présentée par :

Sergei NIKOLAEV

soutenue le : 15 Septembre 2021

pour obtenir le grade de : **Docteur de l'université de Strasbourg**

Discipline/S spécialité : Informatique

**Identification et caractérisation des
conditions aux limites pour des
simulations biomécaniques patient-
spécifiques**

THÈSE dirigée par :

M. COTIN Stéphane

Directeur de recherche, Inria, Université de Strasbourg

RAPPORTEURS :

M. BRIEU Mathias

Professeur, California State University, USA

M. CHAGNON Grégory

Professeur, Université Grenoble Alpes

AUTRES MEMBRES DU JURY :

Mme. BAHLOULI Nadia

Professeure, Université de Strasbourg

M. ELLE Ole Jacob

Directeur de recherche, Oslo University Hospital

DOCTORAL SCHOOL _____ MSII_269_____

ICube Laboratory

THESIS presented by :

Sergei NIKOLAEV

defended on the : 15th of September, 2021

to obtain the : **PhD degree at the University of Strasbourg**

Discipline/Specialty : Computer Science

**Identification and characterization
of boundary conditions for
patient specific biomechanical
simulation**

THESIS supervised by :

M. COTIN Stéphane

Research director, Inria, University of Strasbourg

REVIEWERS :

M. BRIEU Mathias

Professor, California State University, USA

M. CHAGNON Grégory

Professor, University Grenoble Alpes

OTHER MEMBERS OF THE JURY :

Mme. BAHLOULI Nadia

Professor, University of Strasbourg

M. ELLE Ole Jacob

Research director, Oslo University Hospital



Acknowledgements



First of all, I would like to extend my deepest gratitude to my supervisor Stéphane Cotin. Stéphane, thank you for considering my candidature on this PhD position with a bit untypical but interesting topic. And also many thanks for helping me to understand many aspects related to the complex process of scientific research.

I gratefully acknowledge the help of my informal supervisor Igor Peterlik. He supervised me during the first years of my PhD project. Thanks for the initial proposal of using Kalman filters for estimation purposes, creating Optimus software that implements data assimilation process, and giving me a lot of valuable information during the first years of my PhD study. I also would like to thank Hadrien Courtecuisse for giving me information about Lagrangian multipliers technique and participating in my PhD project during the first year, through finally I chose Kalman filtering approach as a more appropriate estimation technique.

I very much appreciate the members of my jury, namely Nadia Bahlouli, Grégory Chagnon, Mathias Brieu, and Ole Jacob Elle, for the manuscript revision and their participation in my thesis defense. In addition, I am grateful to the members of my thesis mid-defense commission: Axel Hutt, Simon Chatelin, and Stéphane Bordas. Axel, special thanks for giving a lot of advice related to Kalman filtering approach as well as showing interesting papers and books on the topic. Thanks also for the proofreading of my thesis. Simon, thank you for consulting me about biomechanical properties measurements and hyperelastic model behaviors. Also, I appreciate the idea with the phantom experiment related to my topic motivation. Moreover, special thanks to Stéphane for spending his time in my mid-thesis defense and helping to validate my work during the difficult coronavirus times.

I would like to extend my gratitude to Antoine Petit, Raffaella Trivisonne, and Nava Schulmann for their numerous advice, their help and corrections. Antoine's work showed me that there exist tools and solutions to perform complex markerless tracking of the visible area, which increases the possibilities for data assimilation process. Finally, it gives me an additional motivation to look deeper in my topic. And also I am extremely grateful for the corrections made for my thesis text. I would like to extend my sincere thanks to Raffaella for collaborative working on Optimus plugin, for the corrections she did for my presentations and, finally, for events she organized not only in the office, but also outside it. As well, I very much appreciate Nava for her contribution and remarks about my numerous experiments with Kalman filters and for the corrections, she did in my papers.

Also, I want to say many thanks to Jean-Nicolas Brunet, Omar Mohamed Boukhris, and Hugo Talbot for their help with SOFA framework software and its plugins. I wish to thank Jean-

Nicolas for the software module he created, which helped me perform more accurate simulation, and Omar for his components that allow performing images and point cloud processing. And also special thanks to Hugo for providing a lot of information about SOFA structure, various technical events, and help with publishing the Optimus software as an open source project.

As well, I gratefully acknowledge the help of Juan Manuel Verde for his commentaries about my thesis paper from more clinical point of view. And also for his assistance related to a real liver manipulation experiments and data processing including segmentation.

Additionally, I would like to extend my thanks to Michel Duprez for the commentaries he made for my text and to Nicolas Golsse for liver meshes with segmented ligaments. I would also like to acknowledge the assistance of Andrea Mendizabal, who helped with technical stuff needed for videoconferencing during my thesis defense, and Pierre Galmiche, who made corrections for my French abstract.

I am very grateful to other members of Mimesis (now MLMS) team for their numerous remarks, proposals, hints and collaborative work during my thesis project. I want to say thanks to the SOFA consortium for framework development, which allows using various physically based simulation models. Using it as a base, we created the Optimus plugin, and performed estimation procedures necessary for my work.

In addition to that, thanks also should go to the HiPerNav team and its members. The program create by the HiPerNav allows me to look on the liver surgery problem and augmented reality systems in various contexts and from different points of view. As well, I would like to say many thanks to Andrea Teaniti and Egidijus Pelanis for their interesting discussions and experimental results they obtained in Oslo.

I would like to acknowledge the assistance of the members from Deformetrica and FEBio communities for their freely available software, which was used in the atlas construction process and some aspects of biomechanical modeling.

Finally, I am extremely grateful to my family for maintaining me morally: my parents Nicolai and Luibov, my sister Elena, and my grandmother Zoia. Despite some disagreements and worries about my living in another country, this trip might not be possible without their warm words and support.

Thank you very much for everything you have done for me.



Résumé



Objectif: Le traitement du cancer du foie, qui est l'une des maladies les plus courantes dans le monde, s'effectue souvent par la résection hépatique. Les chirurgiens enlèvent complètement les tumeurs, tout en préservant autant de tissus sains que possible. La tendance principale est d'utiliser des procédures plus sûres avec moins de complications, comme les traitements mini-invasifs. Cependant, au cours de ces procédures, la zone de fonctionnement n'est pas visible directement. Par conséquent, les chirurgiens utilisent diverses modalités d'imagerie médicales, comme les caméras laparoscopiques 3D, mais dont le champ de vision est limité. Dans ce cas, une solution complémentaire consiste à utiliser un modèle qui simule le comportement de l'organe réel. Adapté à la partie visible, il prédit les positions des parties invisibles de la zone d'intérêt et montre les emplacements des tumeurs préalablement recalées.

Les conditions aux limites (CAL) sont l'un des éléments cruciaux de la description mécanique d'un système biologique. Pour décrire un problème mécanique, il est nécessaire d'établir ce qui se passe à la surface ou à la limite du domaine d'intérêt. Des CAL soigneusement définies améliorent grandement la capacité prédictive du modèle biomécanique du foie. Par conséquent, le but principal de ce travail est d'estimer les CAL du tissu hépatique.

Malheureusement, dans le cas du foie, les CAL sont principalement donnés par des ligaments, les vaisseaux sanguins et les organes environnants, dont les propriétés ne peuvent pas être mesurées de manière fiable. Quoi qu'il en soit, le comportement du foie peut être enregistré sur une caméra stéréo, une échographie ou une autre modalité, mais, en raison d'erreurs d'observation, il existe une incertitude dans les données obtenues. L'idée est donc de proposer une approche d'assimilation des données où la déformation du tissu hépatique est utilisée pour estimer les attaches d'organes.

Méthode: Nous proposons d'estimer les CAL en deux étapes. Dans la première étape, nous créons une approximation initiale généralisée des CAL. Dans la deuxième étape, nous corrigeons l'approximation basée sur des données spécifiques au patient obtenues à partir d'une modalité d'image utilisée en intervention chirurgicale.

Pour une première approximation, l'idée générale est d'utiliser un atlas statistique. Malgré le fait que l'atlas ne peut pas nous fournir les propriétés exactes des CAL pour un patient spécifique, il contribue toujours à réduire la zone d'intérêt couverte. Cela permet d'éviter de nombreux calculs redondants, ce qui est crucial pour le processus en temps réel. Généralement, l'atlas contient une forme moyenne qui est générée à partir de la base de données de plusieurs images hépatiques. Pour construire l'atlas, des structures anatomiques segmentées manuellement sur les modèles hépatiques sont utilisées. Les formes de la base de données sont enregistrées de manière déformable sur la forme moyenne, en utilisant une cartographie métrique difféomorphe. Puis les différences entre les éléments segmentés sont utilisées pour créer des informations statistiques sur l'emplacement des CAL. La procédure de recalage de l'atlas vers

le foie du patient utilise la même méthode, mais dans ce cas, la correspondance des foies est utilisée pour transférer les données statistiques sur le modèle de foie cible.

Par contre, pour corriger les CAL, il est nécessaire d'estimer les paramètres par simulation inverse. Pour atteindre cet objectif, nous proposons de modéliser les CAL avec des ressorts dont les raideurs sont des variables stochastiques et d'utiliser le filtre de Kalman pour estimation de ces paramètres. Le filtre consiste en un processus itératif, calculant les valeurs attendues en fonction des probabilités pour des données et une simulation par éléments finis qui décrit la déformation du foie. Les données initiales pour le filtre sont prises à partir de matériaux préopératoires, puis, pendant la procédure chirurgicale, les valeurs des paramètres sont corrigées. Afin de répondre aux exigences de la chirurgie, le filtre doit effectuer les calculs en temps réel. Par conséquent, nous utilisons un filtrage de Kalman d'ordre réduit, qui diminue la complexité de notre modèle, améliorant ainsi la vitesse de calcul.

Résultats: Nous avons réalisé des expériences synthétiques pour évaluer l'approche de filtrage. Plusieurs ressorts non linéaires ont émulé les CAL pour le modèle de foie. Pour manipuler le foie, une force périodique a été appliquée. L'approche par filtrage de Kalman corrige généralement les valeurs des paramètres estimés en fonction des informations obtenues à partir des données disponibles. Les paramètres de certains ressorts n'ont pas été bien estimés, mais dans les expériences de validation, le comportement du modèle de foie avec les CAL corrigés était généralement plus proche des données de base qu'avec les approximations initiales. L'erreur entre les différences maximales et moyennes était plus faible après la correction. Par conséquent, le processus de filtrage améliore l'approximation initiale pour simuler plus précisément l'influence des CAL sur le modèle de foie.

L'atlas a été créé à l'aide de 15 modèles de foie, et les positions de leurs ligaments respectifs ont été segmentées par un expert. Pour garantir la qualité de l'atlas, un modèle a été sélectionné pour validation. Des statistiques ont été calculées, puis l'atlas a été enregistré sur un modèle ciblé afin de comparer la position moyenne avec la position segmentée. Les résultats montrent une différence moyenne de quelques centimètres. Par conséquent, la moyenne statistique diffère de la position des ligaments segmentés, mais nous pensons pouvoir compenser cet écart en estimant des paramètres légèrement différents lors de l'étape de correction.

Finalement, nous avons réalisé une expérience basée sur des données réelles. Nous avons pris un cadavre humain et simulé la procédure chirurgicale. Nous avons attaché plusieurs marqueurs qui servent comme observations et données de validation. Pour effectuer la manipulation, on a utilisé des outils chirurgicaux laparoscopiques. Nous avons aussi enregistré ces manipulations à l'aide d'une caméra RGB-D. À partir de l'image tomographie par ordinateur, nous avons reconstruit le modèle volumique du foie ainsi que les positions des marqueurs. Pour générer les attaches de ressort, nous avons construit ensuite un atlas statistique à partir de 15 modèles et l'avons enregistré sur le maillage volumique. Après on a effectué un processus d'assimilation de données. Pour la validation, nous avons extrait des marqueurs d'une autre tomographie par ordinateur, où le foie est déformé. Nous comparons les cas où seuls les vaisseaux sanguins ont été fixés, avec estimation initiale des ligaments et avec estimation des paramètres des ligaments. Les résultats montrent une faible amélioration de la précision lorsqu'un modèle de ligament est ajouté et ensuite estimé, mais cela pourrait dépendre de questions supplémentaires comme les contraintes unilatérales que nous ne prenons pas en compte.



Abstract



Objective: A typical treatment for the liver cancer, which is one of the most common worldwide diseases, is hepatic resection. Surgeons completely remove the tumors, while preserving as much healthy tissue as possible. The main trend is to use safer procedures with fewer complications, such as minimally invasive treatment. However, during such procedures, there is no direct vision at the area under operation. Therefore, surgeons use various medical modalities, such as three-dimensional laparoscopic cameras, which limit their range of views. In this case, a complementary solution is to use a model that simulates the behavior of the real organ. Matched to the visible part, it predicts the positions of invisible parts for the area of interest and shows the locations of initially registered tumors.

Boundary conditions (BCs) are among the crucial elements in mechanical description of a biological system. To describe a mechanical problem, it is necessary to establish what is happening on the surface or boundary of the considered domain. Carefully defined BCs essentially improves the predictive capacity of the biomechanical model. Therefore, the main purpose of the work is to estimate the BCs of the liver tissue.

Unfortunately, in case of the liver, BCs are mainly given by ligaments, blood vessels, and surrounding organs, the properties of which cannot be measured reliably. Anyway, the behavior of the liver can be recorded on a stereo camera, ultrasound, or some other modality, but, due to observational errors, there is a high amount of uncertainty in the system. Therefore, the idea is to propose the data assimilation approach where the deformation of the liver tissue is used to estimate the organ attachments.

Method: We propose to estimate BCs in two steps. In the first step, we create generalized initial approximation of BCs. In the second step, we correct the approximation based on patient-specific data obtained from a modality used in surgical intervention.

For initial approximation, the general idea is to use a statistical atlas. Despite the fact that atlas cannot provide us with the exact properties of BCs in patient specific case, it still helps to reduce the covered area of interest. This helps to avoid a lot of redundant calculations, which is crucial in real-time context. Generally, the atlas contains a mean shape that is generated from a database of several liver images. To construct the atlas, manually segmented anatomical structures on the liver models are used. The database shapes are nonrigidly registered on the average shape, using diffeomorphic metric mapping. Then the differences between segmented elements are used to create statistical information about the location of BCs. The procedure of atlas to patient registration uses the same method, but in this case, the livers matching is used to transfer the statistical data on the target liver model.

On the other hand, to correct BCs, it is necessary to estimate parameters using an inverse simulation. To achieve this goal, we propose to model BCs as stochastic stiffness values for elastic springs and to use the Kalman filter for their estimation. The filter obtains the result as an iterative process, computing the expected values based on probabilities for given data

and for a finite element simulation that describes liver deformation. The initial data for the filter is taken from pre-operative materials, and then, during the surgical procedure, the values of parameters are corrected. In order to fulfill the requirements of surgery, the filter needs to perform computations in real time. Therefore, we are using the reduced order Kalman filtering, which decreases the complexity of our model, thus improving the computation speed.

Results: We performed synthetic experiments to assess the filtering approach. Several non-linear springs emulated BCs for the liver model. To manipulate the liver, a periodic force was applied. The Kalman filtering approach in general corrects the values of estimated parameters according to obtained information from available data. The parameters of some springs were not estimated well, but in validation experiments the behavior of the liver model with corrected BCs in general was closer to groundtruth data than with initial approximations. The error between maximal and average differences was smaller after correction. Therefore, filtering process improves the initial approximation to simulate the BCs influence for the liver model more accurately.

The atlas was created using 15 liver models, and their respective ligament positions were segmented by an expert. To ensure the quality of the atlas, one model was selected for validation purposes. The statistics were computed and then the atlas was registered on a target model to compare the average position with segmented one. The results show the average difference for a couple of centimeters. Therefore, the statistical average differs from the ground ligaments position, but we expect to compensate for this deviation by estimating slightly different parameters during the correction step.

Finally, we performed an experiment based on real data. We took a human cadaver and simulated the surgical procedure. Several markers attached to it served as observations and validation data. To perform the manipulation, laparoscopic surgical tools were used. The manipulations were recorded using an RGB-D camera. From the computed tomography image, we reconstructed the volume model of the liver as well as the positions of the markers. To generate the spring attachments, a statistical atlas from 15 models was constructed and registered to the volume mesh. Afterwards, a data assimilation process was performed. For validation, we extracted markers from another computer tomography scan, where the recorded liver is deformed. We compare the cases where only blood vessels were fixed with initial estimation of ligaments and with estimation of ligament parameters. The results show some improvement in the accuracy when a ligament model is added and then estimated. But such small rank of improvement might be due to additional issues such as unilateral constraints, which we have not considered yet.




Contents




| | |
|--|-------------|
| Contents | vii |
| List of figures | ix |
| List of tables | xiii |
| Abbreviations | xv |
| Notations | xvii |
| 1 Introduction | 1 |
| 1.1 Liver surgery | 1 |
| 1.2 Image guided systems for liver surgery | 5 |
| 1.2.1 IGS without preoperative data | 5 |
| 1.2.2 IGS based on preoperative data | 8 |
| 1.3 Guided surgery for other highly deformable organs | 13 |
| 1.4 The selected approach for a liver augmented reality | 14 |
| 1.5 Research work overview | 15 |
| 2 Biomechanics of liver and ligaments | 19 |
| 2.1 The fundamentals of liver modeling | 20 |
| 2.1.1 Liver anatomy | 20 |
| 2.1.2 Liver biomechanics | 23 |
| 2.1.3 Simulation of liver deformations | 24 |
| 2.1.4 The numerical simulation of liver | 27 |
| 2.1.5 The selected approach to simulate liver deformations | 30 |
| 2.2 Liver boundary conditions | 32 |
| 2.2.1 Importance of boundary conditions | 32 |
| 2.2.2 Theoretical description of boundary conditions | 33 |
| 2.2.3 Anatomy of liver boundary conditions | 34 |
| 2.2.4 Boundary conditions simplification | 37 |
| 2.3 The fundamentals of ligaments modeling | 37 |
| 2.3.1 Ligaments anatomy | 37 |

| | | |
|----------|--|------------|
| 2.3.2 | Biomechanics of ligaments | 38 |
| 2.3.3 | Ligaments simulation | 39 |
| 2.3.4 | The model for ligaments | 42 |
| 3 | Data assimilation and Kalman filter | 49 |
| 3.1 | Overview of the parameters estimation procedures | 50 |
| 3.2 | Kalman filtering approach and its extensions | 51 |
| 3.2.1 | Kalman filtering approach | 51 |
| 3.2.2 | Reduced-order unscented Kalman filter | 55 |
| 3.2.3 | Ensemble transform Kalman filter | 58 |
| 3.3 | Software module for data-driven simulation | 62 |
| 3.3.1 | Prediction procedure | 64 |
| 3.3.2 | Correction (Analysis) procedure | 65 |
| 3.4 | Estimation using Kalman filtering | 67 |
| 3.4.1 | Object stiffness estimation | 69 |
| 3.4.2 | Estimation of boundary conditions | 72 |
| 3.4.3 | Correction of boundary conditions | 78 |
| 4 | Estimation of liver boundary conditions | 83 |
| 4.1 | Boundary conditions identification | 84 |
| 4.2 | The estimation approach | 85 |
| 4.3 | Statistical atlas construction | 86 |
| 4.3.1 | Overview of the available approaches | 86 |
| 4.3.2 | Large deformation diffeomorphic metric mapping | 89 |
| 4.3.3 | Atlas construction procedure | 91 |
| 4.4 | Ligaments properties initialization | 92 |
| 4.5 | Preconditioning for data assimilation procedure | 93 |
| 4.5.1 | Conjugate gradients and system preconditioning | 94 |
| 4.5.2 | Preconditioned data assimilation process | 95 |
| 4.5.3 | Experiments with preconditioning | 96 |
| 4.6 | Constructed process for boundary conditions estimation | 99 |
| 5 | Estimation experiments with human livers | 103 |
| 5.1 | Estimation based on synthetic observations | 103 |
| 5.2 | Atlas construction results | 107 |
| 5.3 | Experiments with human liver | 111 |
| 5.4 | Discussion | 117 |
| 6 | Conclusion | 119 |
| | Summary in french | 121 |



List of Figures



| | | |
|------|---|----|
| 1.1 | Overview of liver tumors. The size, shape, number, and position of tumors vary from one patient to another. Images taken from [219]. | 2 |
| 1.2 | Laparoscope surgical theater for liver resection procedures. The display on the right-hand side shows an operating field. | 3 |
| 1.3 | Illustration of an augmented reality in surgery: a visual liver model is overlaid with the intraoperative view during open surgery. | 4 |
| 1.4 | Left: structured light emitter for object surface reconstruction. The description of emitter components used to generate a structured light pattern. Right: Example of a miniaturized structured light probe in vivo. Images are taken from [122]. | 7 |
| 1.5 | Left: architecture of the combined navigation system with an endoscope, ultrasound probe and magneto-optic trackers. Right: monitoring screen of the navigation system. It includes preoperative CT, US, and endoscopic view. Images are taken from [105]. | 11 |
| 1.6 | Left: main steps of a motion compensated SLAM approach. The approach sequentially in a loop improves the accuracy of the model and specifies the positions of the camera. Right: respiratory simulation for the liver organ. The motion of the region is tracked and the analytical respiration model is fitted to it. Images taken from [150]. | 12 |
| 1.7 | Overview of research areas that are covered by the HiPerNav project (https://hipernav.eu). | 16 |
| 2.1 | Liver anatomy overview with labeled lobes. | 20 |
| 2.2 | Overview of the liver segments that are commonly used for resection planning. | 21 |
| 2.3 | The functional units of liver shown in the hierarchical order. Image taken from [17] licensed under CC BY 4.0. | 22 |
| 2.4 | Liver parenchyma and tumors reconstructed from CT images. Since tumors have abnormal mechanical properties, they have different intensity on CT and MRI images, which is used to segment and reconstruct their shape. Image taken from [219]. | 23 |
| 2.5 | Overview of the liver constitutive law. Left: four normal livers (dotted lines) and one obstructive liver disease (solid line), taken from [22]. Right: the chart shows the nominal stress τ as a function of the stretch ratio λ , taken from [155]. | 24 |
| 2.6 | Linear and hyperelastic models behavior comparison for a bending deformation. For all cases, the same mesh, applied forces, and BCs were used. | 31 |
| 2.7 | Left: membrane stretching experiment. Original conditions and model parameters. Right: visual overview of membrane stretching results under modified conditions. | 32 |
| 2.8 | The difference between validation points for membrane stretched under original and modified conditions. The type of modification is given in the legend. | 33 |
| 2.9 | Liver ligaments anatomy. | 35 |
| 2.10 | Liver vascular system description. | 36 |
| 2.11 | Anatomy of a typical ligament. The internal fibrils and fibers are organized in a hierarchical structure. Image taken from [113]. | 39 |

| | | |
|------|--|----|
| 2.12 | Overview of the experimental porcine peritoneum stress-strain curve (left image) taken from [246] and the theoretical ligament constitutive law (right image) taken from [203]. The constitutive law of ligaments is very dependent on collagen. After the fibers untwisting and a short linear response, fiber elasticity rapidly reaches a limit where microscopic failures appear. | 40 |
| 2.13 | The elastic behavior of ligament for different types of deformations (E is the Young's modulus). The nonlinear tension stress is much larger than compression or shear stress. Image taken from [200]. | 40 |
| 2.14 | The selected experimental stress-strain curves related to the softest sample (left image) taken from [246] and the stiffest sample (right image) taken from [58] | 45 |
| 2.15 | MSM stretching experiment description. A thin rectangle shape modeled by FE with Neo-Hookean material and the estimated Young's modulus was used as a reference model (left image). The MSM with diagonal springs were constructed and the parameters were assimilated from the FE model (right image). | 46 |
| 2.16 | Comparison for stretching deformation. The average and maximal difference is computed for FEM with Neo-Hookean material and the indicated Young's modulus and MSM with assimilated springs coefficients. | 46 |
| 3.1 | The transformation of uncertainty. To transform the Gaussian distribution through a nonlinear system a set of sigma points have to be selected in space. The new distribution is reconstructed based on the transformed points. | 53 |
| 3.2 | State covariance transformation to a reduce order. Since state covariance matrix is symmetric, it looks like matrix spectral, eigen, or VDV decomposition. | 56 |
| 3.3 | Transformation of ensemble members through a nonlinear system. The probability distribution is reconstructed using Monte Carlo estimator applied to the transformed members. | 59 |
| 3.4 | Overview of the components implemented in the Optimus plugin. They can be classified into four groups. Filtering group holds implementation of different Kalman filtering approaches. Observation manager group contains solutions to process the available observations. State wrappers group contains classes to transfer between filtering and mechanical states. Auxiliary components handle some additional issues to launch the data-driven process. | 63 |
| 3.5 | C++ example to illustrate the main idea of the FilteringAnimationLoop component. | 64 |
| 3.6 | C++ example to illustrate the transformation of sigma points inside the state wrapper. | 65 |
| 3.7 | C++ example to illustrate the data transferring from a sigma point to a mechanical state. | 66 |
| 3.8 | C++ example to illustrate the computation of predicted observations. | 67 |
| 3.9 | C++ example to illustrate the verification of observations availability. | 67 |
| 3.10 | Overview of the experimental workflow. The process contains three steps. The first part relates to generating groundtruth observations and validation. Then, data assimilation process is performed to estimate the unknowns. Finally, the simulation based on estimated parameters is compared with the groundtruth one. | 68 |
| 3.11 | Left: Overview of the experimental model, attached observations, and applied force. Right: Estimation results of the Young's modulus for different filters. The meshes used to generate the groundtruth and to simulate the object deformations are the same. | 70 |
| 3.12 | The average difference for validation points between groundtruth and simulation results obtained for initial and estimated Young's moduli. The meshes used to generate the groundtruth and to simulate the object deformations are the same. | 70 |
| 3.13 | Estimation results of the Young's modulus for the reduced filter versions. The mesh used to generate the groundtruth is denser than the one used to simulate the object deformations. | 72 |
| 3.14 | The average difference for validation points between groundtruth and simulation results obtained for initial and estimated Young's moduli. The mesh used to generate the groundtruth is denser than the one used to simulate the object deformations. | 73 |
| 3.15 | Left: Overview of the experimental model, attached springs, observations, and applied force. Right: Estimation results of cubic springs coefficients for different types of filters. The meshes used to generate the groundtruth and to simulate the object deformations are the same. | 75 |

| | | |
|------|---|-----|
| 3.16 | <i>The average difference for validation points between groundtruth and simulation results obtained for initial and estimated spring coefficients. The meshes used to generate the groundtruth and to simulate the object deformations are the same.</i> | 75 |
| 3.17 | <i>Estimation results of cubic springs coefficients for the ETKF and ROUKF. The mesh used to generate the groundtruth is denser than the one used to simulate the object deformations.</i> | 77 |
| 3.18 | <i>The average difference for validation points between groundtruth and simulation results obtained for initial and estimated spring coefficients. The mesh used to generate the groundtruth is denser than the one used to simulate the object deformations.</i> | 77 |
| 3.19 | <i>Estimation results of springs coefficients after the "cutting procedure" for the ETKF and ROUKF. The mesh used to generate the groundtruth is denser than the one used to simulate the object deformations.</i> | 79 |
| 3.20 | <i>The average difference for validation points between groundtruth and simulation results. The initially estimated coefficients are compared with ones estimated after the "cutting procedure". The mesh used to generate the groundtruth is denser than the one used to simulate the object deformations.</i> | 80 |
| 4.1 | <i>General identification process of BCs. The BCs are presented as cubic springs with unknown parameters. The estimation workflow contains two steps. 1 - Initial approximation based on statistics from liver database and constitutive laws from the literature. 2 - Correction based on intraoperative images. Ligament constitutive law taken from [246].</i> | 86 |
| 4.2 | <i>Statistical atlas construction for ligaments location. A set of models (left part of image) with segmented ligaments (white strips between red and blue) are registered on a mean liver model (right part of image). A registered statistical atlas has expected ligament location (green strips) and a range of probable locations (white region).</i> | 91 |
| 4.3 | <i>Main steps of the atlas construction process and its registration on the target model. Green rectangles depict actions, while red ones describe correspondent data.</i> | 92 |
| 4.4 | <i>Main steps of the elastic parameters initialization process for ligaments. Green rectangles depict actions, while red ones describe correspondent data. Ligament constitutive law taken from [246].</i> | 93 |
| 4.5 | <i>Preconditioning for the transformation of uncertainty or the ensemble members transformation. The idea is to compute an inverse matrix for one of sigma points or an ensemble member and use it as a preconditioning in transformation for all other points.</i> | 95 |
| 4.6 | <i>The average difference for validation points between groundtruth and simulation results. The results are computed for the ETKF and ROUKF with and without preconditioning. The mesh used to generate the groundtruth is denser than the one used to simulate the object deformations.</i> | 98 |
| 4.7 | <i>Stochastic process workflow for estimation of boundary conditions. In the prediction step, the deformation related to next video frame is modeled based on given springs stiffness estimation. In the correction step, the values are corrected based on difference between simulated observations and real data.</i> | 100 |
| 5.1 | <i>Numerical simulation mimicking a manipulation of the liver, from which observations were extracted.</i> | 104 |
| 5.2 | <i>Left: Overview of generated observations attached to the liver anterior surface. Right: liver model created for AR with generated validation points. To observe the internal part, the liver model is made transparent.</i> | 105 |
| 5.3 | <i>Average difference for validation points between groundtruth and simulation results with estimated BCs (green curve) and between groundtruth and simulation with fixed BCs (blue curve). The x-axis shows iterations of the validation process.</i> | 106 |
| 5.4 | <i>Maximal difference for validation points between groundtruth and simulation results with estimated BCs (green curve) and between groundtruth and simulation with fixed BCs (blue curve). The x-axis shows iterations of the validation process.</i> | 106 |

| | | |
|------|---|-----|
| 5.5 | <i>The liver meshes used to construct the statistical atlas for ligaments, anterior view. White strips (between red and blue) represent annotated layers of falciform ligaments. Yellow meshes are round ligaments.</i> | 108 |
| 5.6 | <i>The liver mesh selected to validate the statistical atlas, anterior view. The two layers of the falciform ligament are segmented as white strips (between red and blue), and yellow mesh is a round ligament.</i> | 108 |
| 5.7 | <i>The annotated right layers (left image) and left layers (right image) of the falciform ligament taken from liver dataset and registered on the average shape. Each curve is marked as a set of points with a color from green-blue color map.</i> | 109 |
| 5.8 | <i>The annotated upper layers (left image) and lower layers (right image) of the coronary ligament with left and right triangular ones taken from liver dataset and registered on the average shape. Each curve is marked as a set of points with a color from green-blue color map.</i> | 109 |
| 5.9 | <i>Registration results overview for the right (left image) and left (right image) falciform ligament layers on the validation model. The mean (yellow dots) and standard deviation (violet dots) are superposed with manual segmentation (black dots).</i> | 110 |
| 5.10 | <i>Registration results overview for the upper (left image) and lower (right image) coronary layers, together with left and right triangular ligaments, on the validation model. The mean (yellow dots) and standard deviation (violet dots) are superposed with manual segmentation (black dots).</i> | 110 |
| 5.11 | <i>Overview of the operating field for the human liver experiment. To track the liver movements, a set of laparoscopic tackers were attached to its surface. To simplify segmentation, we added colored papers to the tackers. The manipulations were recorded with an Intel RealSense D435 RGB-D camera.</i> | 112 |
| 5.12 | <i>Left: Example of a laparoscopic tacker, which is generally used to attach artificial materials to the tissues. Right: Overview of an Intel RealSense D435 RGB-D camera. It has a color sensor to obtain an RGB image and an infrared (IR) projector with two IR sensors to reconstruct the scene depth.</i> | 112 |
| 5.13 | <i>Left: overview of the reconstructed laparoscopic tackers on the liver surface. Right: overview of the reconstructed deep fiducials labeled with numbers. To observe the internal parts, the liver model is made transparent.</i> | 113 |
| 5.14 | <i>Statistical average registration result for the falciform ligament on the human liver shown in superior (left image) and anterior (right image) views. The layers colored differently to clarify their registered positions.</i> | 113 |
| 5.15 | <i>Statistical average registration result for the upper (left image) and lower (right image) layers of the coronary ligament and both triangular ligaments on the human liver. The layers colored differently to clarify their registered positions.</i> | 114 |
| 5.16 | <i>Finite element model with the constructed mass-spring system shown in the superior (left image) and posterior (right image) views.</i> | 114 |
| 5.17 | <i>Visual comparison of two CT scans for the liver organ in its rest shape (left image) and deformed shape (right image).</i> | 115 |
| 5.18 | <i>TRE for the internal fiducials between the groundtruth and the simulation results in three scenarios. In the first scenario, only blood vessels as fixed BCs are modeled. In the second one, a mass-spring system is added with parameters estimated from pig peritoneum stress-strain curve. In the third one, a mass-spring system is added with spring parameters estimated by the preconditioned ROUKF. The fiducials labels are correspondent to ones in Fig. 5.13 right.</i> | 116 |



List of Tables



| | | |
|-----|--|-----|
| 2.1 | <i>Computation time of models with different hyperelastic materials for a bending deformation. For all cases, the same mesh, applied forces, and BCs were used.</i> | 31 |
| 2.2 | <i>Stretching simulation computation time. For FEM, a mesh with 3700 elements were taken, and MSM was composed of 6 edge springs and 12 diagonal ones (Fig. 2.15).</i> | 46 |
| 3.1 | <i>Young's modulus estimation for the case when the meshes used to generate the groundtruth and to simulate the object deformations are the same. Estimation results relative comparison and computation time for different filters.</i> | 71 |
| 3.2 | <i>Young's modulus estimation for the case when the mesh used to generate the groundtruth is denser than the one used to simulate the object deformations. Estimation results relative comparison and computation time for the ETKF and ROUKF.</i> | 73 |
| 3.3 | <i>Springs coefficients estimation for the case when the meshes used to generate the groundtruth and to simulate the object deformations are the same. Estimation results relative comparison and computation time for the ETKF and ROUKF.</i> | 76 |
| 3.4 | <i>Springs coefficients estimation for the case when the mesh used to generate the groundtruth is denser than the one used to simulate the object deformations. Estimation results relative comparison and computation time for the ETKF and ROUKF.</i> | 78 |
| 3.5 | <i>Springs coefficients estimation for the case when the mesh used to generate the groundtruth is denser than the one used to simulate the object deformations. Estimation results relative comparison for the ETKF and ROUKF in cases when spring parameters are corrected after the "cutting procedure" and not.</i> | 80 |
| 4.1 | <i>Springs coefficients estimation for the case when the mesh used to generate the groundtruth is denser than the one used to simulate the object deformations. Estimation results relative comparison and computation time for the ETKF and ROUKF with and without preconditioning.</i> | 99 |
| 5.1 | <i>The Target Registration Error between registered statistical average and manual segmentation for the given ligaments on the validation model.</i> | 111 |



Abbreviations



| Acronym | Description |
|----------------|--|
| 3D | = three-dimensional |
| 4D | = four-dimensional |
| AR | = augmented reality |
| BCs | = boundary conditions |
| BI | = Bayesian inference |
| CBCT | = cone beam computed tomography |
| CG | = Conjugate gradient method |
| CT | = computer tomography |
| DDDAS | = Dynamic Data Driven Applications System |
| ETKF | = ensemble transform Kalman filter |
| FE | = Finite Element |
| FEM | = Finite Element Method |
| ICP | = iterative closest point |
| IGS | = image-guided surgery |
| IR | = infrared |
| LDDMM | = large deformation diffeomorphic metric mapping |
| MRI | = magnetic resonance imaging |
| MSM | = Mass-Spring Model |
| PCA | = principal component analysis |
| ROUKF | = reduced-order unscented Kalman filter |
| SLAM | = simultaneous localization and mapping |
| StVK | = Saint Venant-Kirchhoff |
| TRE | = Target Registration Error |
| UKF | = unscented Kalman filter |
| US | = ultrasound |



Notations



Chapter 2

D object damping matrix

\mathcal{F} deformation gradient

\mathbf{c} Cauchy-Green (Cauchy) deformation (strain) tensor

\mathbf{u} element displacement

Ω region of interest

\mathbf{s} infinitesimal element

M object mass matrix

ζ_M reyleigh mass

ζ_K reyleigh stiffness

ξ Green-Lagrange strain tensor

β Cauchy stress tensor

t time

Φ transformation for region of interest

\mathcal{S} surface unit

V volume unit

\mathcal{B} description of the boundary domain

\mathbf{n} finite element mesh face normal

\mathbf{f} forces

I identity matrix

I_1 first principal scalar invariant of strain tensor
 I_2 second principal scalar invariant of strain tensor
 I_4 forth principal scalar invariant of strain tensor
 J deformation Jacobian
 μ_1 first Lamé coefficient
 μ_2 second Lamé coefficient
 \mathbf{R} rigid rotation matrix
 τ nominal stress
 Δl spring length variation
 $k_\alpha, k_\beta, k_\gamma$ spring stiffness parameters
 \mathbf{K} elastic stiffness matrix
 Ψ strain energy density function
 λ_i principal deformation stretches
 λ stretch ratio
 λ^* collagen stretch threshold
 E Young's modulus

Chapter 3

\mathbf{e} spatial canonical vector
 \mathcal{N} Gaussian distribution
 \mathbf{b} Kalman filtering control sequence
 J Kalman filtering cost function
 cov Kalman filtering cross covariance
 \mathcal{K} Kalman filtering gain
 \mathbf{v} Kalman filtering observations noise
 \mathbf{V} Kalman filtering observations noise covariance
 \mathcal{H} Kalman filtering observations operator

\mathbf{z} Kalman filtering observations
 $\tilde{\mathbf{z}}$ Kalman filtering predicted observations
 \mathbf{x} Kalman filtering state
 $\tilde{\mathbf{x}}$ Kalman filtering predicted state
 $\bar{\bar{\mathbf{x}}}$ Kalman filtering corrected state
 \mathbf{P} Kalman filtering state covariance
 $\tilde{\mathbf{P}}$ Kalman filtering predicted state covariance
 $\bar{\bar{\mathbf{P}}}$ Kalman filtering corrected state covariance
 N_x Kalman filtering state size
 $I^{(k)}$ Kalman filtering unitary sigma points
 \mathcal{T} Kalman filtering system process
 \mathbf{q} Kalman filtering system process noise
 \mathbf{Q} Kalman filtering system process noise covariance
 \mathbf{U} reduced-order Kalman filtering core
 \mathbf{L} reduced-order Kalman filtering transformation
 \mathbf{H} reduced-order Kalman filtering transformation for observations
 \mathbf{x}^σ Kalman filtering sigma point
 $\tilde{\mathbf{x}}^\sigma$ Kalman filtering predicted sigma point
 $\tilde{\mathbf{z}}^\sigma$ Kalman filtering predicted sigma point observation
 \mathbf{x}^{en} ensemble transform Kalman filtering member
 N_{en} ensemble transform Kalman filtering members amount
 $\tilde{\mathbf{x}}^{en}$ ensemble transform Kalman filtering predicted member
 $\tilde{\mathbf{z}}^{en}$ ensemble transform Kalman filtering predicted member observation
 $\bar{\bar{\mathbf{x}}}^{en}$ ensemble transform Kalman filtering corrected member
 $\tilde{\mathbf{X}}^{diff}$ ensemble transform Kalman filtering members difference
 $\tilde{\mathbf{Z}}^{diff}$ ensemble transform Kalman filtering observations difference

ω ensemble transform Kalman filtering state coefficients in ensemble space

ι ensemble transform Kalman filtering inflation

\mathcal{E} statistical mean

Chapter 4

s simple element

v element displacement

g diffeomorphic metric control point

N_p diffeomorphic metric control points amount

w diffeomorphic metric moments (weights)

Γ diffeomorphic metric interpolation kernel

η interpolation kernel width

l trajectory parametrization variable for diffeomorphic displacement

e mesh face center point

n mesh face normal

\mathcal{M} mesh of object

Φ^{diff} diffeomorphic transformation

\mathbf{e} approximation error

C hyperelastic material parameters

κ matrix condition number

\mathbf{C} preconditioner matrix




INTRODUCTION




Contents

| | |
|--|-----------|
| 1.1 Liver surgery | 1 |
| 1.2 Image guided systems for liver surgery | 5 |
| 1.2.1 IGS without preoperative data | 5 |
| 1.2.2 IGS based on preoperative data | 8 |
| 1.3 Guided surgery for other highly deformable organs | 13 |
| 1.4 The selected approach for a liver augmented reality | 14 |
| 1.5 Research work overview | 15 |

1.1 Liver surgery

The liver is the largest organ in the human body. It plays a fundamental role in several vital functions, which include detoxification of various metabolites, synthesizing of proteins, and production of biochemicals necessary for digestive and growth. Therefore, liver diseases might cause severe problems that lead to a variety of abnormalities and shorter life expectancy. Particularly, it is vulnerable to primary liver cancer, which is the fifth most common cancer worldwide and the third most common cause of cancer mortality [53]. The liver organ also suffers from other metastases, caused by illness such as colorectal cancer diseases. This includes approximately 16 % of all cases [110]. The cancer disease is characterized by appearing of abnormal cells that do not provide their ordinary functions. Tumors are abnormal tissues made of such cells. They have different size and shape and might appear in any part of the organ (Fig. 1.1).

Despite the fact that several therapies and a transplantation procedure are available to treat the hepatic cancers, liver resection remains the most common treatment for patients with such metastases (Fig. 1.2). This includes patient cases with intrahepatic cholangiocarcinoma and with hepatocellular carcinoma [57, 167]. The median survival for untreated patients is 36 months for early stage tumors and 3-16 months for more complex cases [57], while after resection the estimated 5-year survival rate is approximately 70 % [199]. A successful surgical resection requires complete removal of the tumor(s) while keeping as much healthy tissue as possible.

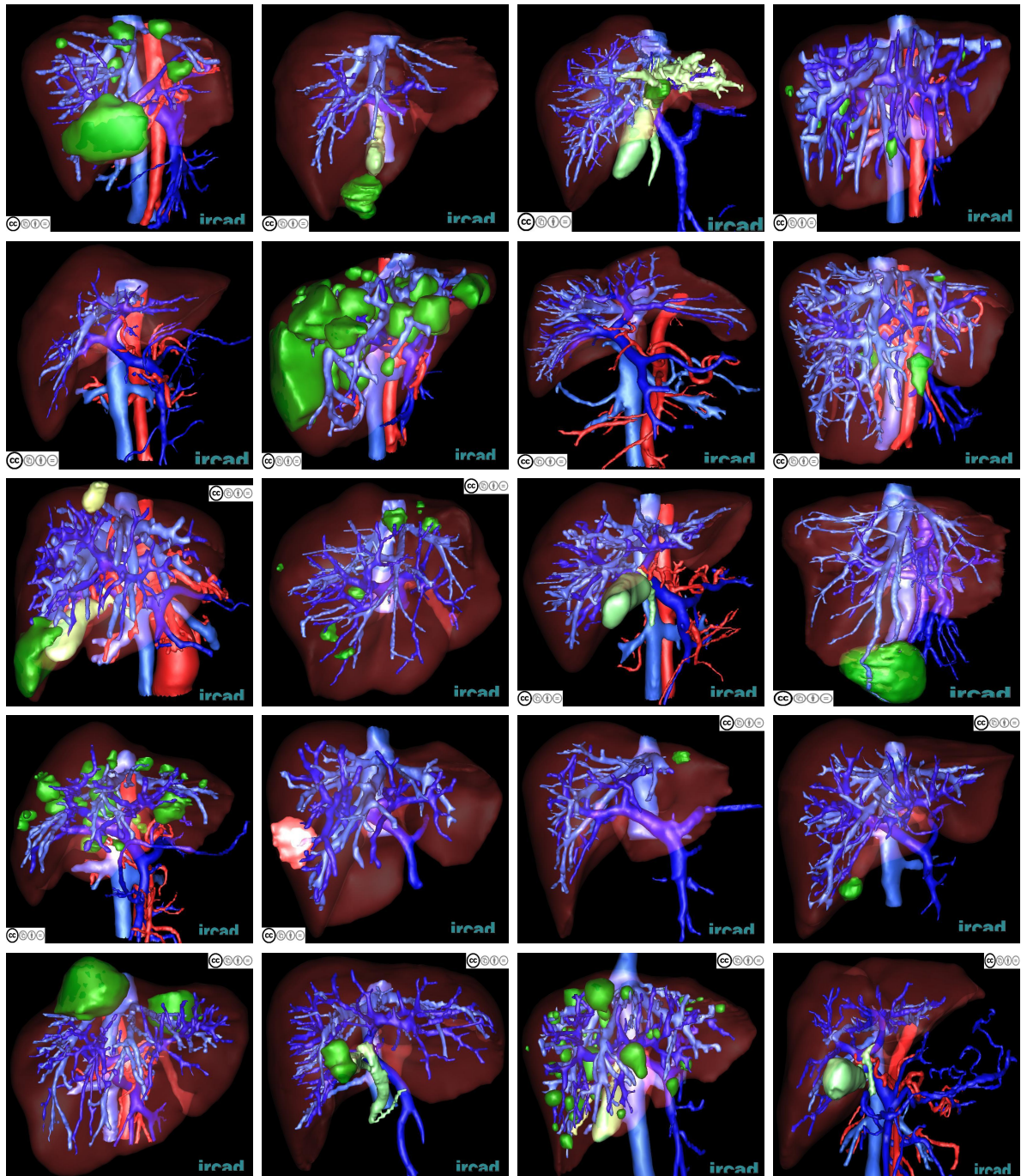


Figure 1.1: Overview of liver tumors. The size, shape, number, and position of tumors vary from one patient to another. Images taken from [219].

Also the vital liver parts like arteries and veins must be spared by the surgery or strategically divided. All these restrictions make resection procedures quite complex both from technical and clinical views. So it is important to help surgeons with additional tools that will simplify

the resection procedure and, therefore, increase the patient eligibility and improve the survival prognosis.



Figure 1.2: Laparoscope surgical theater for liver resection procedures. The display on the right-hand side shows an operating field.

Nowadays surgical rooms are equipped with different complex technologies. Various imaging modalities, including computer tomography (CT), magnetic resonance imaging (MRI), ultrasound (US) probes, and laparoscopic/stereoscopic cameras, allow surgeons looking inside the patient body and observe organs and tissues that are not visible directly. Besides that, the new laparoscopic tools help surgeons to perform treatment procedures for organs without having direct access to them.

Typically, before clinical procedures, for diagnosis and planning purposes, an initial CT scan is acquired, which is called preoperative CT, of the patient abdominal area. Based on the obtained data, the doctors localize the positions of tumors and then select the best treatment procedure.

In case of liver resection, two options are available: open or laparoscopic surgery. Open surgery is a more traditional procedure, when surgeons open abdomen of a patient and then eliminate tumors, having a direct access to a liver. The back part of the liver and internal liver structure, however, remain invisible. This simplifies eliminating tumors, close to vital liver parts and blood vessels, or which are hard to get access to, but such operations have certain consequences. The main problems for such type of surgery are that they have lower survival rate prognosis, higher risk of getting an infection with complications, the larger rehabilitation hospital time, and the big scar on the abdomen skin.

However, new technologies lead to new surgical procedures such as minimally invasive liver surgery. In this case, small holes in abdomen of patients are made, through which some trocars are placed. The surgeons then perform resection procedures by manipulating the laparoscopic tools inserted through the trocars and observe the operating field with surgical mini camera. The advantage is that for laparoscopic procedure patients have less hospital stays and fewer complications [31, 61]. But surgical modalities can provide surgeons only with limited amount

of view related to the disease and its problematic nature. And laparoscopic tools apply certain restrictions to manipulations.

To compensate for the lack of observability, the general tool that is currently used in surgery is image-guided surgery (IGS) systems. The main idea is to reconstruct the three-dimensional liver model with internal structures from preoperative CT images. During image analysis and preoperative planning, surgeons manually segment the tumors. The three-dimensional model can be then presented together with intraoperative view. Thus, during surgery, the clinicians can observe the liver model with marked tumors and approximately locate their positions on the real organ. In addition to that, the image-guided surgery system allows to mark the positions of surgical tools. However, such systems show only static images and do not depict the dynamic behavior of an organ. Also, the model does not take into account additional conditions like abdominal area inflation or different patient position compared with the preoperative scan. Therefore the position of a real tumor might vary significantly from the positions on the reconstructed model. Moreover, the model construction process is semi-automatic (or even manual) and, thus, requires a lot of human resources.

A described approach could be improved by providing an augmented reality (AR) view for the liver organ (Fig. 1.3). An augmented reality is an interactive demonstration of a real-world environment where the objects from the real world are enhanced by computer-generated perceptual information. This would be a nice addition. It not only provides an overlay of the three-dimensional liver model with the intraoperative view, but also performs a corresponding real-time deformation update. Therefore, surgeons will observe and track various structures such as tumors, blood vessels, and unobserved part of the organ. Consequently AR system will increase surgical accuracy and reduce the time of the operation.



Figure 1.3: Illustration of an augmented reality in surgery: a visual liver model is overlaid with the intraoperative view during open surgery.

The liver is a soft organ, which changes in shape not only because surgeons perform manipulations, but also due to respiratory and cardiac motions. Providing such a system during surgery, therefore, requires to solve several challenges. Firstly, the three-dimensional (3D) model of the virtual anatomy needs to be registered onto the real organ using an available intraoperative image data, which in general covers only small part of a surface. It has to take into account the

organ properties of a specific patient. It also needs to follow the actual tissue motion in real time, and it has to provide an accurate estimation of the location for the internal structures. In addition to that, it has to be integrated easily in most modern operating rooms, which requires no additional equipment, necessary for AR.

1.2 Image guided systems for liver surgery

The last decade of image-guided surgery, including liver surgery, has progressively become more popular due to various clinical benefits. Therefore creation of computer-aided tools became an attractive topic for scientific research. This resulted in increasing number of published works during the recent years. The proposed ideas are different in terms of applied methodology and numerical approaches. However, we think they can be classified into several groups. In this section we will give a general overview of them.

1.2.1 IGS without preoperative data

Generally, to understand the nature of a disease, an initial scan is acquired for the patient (usually computer tomography, but sometimes it might be magnetic resonance imaging), based on which the plan for liver treatment is created. These images describe the liver organ in good details. Thus, the idea for most of IGS methods is to use this data for providing additional information on intraoperative view. However, some approaches attempts to create an augmented reality view based only on intraoperative data.

The main advantage of such approaches is that they do not require preoperative data registration. This registration has several challenges since preoperative images, and intraoperative data are not in the same coordinate system. Moreover, they are taken for a patient in a different position.

1.2.1.1 IGS based on intraoperative CT or MRI

To obtain internal data intraoperatively, some approaches rely on a special equipment, such as intraoperative CT or MRI. Such devices allow acquiring an up-to-date three-dimensional view of the operating field. But they are very rare and not available in a typical hospital. Moreover, the quality of intraoperative scans is bounded because of the radiation doses issues (in case of CT) or operation time limitations (in case of MRI). In case of liver operation, surgeons also prefer to create a resection plan based on preoperative data. Therefore, it is quite important to register it for visualization during surgery, which can be done alongside with preoperative data fusion.

The main ideas are listed below.

Renewable view with CT. Some works use intraoperative CT to create a volumetric view. In the work [54], to get an augmented view, the main idea is to find the position of cone beam computed tomography (CBCT) and a laparoscope camera in the common world coordinate system. So, the researchers put retroreflective spherical markers on the laparoscope pointer and on flat-panel detector of the CBCT and use optical cameras for their simultaneous tracking.

In [55], the same authors extend their approach by acquiring additional intraoperative CT scan, which is done before an operation procedure, in order to get images with better quality. To register the new scan, a set of self-adhesive fiducials is attached to the patient skin. After CT acquisition, the laparoscope device is moved around each fiducial, to find a reference between retroreflective markers and fiducial positions.

The authors from [215] use intraoperative CT together with a laparoscopic camera. Initially, they perform a contrast-enhanced CT to get the volume view of the operation field, which is then followed by low-dose scans just to update the intraoperative view. The registration between computer tomography scans is done based on the algorithm from [238], while initial alignment with laparoscopic camera is done using Amira software [222]. To keep overlaying the laparoscopic images with CT data, the authors use an optical tracker and infrared markers, which are attached to the laparoscopic pointer.

Renewable view with MRI. To get the volume rendering, the researchers from [140, 148, 209, 230] rely on intraoperative MRI, which consequently generated images during operation. In [230], infrared markers attached to a laparoscopic camera and an optical device is used to track the camera movements. Additionally, some markers are put on patient abdomen skin to register MRI in optical tracking coordinates. The authors from [209] propose to show the several types of MRI images at the same time to simplify the needle navigation for ablation. In [140], the authors set markers on magnetic resonance imaging device and laparoscopic camera to register them in the global coordinate system. To avoid misregistration during respiratory motion, they use an optical displacement indicator, which follows the abdomen movement. The MRI image is updated with laparoscopic view only when certain thresholds in respiratory phases are achieved.

MRI based respiratory adjustment. Researchers from [227] perform a respiratory compensation with intraoperative MRI. Firstly, they propose to record the respiratory motion before operation. To monitor respiratory phase, they use a magnetic resonance echo tracking of the liver and diaphragm. During operation, a respiratory state is again monitored, and a 3D image correspondent to the current state is extracted from the recorded four-dimensional (4D) data, therefore providing surgeons with a pseudo-real 3D image.

1.2.1.2 IGS based on intraoperative laparoscopic devices

Another group of methods attempt to construct an internal presentation based only on a laparoscopic camera. Sometimes they also use structured light emitters with patterns that are easy to detect and reconstruct.

The main limitation of such approaches is that laparoscopic cameras can provide us only with a superficial view. Since tumors and other areas of interest are mostly located inside the liver, these approaches cannot give us valuable information about them. Also, laparoscopic imaging modalities have limited observability in abdominal area; thus, we cannot reconstruct hardly accessible parts (such as back parts of organs, for example). However, assuming the availability of preoperative data, the intraoperative surface reconstruction will simplify its registration and following navigation process.

SLAM-based technique. In [25], the authors propose to create internal view based only on stereoscopic camera. The idea is to build incrementally a geometric representation of an

abdominal area based on the simultaneous localization and mapping (SLAM) approach. The SLAM method helps to identify the position of the camera, while simultaneously performing cross-correlation stereo matching.

The main limitation of this idea is an impossibility to detect features on the liver surface. A lot of particular points could be easily detected on objects with textured surfaces. But liver is rather homogeneous, and, consequently, not a lot of features are detectable on its surface, which dramatically influence the whole reconstruction process.

Structured light. The authors from [122, 123] propose to use a multispectral laser light emitter and reconstruct the organ surface based on its reflections (Fig. 1.4). First of all, the reflected light is segmented to find spots with colors. Then, the pattern is decoded together with estimation of the relative position between the laparoscopic camera and the laser probe. Given that information, the 3D positions of the found spots are determined. The object surface is then reconstructed by spline-based interpolation over spot centers positions. In [67], the authors insert a grid pattern projector inside a laparoscopic camera. They use a neural network system to extract features from a structured projection. Then, with light sectioning method, a depth image is obtained. Finally, the researchers construct an active bundle adjustment method to stitch all 3D shapes in a single surface.

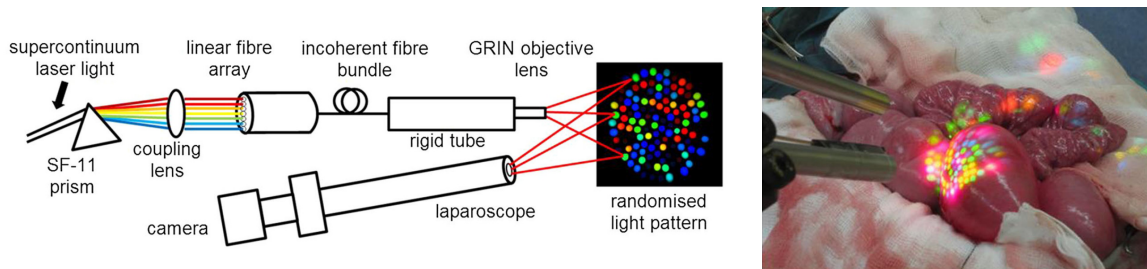


Figure 1.4: Left: structured light emitter for object surface reconstruction. The description of emitter components used to generate a structured light pattern. Right: Example of a miniaturized structured light probe in vivo. Images are taken from [122].

1.2.1.3 IGS based on ultrasound

Because of its simplicity and low price as well as the possibility to observe the internal structures, the ultrasound technology is often used to track the positions of tumors or other organ structures. In general, small part of the organ is visible through the US probe. Therefore, to plan surgery a scan of the whole abdominal (usually preoperative) area has to be acquired. However, the proposed ideas use only US probe scanning the region of interest during the procedure. Therefore we decided to classify them in a separate section of methods based only on intraoperative data.

The main drawback of the proposed ideas is that ultrasound probe allows tracking only very limited region. Because of attachments and various constraints, different parts of organ change its shape in a different way. Consequently the deformation behavior can be predicted more or

less accurately only in a small neighborhood around the tracked area. It is also worth marking that not all lesions are observable by the US.

Deformation compensation. In the papers [151, 213], the scientists propose ideas to compensate for respiration motion using a 3D US system. In [151], the porta hepatic of the pig is tracked with an US probe during several respiratory circles. At the same time, the probe itself is tracked with a magnetic tracker. From the tracking data, using multilevel B-splines, the displacement and deformation fields are recovered, which in its turn allows constructing a 4D respiration motion.

The researchers from paper [213] propose to compensate for a tumor movement with ultrasound. Initially, using an US probe, they detect and segment a tumor. During surgical operation, the position is updated using newly acquired ultrasound images. The position tracking allows target location correcting during navigation.

The researchers from [237] use 4D ultrasound for navigation. To compensate for various motions, they propose to use a non-rigid registration technique from [142] for the entire 4D sequence. The motion between two US frames is interpolated based on free-form B-spline deformation model. This way, it becomes possible to get object description at any moment.

1.2.2 IGS based on preoperative data

To improve the liver image-guided surgery systems, other proposed solutions take into account the preoperative data done for treatment planning. The main advantage of such approaches is that the preoperative scan is acquired to analyze the anatomical situation of the patient and therefore has sufficient quality. The additional information, such as resection planning, could also be fused with the intraoperative view.

But, using preoperative scans requires its matching with intraoperative data. The fact that it was done at another time, with a different acquisition device, for a particular patient position, and in another environment makes the registration process nontrivial. Anyway, scientists propose various ways to perform it. There are four main groups, in which the preoperative data registration methods can be classified. In case of manual registration, the user manually performs transformation of preoperative data. The markers-based registration group is focused on minimizing the difference between selected points in preoperative and intraoperative data. They might have a complex shape and be set either manually or in some specific way. The shape-based registration approaches attempt to find surface shape similarities for both sets of data. Finally, anatomical-landmarks based registration is focused on searching for anatomical features and fusing them.

Based on the presented ideas, the image-guided surgery techniques can be classified according to the liver motion model, used during surgical navigation. The following sections provide a more detailed description of the available solutions.

1.2.2.1 IGS with a rigid model

Some approaches rely on rigid liver model, which does not follow the organ deformations. They are based on the visible part of the intraoperative surface and focus mainly on its segmentation

and reconstruction. During the final step, the preoperative data is registered on the reconstructed shape. Numerous papers are also focused on the development of the new equipment used to track the liver motions. The augmented reality, constructed in such a way, is not accurate and, therefore, gives only approximate information about internal structures. On the other hand, the main advantage of these methods is their simplicity and stability.

The main ideas of this category proposed recently are given further.

Manual markers selection for registration. In [11, 177], the authors proposed a new navigation system in the operating room. The laparoscopic camera or any other device is tracked using several markers attached to a tool and a passive optical system. To register the preoperative data, several points have to be selected. After selecting the same points in the intraoperative data, the system reconstructs their three-dimensional positions and perform preoperative data registration. Recently, the created system was tested during surgical operations [12, 96, 189]. It has been proven that such systems will bring a benefit to a surgical navigation during surgery.

SLAM-based technique. In [133], the authors rely on a SLAM approach, similar to the case given in [25]. They introduce modifications to handle several issues, such as tracking failure due to occlusions. The registration of the preoperative data is based on several anchor points manually selected on a skin mesh. Then, the same points have to be selected on a camera video stream. By combining the two sets of points, the preoperative data can be matched to the live video stream.

The main drawback of these approaches is manual selection of marker positions. It heavily depends on a human expertise and might negatively affect the surgical workflow depending on the level of interaction.

Machine learning based registration. The authors from [131] propose to perform surface creation based on a machine learning algorithm. They use neural networks to reconstruct the surface from intraoperative images and to preprocess the preoperative scan. Then the reconstructed surface is registered to the preoperative surface with globally optimal iterative closest point method [253].

Depth reconstruction with laser scanner. In [68, 198], a system combined of a laparoscopic camera and a laser range scanner is proposed for three-dimensional intraoperative surface reconstruction. The scanner is used to generate visible points, which are detected by the camera. Several retroreflective markers are attached to both the devices, and the optical tracking system is used for tracking them and compute a relative position. By computing transformations, it is possible to register the coordinate systems of the devices in a global space and, thus, to get the 3D coordinates of laser points. After generating a sufficient number of points, a preoperative data can be registered.

Intraoperative depth sensors. Researchers in [141, 175] introduce a time of flight system to get a depth information. The time of flight sensor allows estimating the distance to the observed scene using the disparity between the light emitted by a diode and its reflection. The obtained depth map is used to reconstruct the observed 3D surface, to which the preoperative data can be registered. In [141], the authors perform CT registration by minimizing differences between surfaces. In the following works [102, 103], the time of flight sensor was replaced by a range sensor since it has a better spatial resolution.

1.2.2.2 IGS based on ultrasound

A number of works propose to use ultrasound probes (or its modifications) for intraoperative navigation. In general, the US gives a possibility to see the internal structures of the organ or tissue in real time. Besides that, it is the only modality that is cheap, easily accessible and, consequently, widely distributed.

However, a huge drawback is that ultrasound probe allows observing only very limited region and, thus, the localization of structures and tumors in liver parenchyma is difficult. Also, it is an active technique, which means that it is possible to observe data only in performing scanning. So, when the scanning procedure is stopped, the data becomes outdated. As well, the main issues are ultrasound registration, because limited range images are hard to register, and reliable tracking inside abdomen, since US probes are in general flexible and can easily change their shape.

Despite all these drawbacks, people propose solutions to guide surgery with ultrasound probes. The recent ideas are presented further.

Probe tracking. In [13, 97], the authors track the rigid US probe with infrared markers and optical tracking system. In [13], the tracking system is used to align images in global coordinate system and generate a volume image. Also, the authors use their own marker-based approach to perform a CT to US registration [14].

The authors from [115] integrated the tracking ultrasound probe in the da Vinci robotic system. For a stereo endoscope, also tracked optically, they found a relative transformation with US device. After registration, the robot kinematics can follow the endoscope movement, to place the US probe the way that its view corresponds to endoscopic one.

In the approach from [106], a flexible US probe was used. Therefore, the authors equip it with an electro-magnetic sensor, and use a magnetic system to track it. The registration of preoperative CT was based on markers with the usage of external fiducials and internal lesions.

The idea in [176] is to use for preoperative data a registration based on a single landmark with a very particular shape. The method uses the orientation of an optically tracked tool and reference points, which are marked manually, to align roughly the US image data to the reference frame of preoperative CT. Then, based on positions of target lesions visible through ultrasound, a registration update is performed.

Combined optical and electromagnetic tracking. In [104, 105, 152], the main idea is to perform a combined magneto-optic tracking for an endoscope and US probe to integrate ultrasound into an augmented reality system (Fig. 1.5). To track the position of the probe, the authors attach a magnetic sensor to it. They also add a magnetic field generator in their system. By tracking several markers fixed on an endoscope and the magnetic generator, the researchers compute the relative orientation for the devices. By combining the optical coordinate system with the magnetic one, they get the aligned view for the US. To register preoperative data, in [104, 105], the authors rely on the method described in [14].

Respiratory compensation. The authors from [18] propose a compensating alignment of a preoperative plan with the intraoperative US images. As in most works, the ultrasound probe is tracked using an optical tracking system. Before navigation procedure, a sequence of MRI images is acquired. Then the rigid registration between US and MRI is performed based on sim-

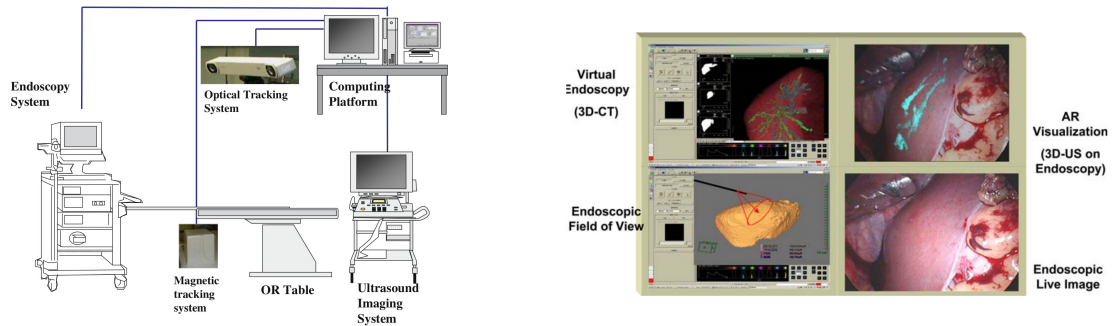


Figure 1.5: Left: architecture of the combined navigation system with an endoscope, ultrasound probe and magneto-optic trackers. Right: monitoring screen of the navigation system. It includes preoperative CT, US, and endoscopic view. Images are taken from [105].

ilarities of the blood vessel system. Using interpolation with B-splines, the researchers construct a motion model and use it to find the best interpolated shape for breathing phase correspondent to US. Finally, to get an augmented reality, they overlay the interpolated MRI data with the US.

1.2.2.3 IGS with respiratory adjusting

The general idea for more advanced image-guided surgery systems is not only to track the surgical equipment, but also to construct a model to adjust the liver deformation that inevitably happens during surgery. It covers mainly respiratory, which is a vital function, and thus can be stopped only for a small period of time (1-5 minutes). So, most of these approaches try to improve the preoperative data registration by compensating the respiratory movements.

However, these approaches mainly consider respiratory as periodic cyclic motion, while in general, the breathing is not regular. Moreover, the proposed approaches update the AR at the correspondent respiratory state, but they do not simulate the organ movement in real time. Apart from cyclic movements like respiratory motion or heart beating, there are also casual movements that might appear because of, for example, surgical tool manipulation. These movements have unpredictable nature; therefore, it is hard (if even possible) to adjust them. Even supposing this, it is difficult to predict the organ behavior on account of the influence from the additional constraints caused by surrounding organs and tissues.

The approaches using ultrasound systems were already discussed in previous sections (see subsection 1.2.2.2). The other approaches are listed further.

Surface markers tracking. In paper [124], the authors propose to put a custom shape markers on patient skin and use a tracking system to track their positions. The same markers are used to acquire a preoperative CT, and therefore the preoperative data can be automatically registered by minimizing the distance between them. The augmented reality is updated only when the shape of a marker cloud corresponds to preoperative shape set.

CT based respiratory adjustment. In [159], the researchers construct the augmented reality with a CT and a visual system composed of two jointly calibrated cameras. They place several radio-opaque markers on skin to register them on CT scan and camera images. To ad-

just the breathing motion, they use respiratory gating technique with a monitoring system. The augmented reality is updated only when respiration phase fits the phase when CT was taken. Finally, they also add a possibility to track a needle, to know its position inside the abdominal area.

SLAM-based technique. In the work [150], the authors propose to construct the AR, based on motion compensated simultaneous localization and mapping technique (Fig. 1.6). The abdominal area around the liver is updated in iterative way, while the stereoscopic camera is tracking inside. To handle respiratory motion the researchers improve SLAM by including asymmetric respiration model [130]. In [149], the authors improve their work further by using CBCT scan as a connective element to register a preoperative CT and a laparoscopic camera in the global coordinate system. The preoperative data is registered on the CBCT using biomechanically driven approach [165]. The registration of the stereoscopic camera on CBCT is done in several steps. First of all, the respiration phase is determined. After that, the correspondent 3D features are detected on the stereoscopic point cloud and cone beam computed tomography. Lastly, a surface to surface registration is performed based on difference minimization approach. The authors claim that the proposed registration allows matching the preoperative data and the stereoscopic images with sufficient accuracy.

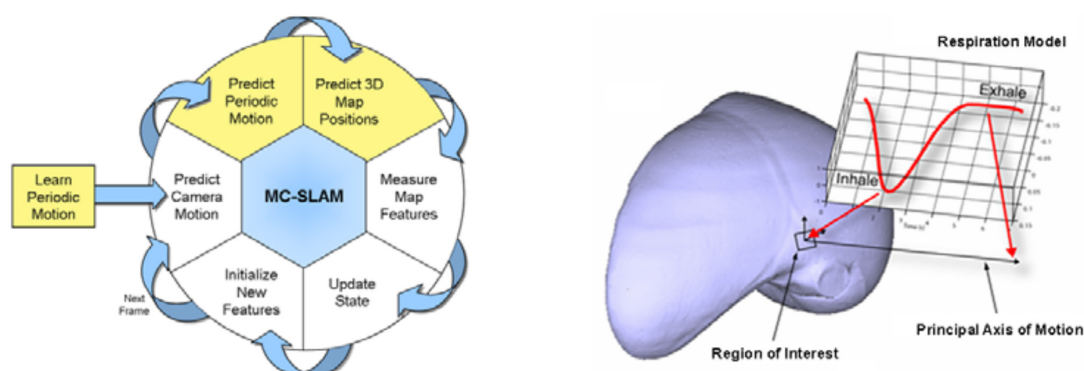


Figure 1.6: Left: main steps of a motion compensated SLAM approach. The approach sequentially in a loop improves the accuracy of the model and specifies the positions of the camera. Right: respiratory simulation for the liver organ. The motion of the region is tracked and the analytical respiration model is fitted to it. Images taken from [150].

The existing drawbacks impose a huge limitation to the application of the available propositions. As a result, we think a simulation model of liver behavior will be a better solution for IGS systems.

1.2.2.4 IGS based on a simulation of liver deformation

The general idea of the simulation methods is to model the behavior of the whole organ based on tracking data that is obtained during surgery procedures. The simulation approaches might vary from simple interpolation between tracking points up to complex biomechanical modeling. They can be split into two groups.

Parametric interpolation. The authors from [114, 134, 136] propose to construct liver deformation by interpolating data from a bundle of electromagnetically tracked flexible needles. They acquire an initial CT scan after needles insertion to use their positions for CT registration on the intraoperative tracking system. The transformation for the registered image is then computed through an interpolation based on tracking positions.

The idea in [135] is to track the motion using a set of optically tracked needles. The initial computer tomography is registered via spline interpolation. During intervention, the three-dimensional model is updated based on similar interpolation and the positions of needles.

The main drawback of the proposed approaches is that they do not consider the anatomy of a specific patient. The interpolated parameters are computed based on some general equations, but depending on liver properties the actual deformation might differ. The accuracy of the simulation is highly dependent on the number of tracked markers, but it is not possible to insert a lot of needles in patient organs due to ethical restrictions and possible additional complications.

Biomechanical modeling. A large set of ideas are based on physically based modeling, which describe, more or less accurately, the organ deformation, based on measured hyperelastic properties. The papers [74, 188] describe the patient-specific biomechanical model and its correspondence to real data from a laparoscopic camera. The registration of preoperative scan is done based on anatomical landmarks visible intraoperatively. The tracking algorithm, which works during an operation process, basically relies on feature detection and Lucas-Kanade optical flow approach.

In [23], the authors describe the linear Finite Element Method (FEM) to simulate the organ deformation during image-guided surgery. During operation, the information about the liver surface is obtained using a laser range scanner. For preoperative CT registration, the scientists minimize the difference between CT and the obtained surface. To control the deformations of biomechanical model, the researchers use an optical tracking system, which tracks probes embedded with infrared diodes attached to the scanner.

Synthetic data-driven biomechanical modeling. In [20], the authors proposed an approach combining a Finite Element Method and a deep neural network to learn complex elastic deformations. From preoperative computer tomography data they construct a biomechanical model, which allows them to simulate various deformations for network training. Based on tracking data obtained from an RGB-D camera, the network predicts the deformation for the whole volume model.

The main disadvantage of the proposed solutions is their complexity. The Finite Element Method is time consuming and, in general, not applicable in real-time context. However, the recent researches [35, 179] show that the finite element based modeling process can be enhanced to provide a real-time simulation.

1.3 Guided surgery for other highly deformable organs

It is important to mark that simulating organ behavior using physically based approach is quite general and could be easily integrated in navigation systems for other deformable organs.

For example, during respiration, the tumors inside lungs can move from 1 to 3 centime-

ters [197]. In addition to that, there is also a change in the lung volume. The difference in volume for free breathing and gated inhale volumes ranged from 25 % up to 46 %.

The papers about computer-assisted lung surgery are not numerous. In [60], the authors publish a study of using an augmented reality system to treat a lung cancer. In [205], the authors use virtual reality as a navigation tool combined with CBCT to guide intraoperative localization of tumors. The researchers from [225] create the three-dimensional virtual model by separating segmentation of trachea and blood vessels. They also merge them with pulmonary parenchyma to get a single model. But all created systems show only the static model of the lung, while a possibility to model dynamic behavior makes navigation more accurate. Recently a Finite Element approach was proposed to compensate for lung deformations during initial data registration [7]. Therefore, a real-time biomechanical model would be a nice solution since it could simulate breathing as well as other motions.

Kidney is another organ that changes its position during respiration. The research show the movement up to 3.5 cm for a normal respiration and up to 8.6 cm for a deep one [144]. However, the scientists from [210] claim that their deformation is limited. Currently, most of the available approaches use rigid registration of preoperative data. But usage of a deformable model gives a better precision for tumor positions. For example, the authors from [56] show that biomechanical model for kidney improves an average localization error on 29 %.

The recent overview of all available approaches is given in [41]. In general, it is possible to say that navigation is done using various approaches, from using additional navigation aids [15] up to more complex intensity-based approach [192]. The latter attempts to fuse the preoperative data with the intraoperative images. Here, to improve the approach, the authors introduce the recovery algorithm because of occlusions that often happen during surgery. Anyway, there are several attempts to model kidneys using biomechanical models. In [6], the authors use an elastic biomechanical model to simulate the behavior of a kidney. The authors from [166] use deformation model based on Biot's consolidation theory. In [56], the authors use linear Finite Element Method to simulate the deformation. Finally, in [162], the authors propose to use hyperelastic FEM, which simulates kidney behavior more accurately. All these works mark an increasing interest in physically based modeling; it is considered as a feasible solution to predict the position of the whole organ and, particularly, invisible tumors with a given limited observability.

1.4 The selected approach for a liver augmented reality

All in all, we decided to construct the augmented reality for the liver using a physically based modeling. During hepatic surgery, only a small part of a liver surface is observed, for both open and laparoscopic surgery, from which a displacement field for the whole organ volume has to be recovered. Also, compared with other organs, such as bones, liver undergo various deformations, due to tools manipulations, respiration motion, and abdominal area inflation (in case of laparoscopic surgery). Biomechanical modeling has proven to be useful in simulating the large deformations, as well as in cases with limited amount of intraoperative data. Consequently, in this case, a physically based solution looks very efficient at estimating the in-depth displacement from surface motion or sparse data. Finally, it is important to mark that biomechanical

modeling can be easily combined with other equipment such as ultrasound probes, which are used during surgery, or other tracking tools.

1.5 Research work overview

There are several main components that every mechanical problem contains: (1) a numerical approach, like a Finite Element Method, that interconnect the geometry of deformation with material strain; (2) an optimization method, such as Newton-Raphson scheme that deals with equations of equilibrium between stress and applied tensions or external forces; (3) a constitutive law that describes material properties by interconnecting stress and strain; and (4) a set of boundary conditions (BCs) that characterize how the region of interest is connected to the surrounding environment.

In the domain of modeling for medicine, the described components are highly dependent on anatomical characteristics of a specific patient. Indeed, the shape of the organ, tissue properties, the connections to the surrounding environment, and biological processes influence the mechanical behavior and, therefore, play an essential role in the simulation process. While some information, such as liver shape, can be easily obtained using medical equipment, other is much harder to get. The main issue comes from the impossibility to identify certain anatomical structures and their biomechanical properties.

Some time ago, a new *Dynamic Data Driven Applications System (DDDAS)* paradigm was introduced [38]. According to this paradigm, the behavior of a dynamic system is predicted by a model, but also the model is updated by indirect measurements obtained from the system. It seems that to construct the accurate AR system for surgical needs we have to follow this paradigm. The characteristics could be estimated by assimilating data from various medical modalities such as laparoscopic cameras, ultrasound probes, and so on that are used during surgery. It is worth marking that anatomical properties of organs might vary a lot between different patients. But the main advantage here is that the assimilated data takes into account the characteristics of a specific organ or tissue.

The liver organ is studied quite extensively. A lot of works are dedicated to it; the good overview is presented in [138]. However, a huge number of papers related to the construction of a numerical model for real-time liver simulations, necessary to create the AR system, and to the description of hyperelastic liver material properties. And much less work has addressed to the role of BCs in the organ deformation.

Liver boundary conditions are described very poorly. The description of liver connective tissues is limited only to a generic idea about their position. The main issue comes from the impossibility to identify anatomical structures constraining the liver, mainly ligaments, from the preoperative CT and the difficulties with their visibility on magnetic resonance imaging [50], especially healthy tissues. Moreover, in generally, the information about their biomechanical properties is missing. As a result, it is nearly impossible to get a correct description of the liver boundary conditions.

The main idea, therefore, is to use the available recordings of liver deformation. Since there is no exact information about BCs, their properties can be estimated based on a data assimilation

process. Also, it is essentially important when surgeons operate a person who has a non-typical anatomy. For example, the patient might not have a specific ligament because it was cut during previous surgery. We have to note that, in general, it is not possible to identify a ground truth BCs, only to find some effective values, which will allow improving the simulation accuracy. However, the corrected BCs will clarify the biomechanical model, which results in an accurate simulation of the organ even when observations are very noisy or missing.

All in all, there is a need to develop a better navigation platform that will overcome the drawbacks of the traditional IGS system. This is the main goal of "HiPerNav" project, which aims to provide a navigation mechanism to deal with issues during cancer and metastases treatment. The project is based on collaboration among several universities, research institutes, hospitals, and industrial partners from all over Europe. Every partner is responsible for work on a specific aspect of the constructed platform (Fig. 1.7).

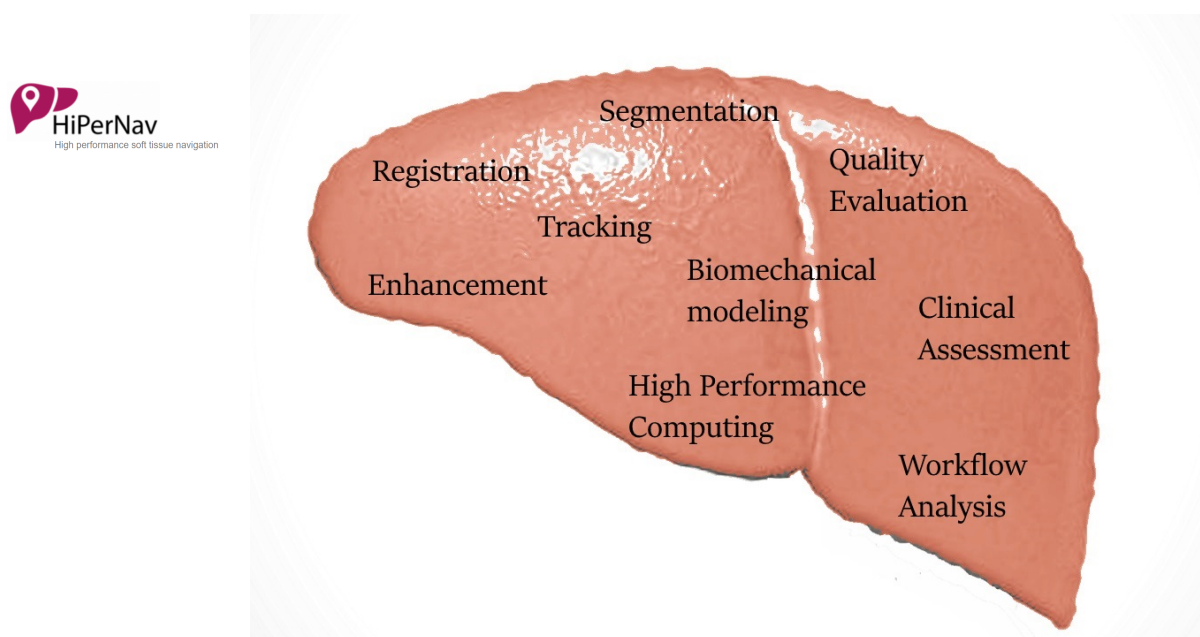


Figure 1.7: Overview of research areas that are covered by the HiPerNav project (<https://hipernav.eu>).

Together with other requirements, the platform should be able to monitor and manage the surgical workflow in intraoperative resection and ablation. The current PhD work is also a part of HiPerNav project. The main idea of the work is to propose a solution to estimate the liver boundary conditions, in order to predict its behavior more accurately. It will result in a well suited augmented reality for image-guided liver surgery. The introduced idea relies on a combination of a liver model and a nonlinear ligament presentation, initialized from a statistical atlas and corrected by a stochastic data assimilation process.

The main contribution of this thesis is the following:

- A presentation of boundary conditions around liver as a mass-spring system with poly-

nomial stress-strain relation. The nonlinear mass-spring system allows simulating hyper-elastic tissues with sufficient accuracy, but nearly 30 times faster.

- A nonlinear Kalman filtering based approach to estimate liver BCs using available information from intraoperative data. The essential moment is that the method does not precisely simulate characteristics of liver surrounding organs and tissues, but improves the prediction accuracy of a biomechanical model to simulate liver behavior.
- A statistical atlas construction and parameters initialization procedure to create the initial approximation of BCs around the liver. Based on set of liver meshes, with ligament positions segmented by experts, and a stress-strain curve taken from literature, it constructs the initial set-up for position and elastic properties.
- A preconditioned transformation of uncertainty or ensemble members for stochastic data assimilation process. When used inside the prediction step, it allows decreasing the computation speed without sufficient loss of accuracy in estimated parameters.

The structure of this manuscript is as follows:

Chapter 2 gives an overview of biomechanics for liver and ligaments and depicts a numerical approach that simulates their behavior. Finally, it argues about the selected approach that was made to simulate the behavior of the organs and tissues.

Chapter 3 argues for the Kalman filtering approach as a best choice to perform data assimilation process under given accuracy and time constraints. It then gives a detailed description of the Kalman filtering approach and two of its nonlinear extensions together with some experimental results.

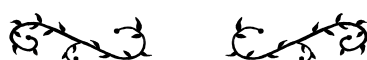
Chapter 4 describes an atlas construction and parameters initialization for initial BCs approximation. It then focuses on the preconditioning approach to improve the performance for data assimilation process.

Finally, Chapter 5 depicts the experiments and estimation results of liver boundary conditions for synthetic model and then for a real human organ. The chapter is followed with their discussion, possible limitations, and future steps.

The manuscript is finalized with the conclusion and bibliography list.



BIOMECHANICS OF LIVER AND LIGAMENTS



Contents

| | | |
|------------|--|-----------|
| 2.1 | The fundamentals of liver modeling | 20 |
| 2.1.1 | Liver anatomy | 20 |
| 2.1.2 | Liver biomechanics | 23 |
| 2.1.3 | Simulation of liver deformations | 24 |
| 2.1.4 | The numerical simulation of liver | 27 |
| 2.1.5 | The selected approach to simulate liver deformations | 30 |
| 2.2 | Liver boundary conditions | 32 |
| 2.2.1 | Importance of boundary conditions | 32 |
| 2.2.2 | Theoretical description of boundary conditions | 33 |
| 2.2.3 | Anatomy of liver boundary conditions | 34 |
| 2.2.4 | Boundary conditions simplification | 37 |
| 2.3 | The fundamentals of ligaments modeling | 37 |
| 2.3.1 | Ligaments anatomy | 37 |
| 2.3.2 | Biomechanics of ligaments | 38 |
| 2.3.3 | Ligaments simulation | 39 |
| 2.3.4 | The model for ligaments | 42 |

In this chapter, we will present how the knowledge about liver and ligaments anatomy and properties is used to construct a biomechanical model. Firstly, we will give a description of the liver organ and depict an overview of available hyperelastic materials to simulate its behavior. We will then argue for our model choice. We will also show that boundary conditions have a huge influence on the shape of the object during a simulation process. For the liver, the BCs are mainly presented by ligaments. So, we will describe a state of the art addressing various ideas to simulate ligaments in human body. After, we will present an experiment to validate our choice for a ligament model.

2.1 The fundamentals of liver modeling

2.1.1 Liver anatomy

The liver is the largest human gland, which produces substance called bile, and plays an important role in digestion. It also has a unique capability of regeneration. The bile substance is stored in the gallbladder and discharged into the duodenum where it aids with the digestion of lipids. Apart from that, the liver provides a number of vital functions to maintain the metabolic balance. It synthesizes various essential molecules, extracts and metabolizes nutriment, filters out different toxins from the blood, stores and processes metabolic products, neutralizes and detoxifies alien antigens and bacteria [9].

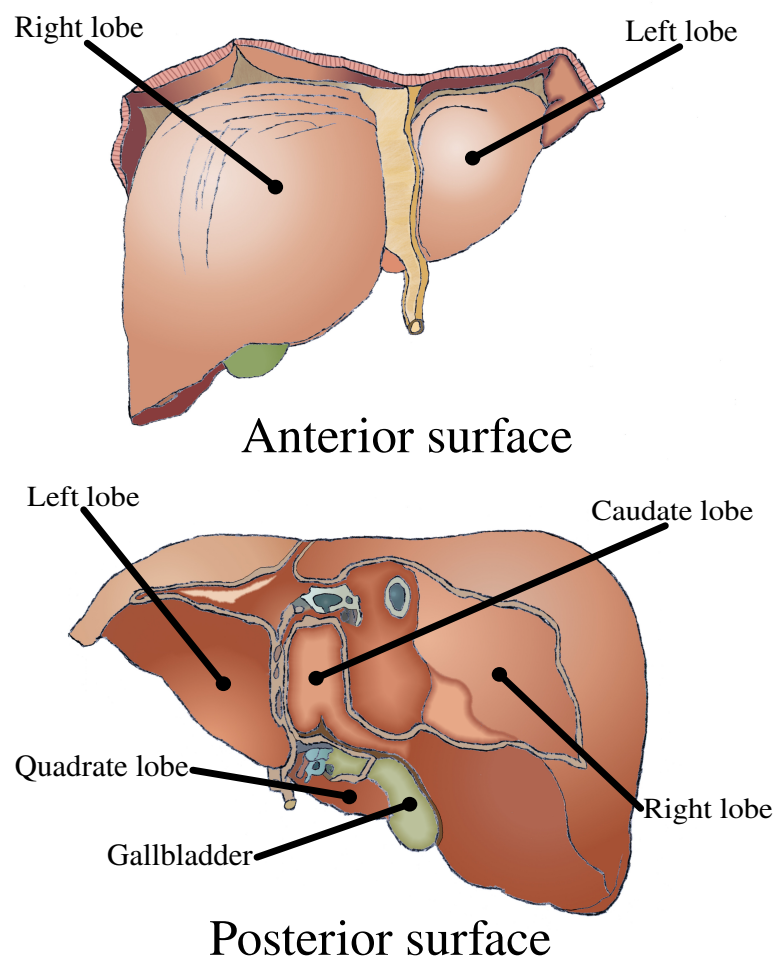
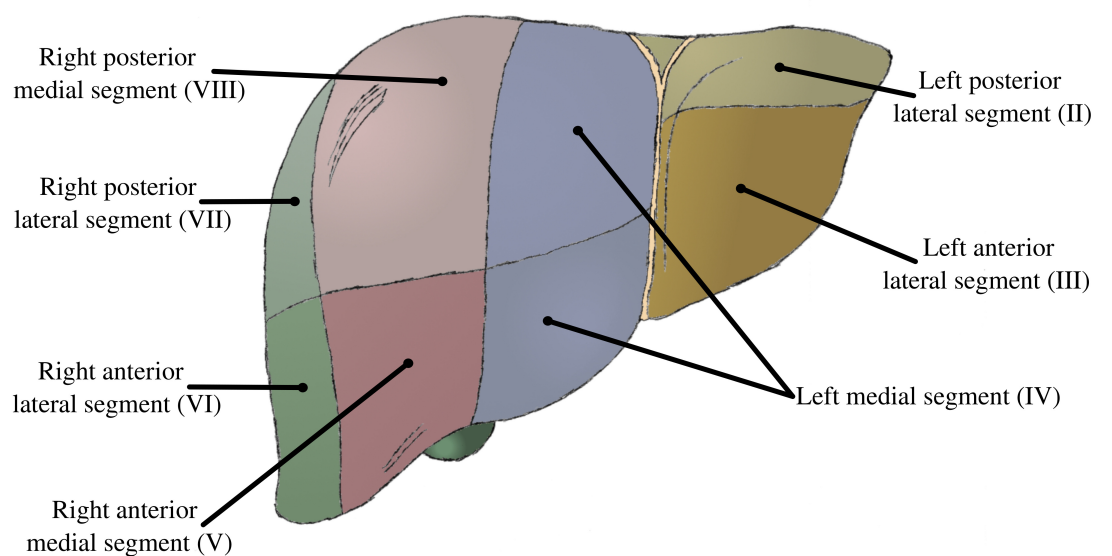


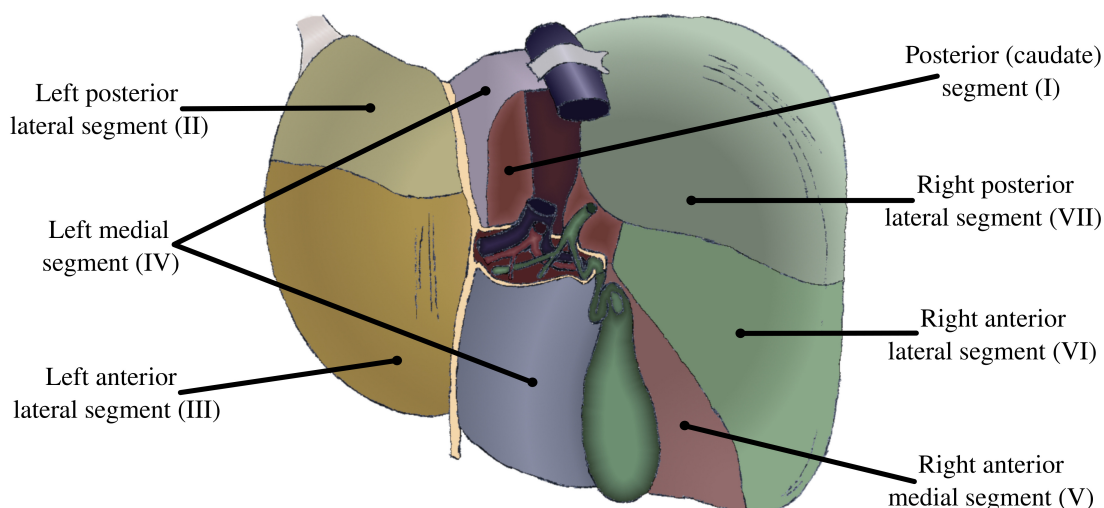
Figure 2.1: Liver anatomy overview with labeled lobes.

The liver is situated in the upper part of the abdomen on the right side. Its color is dark red-brown. The shape of the liver can be roughly described by a triangular prism with the base on the right and top on the left (Fig. 2.1). The weight of the liver varies from 1.2 to 1.6 kilograms.

Generally, its greatest transverse measurement can vary from 20 cm to 22.5 cm. Vertically, near its side or right surface, it measures about 15 cm to 17.5 cm. Also, its greatest anteroposterior size is from 10 cm to 12.5 cm. Opposite to the vertebral column, the size is reduced to 7.5 cm [72]. The liver volume varies from 1200 cm^3 up to almost 2000 cm^3 [8]. The liver contains three surfaces: inferior, superior, and posterior. It can also be split into four lobes: left, right, caudate, and quadrate [72].



Anterior view



Posterior view

Figure 2.2: Overview of the liver segments that are commonly used for resection planning.

But usually, according to *Couinaud classification*, the organ is considered as a composition

of functional segments. The overview of the segments is presented in Fig. 2.2. This classification is more preferred for surgeons since it divides the liver into eight independent parts. Each of these parts has its own vascular inflow and outflow, and, therefore, they could be resected or operated without damaging the remaining part of the organ. The exception is the caudate lobe, which gets the blood supply from the two vascular branches.

The liver is sustained by hepatic artery and two veins. Inside liver tissue, these vessels are divided into hundreds of small capillaries, which are called liver *sinusoids*. The blood from a portal vein and portal artery, which together with a bile duct form a *portal triad*, is filtered and mixed inside the sinusoids. It goes then to a central vein, which transfers it further to a hepatic vein. The sinusoids are covered by hepatic cells of the liver *lobules*. The lobules are small granular bodies with a diameter from 1 mm to 2.5 mm (Fig. 2.3). They are separated by

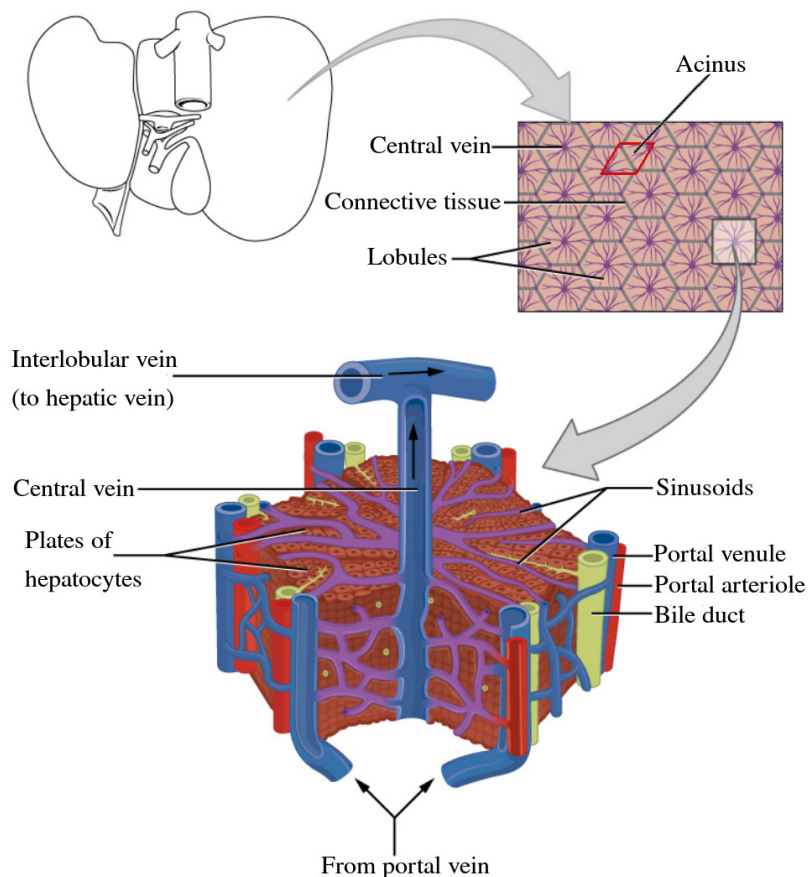


Figure 2.3: The functional units of liver shown in the hierarchical order. Image taken from [17] licensed under CC BY 4.0.

a very thin *areolar* tissue [72]. Most of the metabolic functions are going inside the lobules, so they could be called the main components of liver tissue. However, commonly biologists consider an *acinus* as the elementary functional unit of the liver. The acinus is a rhombic region artificially constructed on parts of two adjacent lobules and bounded by two portal triads and two central veins (Fig. 2.3).

The whole mass of lobules (or acini) together forms the essence of the liver tissue substance, which is called *parenchyma*. The functional units have more or less regular structure; consequently, the parenchyma has quite homogeneous mechanical properties. It is nearly incompressible tissue, and the average Young's modulus ranges from 1 kPa up to 20 kPa. The blood vessels, however, have different properties, and, therefore, they influence the homogeneity of the tissue.

Liver cancer diseases are characterizing by appearing abnormal cells, which do not provide their normal function. A solid tumor is an abnormal tissue made of cancer cells and a variety of normal organ cells (Fig. 2.4). In general, the liver cancer caused more often by metastasis - the cancer cells that are spread from other organs - because of the huge blood flow that is going through the liver. The normal tissue is participated in complex metabolic processes. The metabolism is regulated by biophysical equilibrium between lobules and extracellular matrix [231, 247]. The cancer cells violate this balance, which results in changing the physical and mechanical properties of the cells. The most cancer diseases are characterized by desmoplasia. When it occurs, the fibrous and connective tissues are extensively growing, which increase the stiffness of the tumor. The stiff tissue affects the blood vessels with enormous pressure, making them more leaky and facilitating tumor progression as well as its metastasis. In addition, the tumor stress affects the surrounding normal tissue, helping the tumor to grow and spread further. The recent research shows the increasing interest in the role of mechanical forces and stress in the tumor and its influence on the surrounding tissues. It is described in more details in [86, 158]. As a result, the tumor can be considered as a substance much stiffer than the cells of a normal tissue.

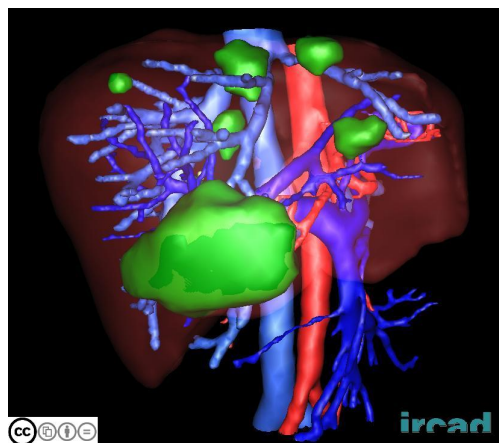


Figure 2.4: Liver parenchyma and tumors reconstructed from CT images. Since tumors have abnormal mechanical properties, they have different intensity on CT and MRI images, which is used to segment and reconstruct their shape. Image taken from [219].

2.1.2 Liver biomechanics

The material properties of an organ or tissue are described by the characteristics of cells as well as the structure of an extracellular matrix. Protein fibers, hepatocytes, and other components

make a contribution to a final mechanical characteristics, so any change in this composition leads to a variation in mechanical properties.

Historically, the stiffness of any material was described in terms of linear elastic theory. It is given by two properties: Young's modulus and Poisson's ratio. However, such as many other soft tissues and organs, the liver has nonlinear behavior. Various experiments were performed to evaluate its characteristics. For example, in [22], to estimate stress-strain relationship, the authors rely on indentation test; in [155], the authors use an aspiration device. The results are presented in Fig. 2.5. It is nice to mark that authors presented stress-strain curves for experiments with both healthy and diseased organs. Despite the fact the diseased liver respond differently to stretching forces, it has the similar curve shape. Consequently it allows us to use the common hyperelastic material for both cases. The ideas that have been proposed to simulate its deformation are described further.

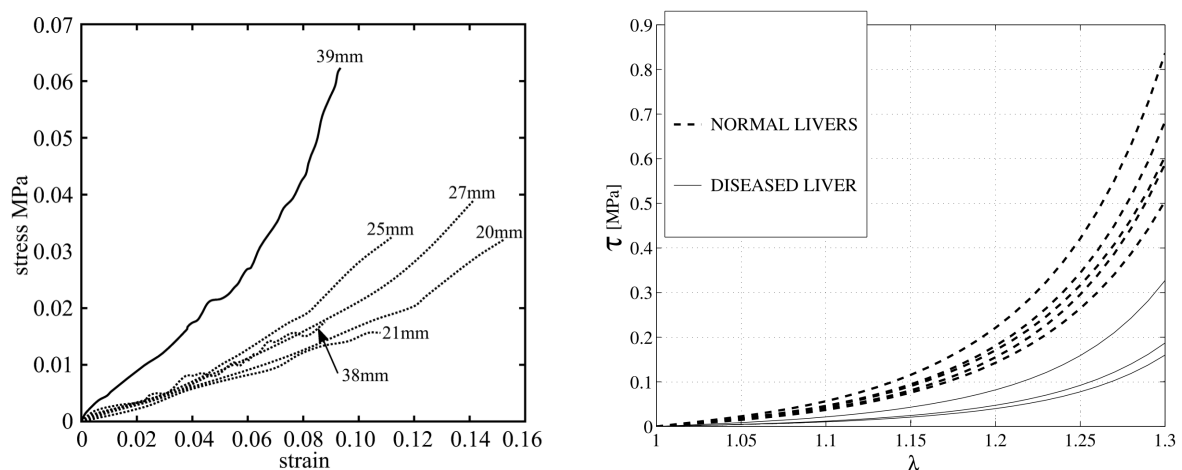


Figure 2.5: Overview of the liver constitutive law. Left: four normal livers (dotted lines) and one obstructive liver disease (solid line), taken from [22]. Right: the chart shows the nominal stress τ as a function of the stretch ratio λ , taken from [155].

2.1.3 Simulation of liver deformations

Because of its regenerating capabilities, the liver organ is an often target for surgical resection procedures. Therefore many research projects focus on liver modeling for various simulators and computer aided tools. Depending on the area of application and performance requirements, various methods are proposed.

2.1.3.1 Elastic materials

Despite the fact that liver has nonlinear behavior, in some works a linear elastic material is proposed. The main motivation in this case is a requirement for real-time computations. In [23], the linear Finite Element Method (FEM) was used to simulate the organ deformation during

image-guided surgery. In [33], the authors also use a linear elastic model to create a real-time liver simulator for surgical training. The result of such simulations is fast, but it is correct only for deformations with small strains. So the main tendency is to use various hyperelastic models.

2.1.3.2 Hyperelastic materials

The simplest approach proposed to model nonlinear deformations is co-rotational FEM [156]. The main idea is to model the deformation as a combination of a rigid object rotation and linear deformation. More precisely, the stiffness in this case can be written as:

$$\mathbf{K}_{nonlinear} = \mathbf{R}^T \mathbf{K} \mathbf{R} \quad (2.1)$$

where matrix \mathbf{R} represents the rigid rotation of the object and \mathbf{K} is the linear stiffness matrix (see section 2.1.4.1 for more details).

Despite the fact that this approach is not truly hyperelastic and various issues, such as stability problems, it can be used easily in real-time context. Therefore, it was used in a number of works related to a fast liver behavior simulation [36, 179].

Among hyperelastic materials, the simplest one is *Saint Venant-Kirchhoff (StVK)*. It is a generalization of the isotropic linear model where the linear strain tensor is replaced by the nonlinear Green-Lagrange strain tensor. The relationship between stress and strain is expressed through the *strain energy density function* Ψ (see section 2.1.4.1) as:

$$\Psi(\boldsymbol{\xi}) = \frac{\mu_1}{2} (tr(\boldsymbol{\xi}))^2 + \mu_2 (tr(\boldsymbol{\xi}^2)), \quad \boldsymbol{\xi} = \frac{1}{2}(\mathbf{c} - \mathbf{I}) \quad (2.2)$$

where $\boldsymbol{\xi}$ is a tensor that describes material strain, μ_1 and μ_2 are two Lamé coefficients, \mathbf{c} is the Cauchy-Green deformation tensor, and \mathbf{I} is the identity matrix. The use of StVK model was proposed, for example, in [40] where the authors make an attempt to create a real-time surgical simulator. Some approaches rely on StVK model with a further improvement for better performance. In [161], the authors apply proper generalized decomposition to accelerate the simulation process. The authors from [183], to achieve a real-time performance, compute the nonlinear part of the force only for the parts which exceed certain threshold. As a result, we can say that StVK material is quite popular for modeling deformations in real-time context.

Neo-Hookean material is another simple hyperelastic material that can be used to predict the nonlinear stress-strain behavior. The strain energy density function Ψ is expressed as:

$$\Psi(I_1) = \frac{\mu_1}{2} (I_1 - 3), \quad I_1 = tr(\mathbf{c}) \quad (2.3)$$

where μ_1 is the first Lamé coefficient, characterizing material stiffness, and I_1 is first invariant or the trace of the Green deformation tensor. Some works, such as [4], propose to use this material to simulate liver behavior.

The Mooney-Rivlin material is usually considered as a generalization of Neo-Hookean model. In generalized version the strain energy density is presented as:

$$\Psi(I_1, I_2) = \sum_{i+j>0}^N \theta_{ij} (I_1 - 3)^i (I_2 - 3)^j, \quad I_2 = \frac{1}{2} [(tr(\mathbf{c}))^2 + tr(\mathbf{c}^2)] \quad (2.4)$$

where θ_{ij} are material constants, I_2 is the second material invariant in the line. For N equals to 1, the generalized equation becomes a simple Mooney-Rivlin material.

The Mooney-Rivlin model is selected quite often for various simulations. For example, in [81], the authors choose Mooney-Rivlin material to find parameters from real tissue samples. The authors from [63, 64, 233] also propose to use a second order Mooney-Rivlin model. In [251], the authors show an extension of the nonlinear FEM based on Mooney-Rivlin model. They use this model to simulate the liver behavior for a surgical training simulator.

Among other models, the commonly used one to simulate the behavior of hyperelastic materials is Ogden model. In this model, the relationship between stress and strain is expressed through the strain energy density function Ψ as:

$$\Psi(\lambda_i) = \sum_{i=1}^N \frac{\nu_i}{\alpha_i} (\lambda_1^{\alpha_i} + \lambda_2^{\alpha_i} + \lambda_3^{\alpha_i} - 3) \quad (2.5)$$

Here ν_i and α_i are material parameters and λ_i are principal stretches. The stretches are connected with material invariants given earlier via simple equations:

$$I_1 = \lambda_1^2 + \lambda_2^2 + \lambda_3^2, \quad I_2 = \lambda_1^2 \lambda_2^2 + \lambda_2^2 \lambda_3^2 + \lambda_3^2 \lambda_1^2 \quad (2.6)$$

The researches who propose to use the Ogden model are given in [69, 81, 125]. In [234], the authors search for Ogden model parameters based on experiments with bovine livers. And in [127, 235, 126], the authors estimate Ogden parameters for human liver.

Alternative solutions consider more complex models. For example, in [125], the authors rely on a reduced polynomial model to simulate the liver. In [202, 204, 258], the authors describe exponential models. The researchers from [255] perform verification on how Veronda-Westmann material fits to the experimental data. Unfortunately, the available material parameters are obtained for animal livers, but not for human ones.

An interesting research was done in [27, 28] where the authors compare different hyperelastic models. Finally, they propose to use a combined logarithmic and polynomial material as the most accurate one. However, the result of their research shows that all models apart from the simplest ones simulate liver hyperelastic behavior with a sufficient accuracy.

2.1.3.3 Viscosity and porosity

Besides hyperelastic aspect, the certain number of papers covers viscosity description in FEM. There are two generally proposed approaches to model liver viscosity.

The first one is related to linear viscoelasticity. In this case, the material constants are presented as a time-dependent relaxation function. To determine this function, it is usually written in a form of *Prony series*, the coefficients of which is easy to obtain from viscosity experiments. For example, in [143], the authors obtain the final material by adding this presentation to a specific form of generalized Mooney-Rivlin model. In [208], the researchers add viscosity to the Ogden model in the same way. And in [154, 153], the authors use reduced polynomial model for the hyperelastic part.

The other one is quasi-linear viscoelastic theory. This theory assumes that the stress history can be expressed as multiplication of two independent terms, where the former describes

time relaxation and the latter indicates the instantaneous elastic response. In [224], the authors show by experiments that quasi-linear model can describe viscoelastic properties of the liver. To simulate stress-strain relationship, they use exponential model described in [65]. In [95], the authors also use quasi-linearity, but with Mooney-Rivlin material to simulate hyperelasticity.

Several research works develop other constitutive models to predict more accurate liver behavior for large strains. In [221], the authors create their own model. The model includes the initial elastic response, the rate-dependent part to simulate viscous flow behavior, and the nonlinear strain hardening that increases sufficiently at large strains. However, they develop their model for high strain rate and high-speed impact loading that happens, for example, during vehicle crashes. The authors from [160] propose to use the similar hardening law to simulate the nonlinear viscosity. They extend the linear relaxation modulus by introducing a stiffness multiplier depending on the strain rate. The parameters for this model are estimated based on experiments with porcine livers.

In other research works, the authors develop even their own approaches or use nonstandard ones for simulation. In [99], the authors simulate liver behavior to insert a needle accurately. They extend the standard Finite Element Method with the cubic function to model nonlinear constitutive law and quadratic function to model viscosity. In [212], the authors propose to create a nonlinear model by adding corrective functions to a linear mass-tensor system. The modification describes the correction for Lamé coefficients and viscous force proportional to speed of deformation. In [226], the researchers extend the Neo-Hookean model with anisotropy and append linear viscosity based on Prony series. To simulate liver deformations, the authors propose to use a Total Lagrangian Explicit Dynamic formulation.

In the paper [98], the authors do not consider hyperelasticity at all. To simulate liver viscosity, they use a Kelvin-Voigt model.

Apart from viscoelasticity, some researchers make an attempt to include porosity for modeling liver behavior. In [88], the authors propose to model viscosity by adding relaxation to the shear modulus and to model porosity by adding it to the bulk modulus. The experiments, done on a porcine liver, show that they can identify parameters to simulate material response accurately. In papers [138, 139, 196], the researchers construct a liver model for injury analysis and surgical training. The porosity is modeled using Darcy's law that describes a flow of a fluid through a porous substance. The solid part of the liver was modeled as viscoelastic material. Unfortunately, all tests were performed with pig livers and just a few obtained parameters, which makes it difficult to set the similar material characteristics for human liver. Also, the simulation for the result poroviscohyperelastic material is far from real time.

2.1.4 The numerical simulation of liver

Solving nonlinear partial differential equations for object deformation requires an appropriate numerical technique. The Finite Element Method (FEM) is often used in this case. The surface of a real object is considered as a domain, also called *mesh*. The mesh is discretized into simple geometrical shapes, called cells or elements. Depending on the nature of the problem and overall geometry, they can be different polytopes.

In this work, we model the liver using tetrahedral elements. The general idea of augmented

reality is to provide surgeons with information about internal tumors and structures. Therefore we need a model to predict the behavior of an invisible internal organ part, which can be done using a volumetric model. The volume elements are easy to generate from a 3D surface mesh of the organ. The surface can be reconstructed from the segmented images of the liver in almost automatic way.

2.1.4.1 Finite element modeling

In this section we describe a finite element method in more details. Let us suppose that some domain $\Omega = \Omega^N$ was deformed under some external forces. The deformation of the body can be described by some transformation function Φ that associates to every element \mathbf{s} its deformed position $\Phi(\mathbf{s})$. The displacement vector $\mathbf{u} = \Phi(\mathbf{s}) - \mathbf{s}$ shows the difference between positions without rigid motion. The deformation is usually characterized through *deformation gradient* \mathcal{F} that describes the distance difference changing between elements [250].

$$\mathcal{F} = \nabla_{\mathbf{s}}\Phi = \mathbf{1} + \nabla_{\mathbf{s}}\mathbf{u} \quad (2.7)$$

In deformation, the distances between elements are changing. The distance changing is characterized by a second order measurements [109]. For strain, a number of second-order tensors from the deformation gradient are obtained. The simplest is the *Cauchy-Green deformation (strain) tensor* \mathbf{c} , which describes the deformation of the domain and equals to a squared gradient:

$$\mathbf{c} = \mathcal{F}^T \mathcal{F} \quad (2.8)$$

The *Green-Lagrange strain tensor* $\boldsymbol{\xi}$ describes the deformation after the rigid motion has been removed. It is defined as:

$$\boldsymbol{\xi} = \frac{1}{2}(\mathcal{F}^T \mathcal{F} - \mathbf{I}) = \frac{1}{2}(\mathbf{c} - \mathbf{I}) \quad (2.9)$$

The deformation of the body occurs because of some external constraints. In this case, the internal forces tend to return the body into its rest position. For any part of the body, the total internal force equals to sum of all the forces acted on small volume elements. On the other hand, the external force is applied only to the surface of the element. Therefore the internal volume forces \mathbf{f} can be converted to a surface form [109]:

$$\iiint_V \mathbf{f} dV = \iint_{S(V)} \boldsymbol{\beta} dS \quad (2.10)$$

where dV is a unit volume, dS is a surface element. The tensor $\boldsymbol{\beta}$ that describes the nature of external forces is called the *Cauchy stress tensor*.

We guess that we consider elastic materials, which means the energy is not lost or transformed to internal state. In this case, the stress and strain tensors are connected to each other, and their relationship is described by a *strain energy density function* Ψ .

$$\boldsymbol{\beta} = \nabla_{\boldsymbol{\xi}}\Psi \quad (2.11)$$

Various strain energy density functions define various types of materials. Several most common hyperelastic materials were presented earlier.

Finally, to combine all deformation characteristics together, we have to set some equilibrium state for the system. Typically, a *principle of virtual work* (or *weak form of linear momentum*) is used. This principle states that we select only those displacements where the total internal work, given by the integral of elastic energy over the body, is equal to the total external work, given by application of volume and surface forces:

$$\int_{\Omega} \Psi_{app} dV = \int_{\Omega} \mathbf{u}_{app} \mathbf{f}_V dV + \int_{\partial\Omega} \mathbf{u}_{app} \mathbf{f}_S dS \quad (2.12)$$

where $\partial\Omega$ is the boundary of the domain, \mathbf{u}_{app} is the appropriate displacement, Ψ_{app} is the appropriate strain energy density function, \mathbf{f}_V and \mathbf{f}_S are volume and surface forces. It is also important to note that the displacement is compatible with the specified boundary conditions.

2.1.4.2 Deformation equilibrium simulation

Once an external impact is applied, the object undergoes some deformation until an equilibrium between external forces and internal stresses is achieved. The deformation goes the way to minimize the total energy inside the object. Since the system is nonlinear, it has to be solved using some iterative optimization method.

In this work, an implicit Euler method is used. The general equation of motion is obtained from the Newton's second law:

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{f}_{int}(\mathbf{u}, \dot{\mathbf{u}}) = \mathbf{f}_{ext} \quad (2.13)$$

where \mathbf{M} is the mass of the body, \mathbf{f}_{int} are internal forces, and \mathbf{f}_{ext} are external ones. Since $\mathbf{f}_{int}(\mathbf{u}, \dot{\mathbf{u}})$ is nonlinear, a Taylor series approximation over time t up to the first order is applied:

$$\mathbf{f}_{int}(\mathbf{u}_t, \dot{\mathbf{u}}_t) \approx \mathbf{f}_{int}(\mathbf{u}_{t-1}, \dot{\mathbf{u}}_{t-1}) + \frac{\partial \mathbf{f}_{int}}{\partial \mathbf{u}} d\mathbf{u} + \frac{\partial \mathbf{f}_{int}}{\partial \dot{\mathbf{u}}} d\dot{\mathbf{u}} \quad (2.14)$$

The matrix that describes the relationship between stress and strain or stiffness can be noted as \mathbf{K} . The derivative $\partial \mathbf{f}_{int} / \partial \mathbf{u} = \mathbf{K}(\mathbf{u})$. The derivative $\partial \mathbf{f}_{int} / \partial \dot{\mathbf{u}} = \mathbf{D}$ is called *damping matrix*, which is responsible for friction modeling. It is usually presented as combination of mass and stiffness:

$$\mathbf{D} = \zeta_M \mathbf{M} + \zeta_K \mathbf{K}(\mathbf{u}) \quad (2.15)$$

where ζ_M and ζ_K are some constants, called *reyleigh mass* and *reyleigh stiffness*. In addition, according to the numerical difference $d\mathbf{u} = \dot{\mathbf{u}}_{t-1} dt + \ddot{\mathbf{u}}_t dt^2$ and $d\dot{\mathbf{u}} = \ddot{\mathbf{u}}_t dt$. Substituting these equations in 2.13 and rearranging them, we obtain:

$$(\mathbf{M} + \mathbf{D}dt + \mathbf{K}(\mathbf{u}_{t-1})dt^2)\ddot{\mathbf{u}}_t = \mathbf{f}_{ext} - \mathbf{f}_{int}(\mathbf{u}_{t-1}, \dot{\mathbf{u}}_{t-1}) - \mathbf{K}(\mathbf{u}_{t-1})\dot{\mathbf{u}}_{t-1}dt \quad (2.16)$$

We can solve this equation to find $\ddot{\mathbf{u}}_t$. Then, using it, to obtain the new estimation $\mathbf{u}_t = \mathbf{u}_{t-1} + \dot{\mathbf{u}}_t dt$. The optimization is performed until the system converges to some final solution \mathbf{u}_{final} or until some maximum number of iterations N_{max} is reached. In the latter case, it is supposed that the data are invalid and the system cannot be solved.

A *corotational formulation* describes a special modification to model nonlinear behavior based on linear elasticity [156]. The authors split the general deformation on rigid rotation and linear elastic deformation; therefore, $\mathbf{K}(\mathbf{u}) = \mathbf{K}\mathbf{u}$. The internal forces are also expressed as two separate members that are linearly dependent on the strain and damping. To extract the rigid motion from the whole deformation, a stiffness decomposition onto orthogonal and upper triangular matrices (QR) is performed. Thus, the equations 2.13 and 2.15 are transformed to:

$$\begin{cases} \mathbf{M}\ddot{\mathbf{u}} + \mathbf{D}\dot{\mathbf{u}} + \mathbf{R}^T \mathbf{K} \mathbf{R} \mathbf{u} = \mathbf{f}_{ext} \\ \mathbf{D} = \zeta_M \mathbf{M} + \zeta_K \mathbf{K} \end{cases} \quad (2.17)$$

Then, this system is solved the way similar to the general nonlinear case.

2.1.5 The selected approach to simulate liver deformations

To simulate liver behavior various approaches rely on FEM with various hyperelastic materials. The choice of material depends on the requirements and is the trade off between real-time performance and accurate simulation.

The crucial element of our system is the real-time simulation. We think it is better to sacrifice the system accuracy, but observe the changes in real time, instead of predicting the behavior of the organ very accurately, but with a certain delay. The latter case makes the interaction with the system almost impossible. Therefore we are looking at the simplest hyperelastic materials since more complex ones are too time consuming.

Commonly, liver undergoes large deformations during surgery, and that means we have to model its nonlinear elastic behavior. We also suggest that the surgical manipulations are quite slow and, therefore, we do not need to consider the tissue viscosity. Consequently, there are several possibilities. We can use corotational FE formulation or nonlinear Finite Element Method with hyperelastic materials. The real-time context constrains us to consider only the simplest nonlinear models. So for the experiment we took several hyperelastic materials: StVK, Neo-Hookean, and Mooney-Rivlin.

To access posterior parts, surgeons usually bend anterior liver part, so we decided to compare liver models based on a bending deformation. We took a beam comparable with liver measurements ($15 \times 3 \times 3 \text{ cm}^3$) composed of approximately 26000 elements. We fixed it from one side and apply a bending load uniformly distributed across the inferior surface. For the Neo-Hookean material, we used parameters from a paper [214] ($\mu_1/2 = 0.2587 \text{ MPa}$), where the authors find values for a rubber material in case of various hyperelastic models using experimental stretch. For the Mooney-Rivlin material, we manually estimated parameter values ($\theta_{10} = 0.2112 \text{ MPa}$, $\theta_{01} = 0.0617 \text{ MPa}$) since the parameter related to the second invariant has to be positive. For more corresponding comparison, we also used the Young's modulus ($E = 1.542 \text{ MPa}$) computed from the Neo-Hookean parameter and the fact that the material is almost incompressible (Poisson's ratio is 0.49).

The results are presented in Fig. 2.6. They show that corotational formulation and nonlinear FEM with hyperelastic materials behaves the similar way. The maximal relative deformation difference for the corotational, Saint Venant-Kirchhoff, and Mooney-Rivlin does not exceed

3.2 %. For Neo-Hookean model, the relative difference is a bit bigger (almost 4.5 %). But their behavior differs a lot compared with behavior of a linear FEM model. The maximal difference is almost 102 %, which is comparable with the deformation displacement.

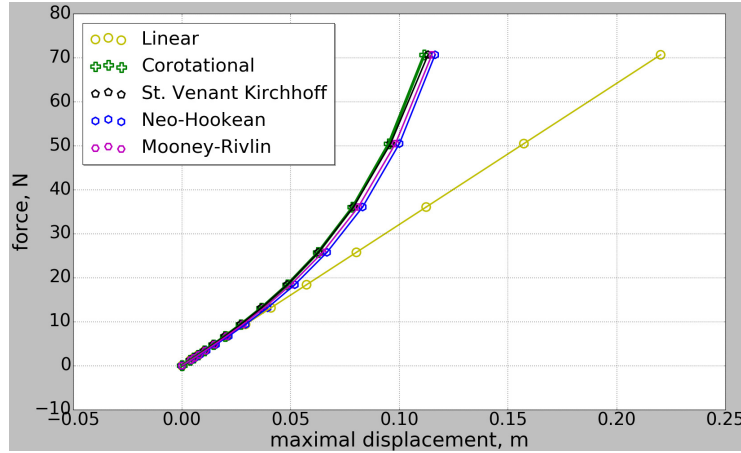


Figure 2.6: Linear and hyperelastic models behavior comparison for a bending deformation. For all cases, the same mesh, applied forces, and BCs were used.

To compare the performance, we computed the total time for 500 iterations of deformation. The average time for one iteration is presented in Table. 2.1. The fastest approach is FEM with corotational formulation. The hyperelastic model with StVK material is 5.53 % slower. The Neo-Hookean material is a bit more than 12 % slower, and Mooney-Rivlin is slightly slower (nearly 27 %).

| FEM type | Corotational | St. Venant-Kirchhoff | Neo-Hookean | Mooney-Rivlin |
|--|--------------|----------------------|-------------|---------------|
| Computation time for one iteration, ms | 402.73 | 425.02 | 451.64 | 503.68 |

Table 2.1: Computation time of models with different hyperelastic materials for a bending deformation. For all cases, the same mesh, applied forces, and BCs were used.

However, despite its simplicity, the corotational model is quite unstable, mainly because any deformation can be described by a certain combination of linear deformation together with rigid rotation. The more serious problem is that this instability might influence the estimation process which will result in worse-parameter estimation. All in all, the deformation difference between simulation results is small for all nonlinear tested variants, and StVK material shows a bit slower results than corotational formulation. Therefore we finally decided to select a hyper-elastic model with StVK material to simulate the deformations of the liver.

2.2 Liver boundary conditions

2.2.1 Importance of boundary conditions

Apart from mechanical properties of the object, its behavior is formed by a way the object is attached to the surrounding environment. Equations that interconnect the geometry of deformation with material strain, equations of equilibrium between stress and applied tensions or external forces, a constitutive law that interconnects strain and stress are necessary to describe the problem, but they are not sufficient. In addition, it is important to define what is happening on the boundary of the domain of interest [62, 79].

The boundary conditions are complement to the mechanics of the object; together they form the unified complete system. The combination of domain equations and BCs is called a *boundary value problem*. This problem is well known in different areas of physics and, particularly, in theory of elasticity. It is important to add that different BCs specify different boundary value problems. Thus, in order to avoid an issue of ambiguity, it is necessary to determine BCs.

To demonstrate the influence of boundary conditions on the object during deformation we show a simple experiment, illustrated in Fig. 2.7. We took a thin rectangle membrane

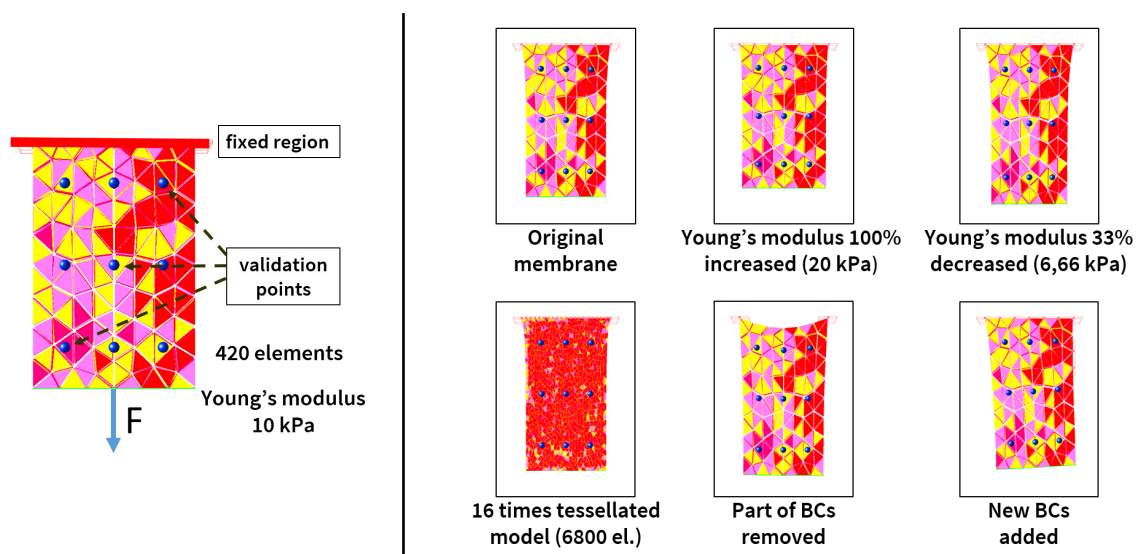


Figure 2.7: Left: membrane stretching experiment. Original conditions and model parameters. Right: visual overview of membrane stretching results under modified conditions.

$10 \times 15 \times 1$ cm, made of approximately 420 elements. To simulate an elastic behavior of almost incompressible tissue, the StVK material was used with the Young's modulus equals to 10 kPa and Poisson's ratio equals to 0.49. A fixed region was formed around one edge, and a uniform force was applied from the opposite edge. To compare the shape of the model, we used several "validation points" uniformly distributed on the front surface of the object. For comparison, we took scenarios where we changed the Young's modulus of the object, the size of the FE, and boundary conditions. After the deformation was imposed, we compared the positions

of points with respect to deformation for initial parameters. The main results are presented in Fig. 2.8. We can see that specification of positions for BCs has a comparable influence for the cases when the material stiffness is increased by 100 % or decreased by 33 %. At the same time, the size of FE has a much less influence (we increased 16 times the number of elements). Thus, these results show that BCs problem has the same rate of importance as the characterization of material properties.

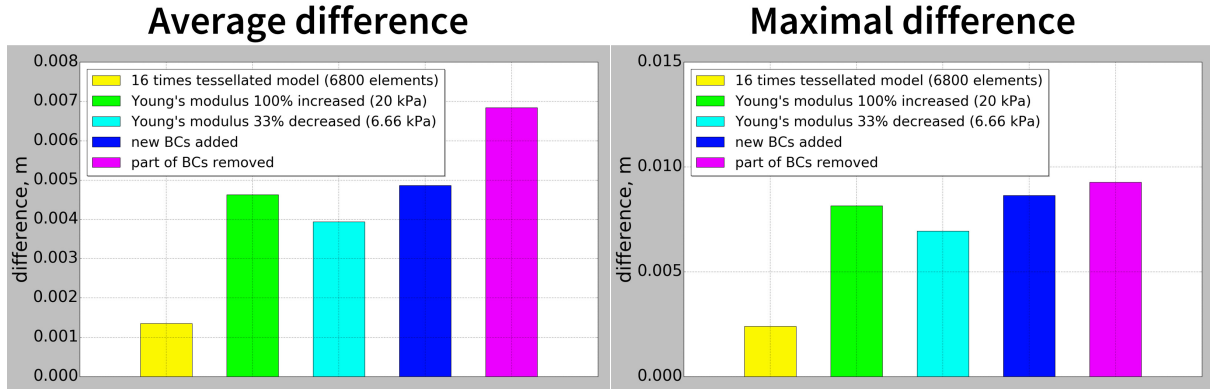


Figure 2.8: The difference between validation points for membrane stretched under original and modified conditions. The type of modification is given in the legend.

2.2.2 Theoretical description of boundary conditions

In formal theory, a boundary value problem is formed by a set of differential equations together with description of boundary conditions. Generally, differential equations have a family of solutions. So, one of them that satisfies the set of BCs is selected. It becomes the solution for a problem.

There are different types of boundary conditions, but all of them represent easy definitions for what happens on the boundary of the considered domain. To show them, let us suppose that we consider some domain Ω again, which has a boundary $\partial\Omega$, and a transformation function Φ that describes deformation for any point \mathbf{s} . We want to find the difference between deformed and rest positions $\mathbf{u}(\mathbf{s}) = \Phi(\mathbf{s}) - \mathbf{s}$.

Dirichlet boundary conditions *Dirichlet boundary conditions* are also called *first-type boundary conditions* or *fixed boundary conditions*. They are fixed in general sense, not in space, since they fix the solution on a boundary as a known function \mathcal{B}_D :

$$\mathbf{u}(\bar{\mathbf{s}})|_{\bar{\mathbf{s}} \in \partial\Omega} = \mathcal{B}_D(\bar{\mathbf{s}}) \quad (2.18)$$

Often the boundary function is defined with zeros to set the BCs fixed in space ($\mathcal{B}_D = 0$), but sometimes they have more complex description.

Neumann boundary conditions In *Neumann boundary conditions* or *second-type boundary conditions* the normal derivatives of the solutions on the boundary are fixed.

$$\frac{\partial \mathbf{u}(\bar{\mathbf{s}})}{\partial \mathbf{n}} \Big|_{\bar{\mathbf{s}} \in \partial\Omega} = \mathcal{B}_N(\bar{\mathbf{s}}) \quad (2.19)$$

where \mathbf{n} is the external normal to the boundary $\partial\Omega$ and \mathcal{B}_N is the known function. They can be considered as impulses obtained from forces and applied to the boundaries of the object.

Mixed boundary conditions In case of *Mixed boundary conditions* the domain is split on disjoint parts. On every part a Dirichlet or Neumann boundary conditions are specified. The definition, therefore, looks like:

$$\mathbf{u}(\bar{\mathbf{s}})|_{\bar{\mathbf{s}} \in \partial\Omega_1} = \mathcal{B}_D(\bar{\mathbf{s}}) \quad \left. \frac{\partial \mathbf{u}(\bar{\mathbf{s}})}{\partial \mathbf{n}} \right|_{\bar{\mathbf{s}} \in \partial\Omega_2} = \mathcal{B}_N(\bar{\mathbf{s}}) \quad \partial\Omega_1 \cup \partial\Omega_2 = \partial\Omega \quad (2.20)$$

Robin boundary conditions *Robin boundary conditions* are boundary conditions where the linear combination of the values and the normal derivatives of the solutions is specified on the boundary.

$$A_1 \mathbf{u}(\bar{\mathbf{s}}) + A_2 \left. \frac{\partial \mathbf{u}(\bar{\mathbf{s}})}{\partial \mathbf{n}} \right|_{\bar{\mathbf{s}} \in \partial\Omega} = \mathcal{B}_R(\bar{\mathbf{s}}) \quad (2.21)$$

where A_1 and A_2 are some nonzero constants and \mathcal{B}_R is the known description. If A_1 is zero, then Robin boundary conditions become second-type BCs; otherwise, if A_2 is zero, then they become first-type BCs. In more general form, the constants can be replaced by some functions.

To find a solution for particular problems in physics, more complex boundary conditions can be specified. But they are out of the scope of this work, and we are not considering them.

2.2.3 Anatomy of liver boundary conditions

The boundary conditions that are applied to the liver are formed by two types of constraints. First conditions are bilateral constraints. These constraints are active all the time. They are induced by connective tissues, ligaments (Fig. 2.9), blood vessels, and bile ducts (Fig. 2.10). The second type is unilateral constraints. They appear when the liver is in contact with surrounding organs such as stomach or diaphragm. This usually happens in case of tissue motion caused by respiratory or application of external forces during surgery.

The description of ligaments and other connective tissues can be found in [1, 9, 72, 157]. Generally, the liver is connected to the lower part of the diaphragm and to the anterior wall of the abdomen by five ligaments: *falciform*, *coronary*, *two lateral ligaments*, and *round ligament*. The liver is also attached to the lesser curvature of the stomach by the *hepatogastric ligament* and to the duodenum by the *hepatoduodenal ligament*.

The *falciform ligament* is a thin and broad antero-posterior peritoneal fold. It obliquely lies in an antero-posterior plane. The one surface of the ligament is in contact with the peritoneum behind the right rectus and the diaphragm, while the other is in contact with the left lobe of the liver. The falciform ligament is formed from two layers of peritoneum that lie close to each other [72].

The *coronary ligament* consists of upper and lower layers. The upper layer lasts from the upper border of the liver bare area to the lower surface of the diaphragm. It is the continuation of the falciform ligament layers. The lower layer is connected to the lower border of the bare area and the right kidney. Both layers are also formed by the peritoneum.

There are two *triangular ligaments*: right and left. The right triangular ligament is located at the right extremity of the bare area. The upper and lower layers of the coronary ligament

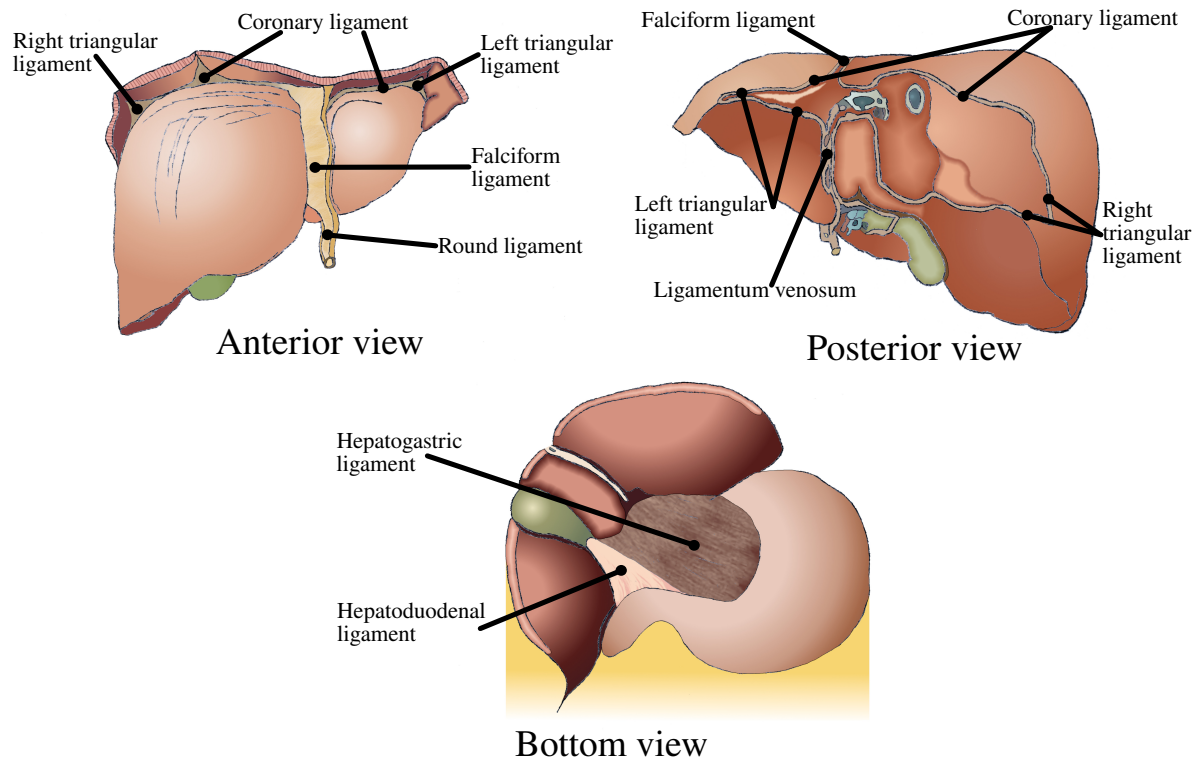


Figure 2.9: Liver ligaments anatomy.

finally turn into it, so the ligament looks like a small fold, connected to the diaphragm. The left triangular ligament is also a fold of considerable size. It connects the posterior part of the liver upper surface and the diaphragm. The anterior layer of left triangular ligament is transformed into the left layer of the falciform ligament [72].

The *round ligament* is a fibrous cord resulting from the *umbilical vein*. It lasts from the umbilicus, near the free margin of the falciform ligament, to the *umbilical notch* (the anterior border of the liver crossed by the falciform ligament) of the liver. Here, it may be tracked on the inferior surface of the liver to the porta where it combines with the ligamentum venosum.

Apart from ligaments, liver is also connected to surrounding organs with several blood vessels and bile ducts (Fig. 2.10). The blood vessels names are *hepatic artery*, *portal vein*, and *hepatic veins* [72, 157].

The *hepatic artery* and *portal vein* are connected to the porta, between the layers of the lesser omentum. It is also the place where the bile duct and the lymphatic vessels go away from the liver. The bile duct lies to the right, while the hepatic artery to the left, and the portal vein between them. The *hepatic veins* transfer the blood from the liver. The cells of liver binds them only closely to the walls of the canals through which they run. Thus, on the section of the organ, they remain widely open and may be easily distinguished from the vessels of the portal vein.

Several factors hold the liver in place. The first are the connections of the liver to the diaphragm, made by the coronary and triangular ligaments. Together with the connection of the inferior vena cava by the connective tissue and hepatic veins, they fix the posterior part of the

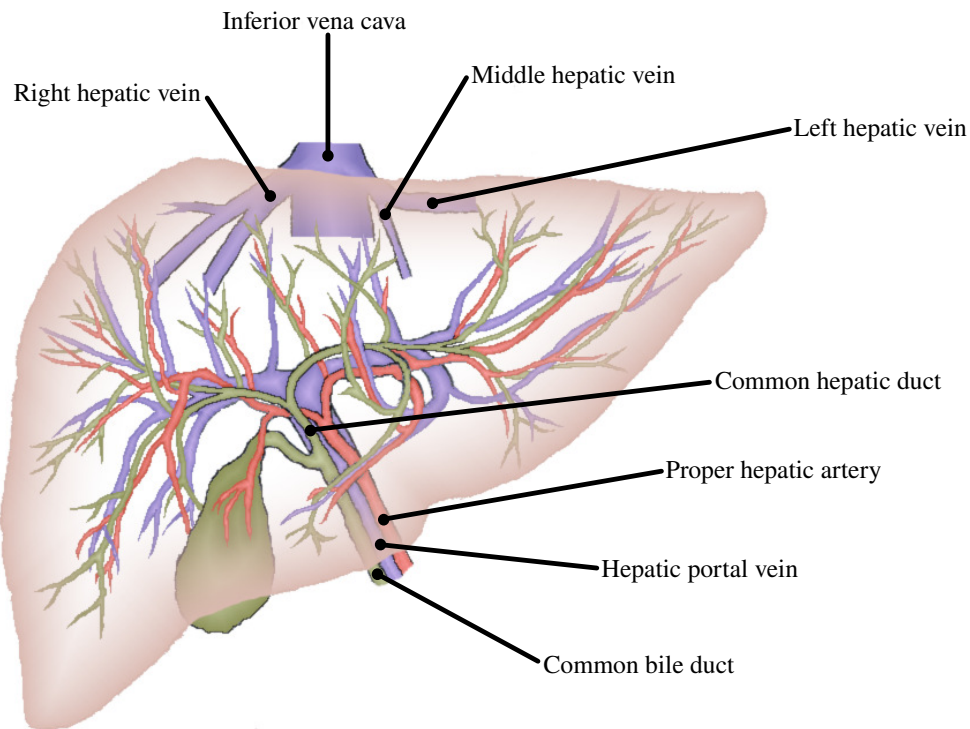


Figure 2.10: Liver vascular system description.

liver. Some support is provided by the pressure of the abdominal viscera. Generally, it completely fills the abdomen and its muscles are always in a state of tonic contraction. The superior surface is perfectly fitted to the bottom surface of the diaphragm. Therefore the atmospheric pressure is enough to hold it against the diaphragm. In its turn, the diaphragm itself is held up by the negative pressure in the chest. The falciform ligament does not give a lot of support, but it might limit the lateral displacement of the liver [72].

Therefore, the attachments around the liver are formed mainly by peritoneum, but, despite this fact, they are usually called as ligaments. Unfortunately, the information about liver attachments is limited to general description of their position. There is no information about neither their biomechanical properties nor the characteristics of connective structures, from which the ligaments are formed. A comprehensive overview about peritoneum is given in [85]. But the authors conclude that it is studied quite poorly since up to the recent time it was not considered as an independent connective tissue.

Another essential restriction for the determination of BCs is the impossibility to observe the whole liver organ during operation. In laparoscopic surgery the visible area is restricted by the camera, but even in open surgery the visible area is limited to the anterior part of the liver. Therefore it is quite complicated to specify the exact place where the connective tissues are attached to the liver. The area of attachment might also vary from one patient to another.

The unilateral constraints are also poorly described. Contacts between the liver and surrounding anatomy depend on organ shapes. The type of manipulations that are performed during

surgery also have an influence. But it is not known in advance what manipulations the surgeons will perform during liver resection for a particular patient.

2.2.4 Boundary conditions simplification

As we can see, the boundary conditions around liver are presented by various tissues [1, 9, 72]. Also, they might be active always or appear only under certain conditions. And so, it is difficult to describe them using simple equations.

However, for this work we decided to consider mainly ligaments attachments. First of all, unlike blood vessels, peritoneum folds cover various parts of liver. So, while arteries and veins predict the position of the liver in general, the ligaments have an influence on deformation behavior of various segments. Secondly, unlike blood vessels, ligaments are not visible on CT scans, and therefore we have to predict their position somehow. Thirdly, smooth muscles, which is the main component of blood vessels, are studied much better than peritoneum. So it is easier to construct an appropriate biomechanical model for them.

Anyway, in this work we are constructing the general model to describe the liver boundary conditions. It is important to mark that our final purpose is to accurately simulate the positions of liver tumors and internal structures. Therefore we need to focus on the impact of boundary conditions for the liver model, but not the internal structure and biomechanical behavior of the BCs. In addition, we can say that all external liver blood vessels are more or less surrounded by ligaments; therefore, they form a combined bilateral attachment for the liver. Consequently, we decided to consider a unified model of BCs, which is based on ligament biomechanics.

Talking about unilateral constraints, the main issue is that there is not enough information about positions and properties of the surrounding organs. Even having this information, it is difficult to understand their influence on the liver shape and the final amount of contribution in behavior prediction. So, for simplicity, in this work we decided to ignore them.

2.3 The fundamentals of ligaments modeling

2.3.1 Ligaments anatomy

Liver ligaments are formed mainly by peritoneum. The peritoneum is a smooth tissue (serous membrane) consisting of two layers of mesothelium that secrete serous fluid, together with a thin layer of connective tissue [85]. The inner layer that covers organs (viscera) in body cavities is called the visceral membrane. A second layer of epithelial cells formed from the serous membrane, called the parietal layer, lines the body wall. Between the two layers, there is a potential space, mostly empty apart from a few milliliters of lubricating serous fluid secreted by the two serous membranes. The peritoneum covers most of the intra-abdominal (coelomic) organs. This peritoneal lining of the cavity supports many of the abdominal organs and serves as a conduit for their blood vessels, lymphatic vessels, and nerves. Around the liver, the folds are formed by the inner layer (visceral) of peritoneum. And the outer layer (parietal) of peritoneum, connected to the diaphragm and posterior abdominal wall, attach them to the liver.

Typically the peritoneum is regarded as a protective barrier and frictionless interphase that covers the abdominal area and organs, but it has much more complex structure with a great variety of functions. Apart from participating in the embryogenesis of primitive gut, peritoneal functions also include selective fluid and cell transport, physiological barrier, immune induction, modulation, and inhibition, tissue repair and scarring, preventing adhesion and tumoral dissemination migration [85].

Addressing the biomechanical properties of the human peritoneum, a review of the literature does not provide any information about it. A few experiments were performed to study the mechanical properties of porcine [246] and bovine [207] peritoneum, but they describe only stress-strain curves and membrane parameters obtained from stretching experiments (Fig. 2.12). Since there was no attempt to create a model for peritoneum, we decided to consider the ligament description in general and give an overview of approaches that attempts to simulate them accurately.

The good description of ligaments and tendons model is given in [113, 200]. Ligaments are multiphasic composites. They are composed of solid substance, which is presented mainly by collagen, and water. The collagen is the main structural protein in the various connective tissues. In ligaments, it is organized in a complex hierarchical structure (Fig. 2.11). Collagen threads are tied in *tropocollagen monomers* to form *fibrils* (50 nm – 200 nm diameter) at nanoscale level. Then fibrils form *fibers* (2 μm – 50 μm diameter) at the microscale level. Fibers are arranged mainly in parallel fashion. At the mesoscale level, fibers are assembled into *fascicles* (5 μm – 500 μm diameter), and, finally, fascicles form the whole ligament at the macroscale level (100 μm – 1 mm diameter). Ligament components, such as fascicles and fibers, are surrounded by a thin fascia.

The extracellular matrix of ligaments is formed by proteins and consists of water on approximately 70 %. The solid part is primarily composed of type I collagen (60–80 %), with the remainder consisting of elastin, proteoglycans and glycosaminoglycans, other types of collagen (types III, IV, V, VI), fibrillin, and other proteins [200].

Therefore, ligaments can be presented as collagen threads that are bundled in various fiber structures and surrounded by a liquid.

2.3.2 Biomechanics of ligaments

As was already said, the human peritoneum is studied very poorly. The experiments that were performed covers only mechanical properties of animal peritoneum [207, 246]. But they give a general idea about its stress-strain relationship (Fig. 2.12).

On the other hand, the properties of other human ligaments are studied in more details [200]. An analysis performed by Witz *et al.* [249] shows that the peritoneum shares several similarities with ligaments. Consequently, we choose to investigate the biomechanics of "true" ligaments for our work. Their general stress-strain curve is presented in Fig. 2.12. It is important to mark that ligaments response to stress is very dependent on the properties of fibers, which form the ligaments. In rest state, ligaments contain twisted fibers, which resistance for small strains is limited. As fibers are straightened, the stress increases rapidly, leading to a stronger nonlinear stress-strain relationship.

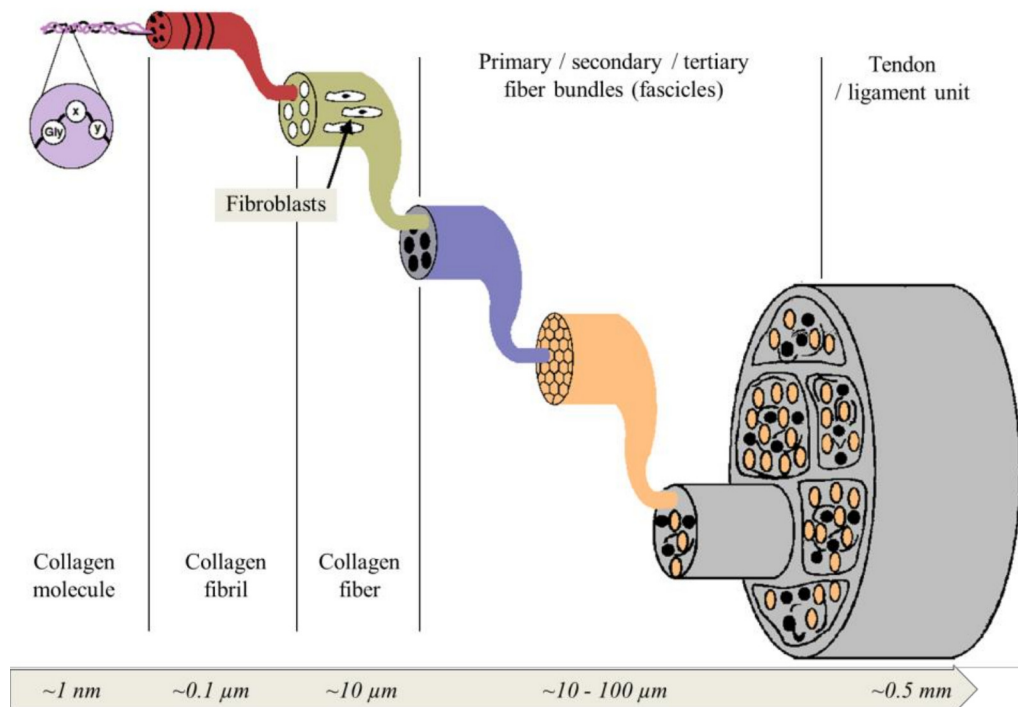


Figure 2.11: Anatomy of a typical ligament. The internal fibrils and fibers are organized in a hierarchical structure. Image taken from [113].

The experimental results to biomechanical response of a ligament for different kinds of deformation is presented in Fig. 2.13. Generally, the tensile stiffness along the fiber direction is an order of magnitude stiffer than in the transverse direction. In shear, the ligament is two orders of magnitude more compliant than in transverse direction. Also, in compression, the ligament is over three times more compliant than in tension.

To conclude, the ligaments can be described by a transversely isotropic model, which is stiff in the direction of fibers, but quite soft in other directions. Because the fibers resist stretching, but do not resist any kind of bending. The model can also be compressed easily since the fibers are twisting during compression.

2.3.3 Ligaments simulation

Unlike liver, ligaments are not so well studied. Most researchers consider ligaments as a component of joints in human body. It mainly includes ankle and knee joints, but also shoulders and arms. The works that study ligament as a separate tissue are quite seldom.

2.3.3.1 Hyperelastic materials

There are several hyperelastic methods that were proposed to simulate ligaments behavior.

The most commonly used approach is to present a material as combination of two components and simulate its behavior using FEM. The first part describes the properties of the extra-

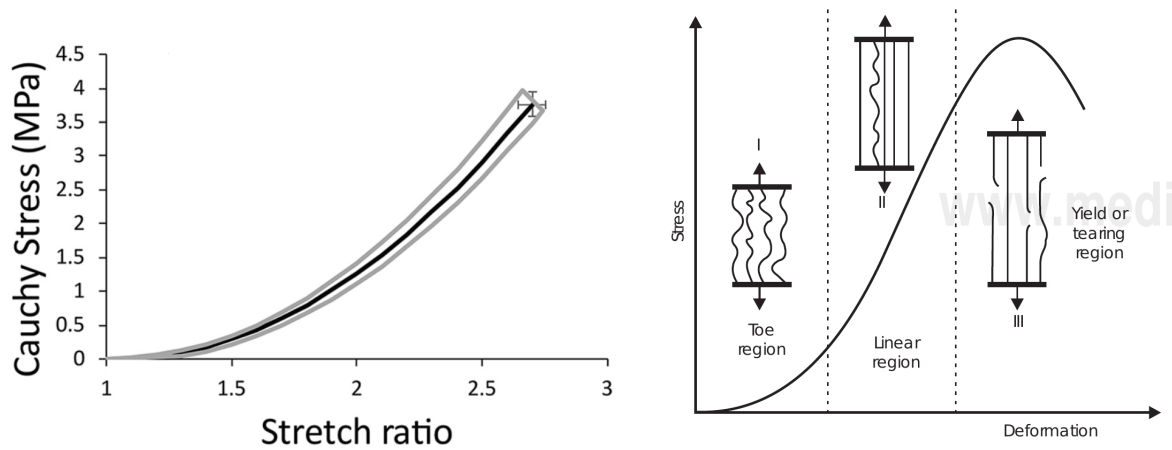


Figure 2.12: Overview of the experimental porcine peritoneum stress-strain curve (left image) taken from [246] and the theoretical ligament constitutive law (right image) taken from [203]. The constitutive law of ligaments is very dependent on collagen. After the fibers untwisting and a short linear response, fiber elasticity rapidly reaches a limit where microscopic failures appear.

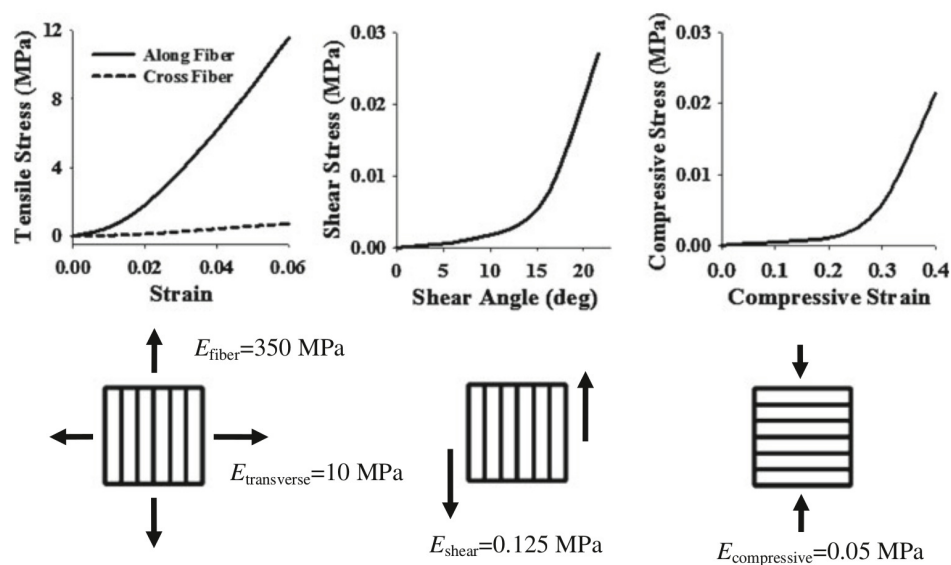


Figure 2.13: The elastic behavior of ligament for different types of deformations (E is the Young's modulus). The nonlinear tension stress is much larger than compression or shear stress. Image taken from [200].

cellular matrix, while the second part characterizes the fiber behavior. In this case, the strain energy density function Ψ is presented as:

$$\Psi = \Psi_m(I_1, I_2) + \Psi_f(\lambda) \tag{2.22}$$

where Ψ_m is strain energy for the extracellular matrix, I_1 and I_2 are first and second material

invariants, Ψ_f is strain energy for collagen fibers, λ is stretching ratio along the local fibers direction.

Commonly, to model the extracellular matrix an isotropic hyperelastic material is used. For example, in [193, 244, 120], the authors use the Neo-Hookean constitutive material. These papers present study of the anterior cruciate ligament in combination with the medial collateral ligament. Some other works use more complex materials. In [195], the Mooney-Rivlin model is proposed, and in [242], the authors use the Veronda-Westman one.

The second part simulates fiber behavior as transversely isotropic hyperelastic material. For its modeling, a special component is proposed, defined as a function of the fiber stretch [244]:

$$\lambda \frac{\partial \Psi_f}{\partial \lambda} = \begin{cases} 0 & \lambda \leq 1 \\ c_1 (\exp[c_2(\lambda - 1)] - 1) & 1 < \lambda < \lambda^* \\ c_3 \lambda + c_4 & \lambda \geq \lambda^* \end{cases} \quad (2.23)$$

Here c_1 scales the exponential stress, c_2 specifies the rate of collagen uncrimping, c_3 is the modulus, and c_4 is the additional parameter of straightened collagen fibers, λ^* is threshold stretch, at which the collagen is straightened. The material constants c_1 , c_2 , c_3 , and c_4 are determined from stretching experiments.

In [49, 70, 241, 257], to study the behavior of ligaments, the authors use the similar combination. However, to make a compressible material, they add an additional member to the strain energy density based on bulk modulus. The same model is used in [173, 239] to investigate the kinematics of a knee joint. In [119, 121], to simplify the response of fibers, the researchers remove the linear part (the third element) from the equation 2.23.

The alternative models are more rarely used, but researchers still propose their own variants of ligament constitutive laws. For example, the authors from [58] investigate and simulate ankle ligaments. The strain energy density function is composed of three parts:

$$\Psi = \Psi_{mi}(I_1) + \Psi_{mv}(J) + \Psi_f(I_4) \quad (2.24)$$

where Ψ_{mi} is responsible for isovolumetric behavior, Ψ_{mv} describes incompressibility of the tissues, and Ψ_f characterizes ligament fibers. They are defined as:

$$\begin{cases} \Psi_{mi}(I_1) = \frac{c_1}{\alpha_1} (\exp[\alpha_1(I_1 - 3)] - 1) \\ \Psi_{mv}(J) = \frac{K_v}{2+r(r+1)} [(J-1)^2 + J^{-r} + rJ - (r+1)] \\ \Psi_f(I_4) = \frac{c_4}{(\alpha_4)^2} (\exp[\alpha_4(I_4 - 1)] - \alpha_4(I_4 - 1) - 1) \end{cases} \quad (2.25)$$

where J is the deformation Jacobian, which specifies the volume change, I_1 is the first principal scalar invariant, I_4 is an invariant that characterizes the transversely isotropic material [80]. The c_1 and α_1 are material parameters that describe the shear stiffness, K_v and r are material parameters characterizing the material compressibility, c_4 defines fiber initial stiffness, and α_4 characterizes wavy fiber conformation.

In [78], to model the anterior cruciate ligament for a human knee, the authors propose to simulate the extracellular matrix as Mooney-Rivlin material and present fibers as two families of connections with bilinear stress-strain relationship.

Some researchers use more specific models to simulate ligaments. Lanir et al. [111] proposed a Lanir model. The fiber tissues, including ligaments, are presented as networks of different fibers incorporated in a fluid intracellular matrix. In [195], the authors show that Lanir model simulates ligament deformation with the satisfactory accuracy. The authors from [94] to learn the behavior of a vocal ligament use Gasser-Ogden-Holzapfel model described in [71]. The parameters for it are obtained from histology of tissue and biomechanical experiments.

In [220], to simulate human knee ligaments, the authors decided to use a unified material for the extracellular matrix and fibers. So, they rely on a Veronda-Westman material, which describes exponential stress-strain relationship.

Some papers attempts to simplify ligament models. It is usually done when the work is focused on the modeling of ligaments impact rather than their accurate simulation. In [73], the authors propose to model ligaments using beam elements. In other works, ligaments are presented as nonlinear springs [117, 248], line elements (squared springs) [147], or straight line fibers [256].

2.3.3.2 Adding viscosity

The papers describing ligaments viscosity modeling are not very numerous. They can be classified into two groups.

In the first group, the authors rely on the quasi-linear viscoelastic model, which is added to the given hyperelastic constitutive law. The researchers use it to simulate, for example, an ankle ligament [66], a medial collateral ligament [193], or an anterior cruciate ligament [39].

In the second group, the authors construct their own nonlinear models. It is worth marking that the general nonlinearity implies the combined expression to simulate the viscoelastic stress response. Consequently the constructed models are quite complex, despite assumptions and simplifications made by researchers [37, 58, 174, 185].

The attempts to include even more complexity in the ligament models are rare; however, some researchers mark that ligaments have complex behavior. For example, in [243], the authors claim that some ligaments such as human medial collateral ligament have permeability properties.

2.3.4 The model for ligaments

As described above, the most of the proposed approaches simulate ligaments as FEM with complex hyperelastic or viscohyperelastic materials. But the general idea in most of the works is to understand how ligament influence the joint connection and its possibilities to bend or twist. In current work, there is no need for a detailed ligaments simulation. The final goal is to create a model for an accurate liver simulation, but not its surrounding environment. On the other hand, the crucial moment is real-time performance, which is obviously not the case for complex simulations. Therefore we have to look for simplified models that can simulate impact behavior. Among others, there are several papers where ligaments are modeled with simplified models [73, 117, 147, 248, 256].

Membranes are very thin structures, which are very hard to segment, assuming they are visible in the preoperative images. Therefore it is quite difficult to construct the appropriate mesh for ligaments. Their geometry calls typically for a surface discretization. Flat elements, such as shells or triangles, are usual options for modeling such structures while accounting for their small thickness. It is also possible to model such thin structures using volume elements though the size of the elements has to be small. Also, for accurate simulation, a sufficient number of elements have to be created, and, to process simulation, these models require enough computational resources.

So to model thin membranes we address the theory of plates. This theory describes deformations of very thin objects and provides a mechanism, which employs the in-plane coordinates in a three-dimensional coordinate system as independent variables. There are several options to build such type of models.

One option is flat elements. It is a way to model the thin object based on the simplest idea. The object is presented as a set of two-dimensional elements. Every element describes elastic behavior, similar to volume FEM, but only on the surface defined by its vertices. However, the elements might rotate around each other independently. Among the advantages we can mark the simple integration with general Finite Element Method. The main drawback is computation complexity. It is similar to the volumetric FEM, which has comparable element sizes and the single layer of elements for depth. We will now give the brief description of the main possibilities.

Constant strain triangle. Constant strain triangle is the simplest flat element that describes elastic deformation. The domain is discretized using triangles. Every triangular element is characterized by six parameters. Three of them are responsible for rigid body motions, and three others indicate the strain. Such elements are capable of representing only states where the strain within the element remains constant [43]. Consequently, constant strain triangles are very simple and computationally effective. But the elements can simulate only linear deformations. So, they are valid only for small strains, and to obtain an accurate result the element size has to be small enough.

Linear strain triangle. The enhancement of the previous approach is linear strain triangle where the idea is to use triangular elements of high order. The element has six nodes and twelve degrees of freedom, which allows it to represent all linear variations in the strain [42]. Linear strain triangle allows simulating much more complex material deformations, but they are more computationally expensive. Additionally, to use them, we have to identify extra model parameters.

Shell elements. A shell is a three-dimensional elastic body that is characterized by its middle surface and thickness. There is a huge theoretical background behind the shells. Unlike plate elements, shells can simulate more complex deformations like twisting [30]. To perform simulations with shell elements, scientists use different approaches. For example, in [32], the authors proposed to use triangular shell elements. They get shell elements by combining an in-plane energy, which is described by a typical triangular FE, and off-plane energy to describe bending and twisting. The latter part is presented as an additional matrix and integrated into the final equation. The main advantage of this method is that shell elements allow simulating complex

deformations. On the other side, it has excessive complexity for scenarios with a simple object behavior.

Another general way is to approximate the membrane behavior. In this case, the membrane model is replaced with a set of simple elements that behave the similar way. This allows performing a fast computation, but in general, we lack the simulation accuracy. However, when we have to simulate the influence of the component on the system rather than the component behavior itself, it might be a possible choice.

Mass-Spring Model. Mass-Spring Model (MSM) is an approach where the object is presented as a set mass points connected with springs. The points depict the position of the object in space, and springs are responsible for simulating the properties of material [93]. This makes them computationally efficient and able to handle large deformations and topological changes with ease. The main drawback of MSM is that they do not necessarily describe the mechanical behavior of actual deformable bodies accurately because the direct link between material properties and spring parameters is absent.

Beam elements. A beam element is a model of a straight bar with a uniform cross section. It can resist axial, shearing, bending, and twisting forces [191]. In beam elements approach the object is presented as a combination of such elements. The stiffness matrix that describes the beam is 12×12 since it takes into account the angular and spatial positions of each its end. The beam element model allows very accurate modeling of material parameters in one specific direction. So it is usually used to model objects, the properties of which can be neglected in all directions apart from the principal one, such as catheters [46]. Among the drawbacks, it is worth marking the difficulties related to an extension of this model to simulate the properties of objects in more than one direction.

Among available approaches, without interest to model the exact location of ligaments, we decided to choose a Mass-Spring Model. It is important to note that springs do not simulate the behavior of real boundaries; they only mimic elastic attachments to particular areas of the liver surface. Also, in case of nonlinear springs, it is quite easy to simulate the desired constitutive law by just fitting spring parameters.

We choose to use a cubic spring for approximating the constitutive law of ligaments. In such springs, the force f applied by each spring is given as:

$$f = \begin{cases} \left(k_{\alpha}|\Delta l|^3 + k_{\beta}|\Delta l|^2 + k_{\gamma}|\Delta l| \right) \frac{\Delta l}{|\Delta l|} & \Delta l > 0 \\ 0 & \Delta l \leq 0 \end{cases} \quad (2.26)$$

where $k_{\alpha}, k_{\beta}, k_{\gamma}$ are the stiffness parameters for a spring and Δl is the length variation. In addition, $f = 0$ when $\Delta l < 0$, which represents the fact that ligaments only generate sufficient forces on the liver under extension.

Talking about space location, for every spring we decided to attach one of its ends to the liver and fixed another one. The spring direction corresponds to the liver surface normal where the spring is attached. The length of the springs was taken based on measurements of the human falciform ligament, available in [118].

From a theoretical point of view, our BCs for liver form a complex model since the springs make a combination between impulses and displacements. However, if we consider MSM as a

part of our model, then BCs of the combined system can be described as Dirichlet boundary conditions with predefined zero displacements.

Since MSM might not describe the mechanical behavior of actual deformable tissues accurately, we decided to compare the MSM with the more accurate FEM. We perform the comparison based on a stretching experiment.

We looked through experimental curves in [246] that describe parameters of porcine peritoneum. The human ankle [58] and knee [184] ligaments are well studied, so we also took into account the available experimental data. Based on considered constitutive laws, we selected experimental data related to the softest and stiffest samples (Fig. 2.14). Then we identified the

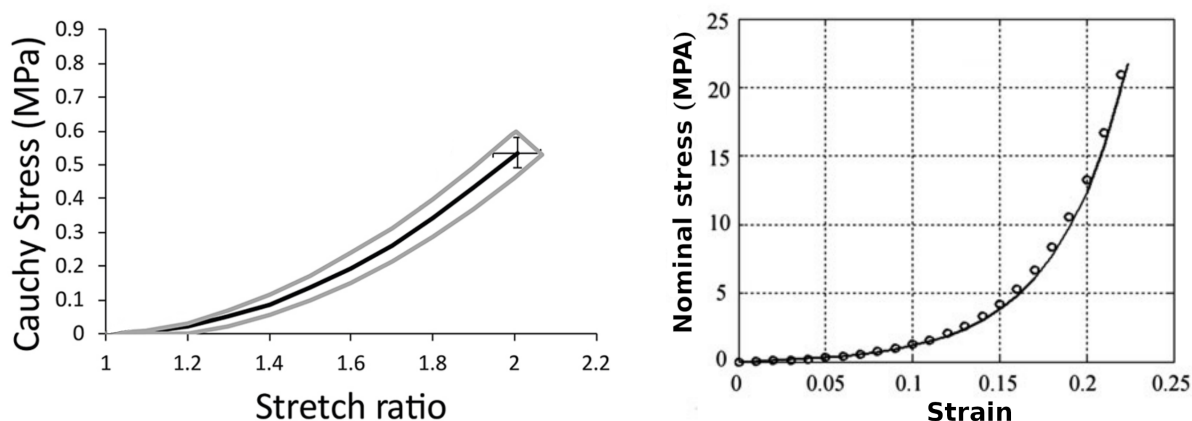


Figure 2.14: The selected experimental stress-strain curves related to the softest sample (left image) taken from [246] and the stiffest sample (right image) taken from [58]

stiffness of the Neo-Hookean material, which behaves similarly to the given stress-strain curves. For this, we used curve fitting approach based on Levenberg-Marquardt algorithm available in FEBio software [132]. We got the Young's modulus of 403 kPa for the softest sample and 72 MPa for the stiffest one. To take into account the possible measurement deviations, we increased the diapason of possible values; therefore, we took 150 kPa as minimal value and 500 MPa as maximal one.

Then we created a thin rectangle shape ($40 \times 100 \times 1 \text{ mm}^3$) composed of approximately 3700 elements, which we initialized using Neo-Hookean materials with minimal and maximal Young's moduli values. A layer of springs with the same rectangular size was generated (Fig. 2.15). We assimilated spring parameters based on Neo-Hookean model stretching and using the reduced-order unscented Kalman filter approach described in the Chapter 3. Then we compared results. Since finally we are interested in predicting the position of the liver, but not ligaments, we compared only the nodes in the region where the ligaments are attached to the liver, marked as yellow spheres in Fig. 2.15.

The results show an average error between both models of less than 2.7 mm in a stretching scenario for both cases and strains up to 45 % (that is, covering the range of ligaments strain in surgery). The maximal error in the same diapason of strains does not exceed 4 mm (Fig. 2.16). We suggest that the error rate is the same for Neo-Hookean materials with Young's modulus

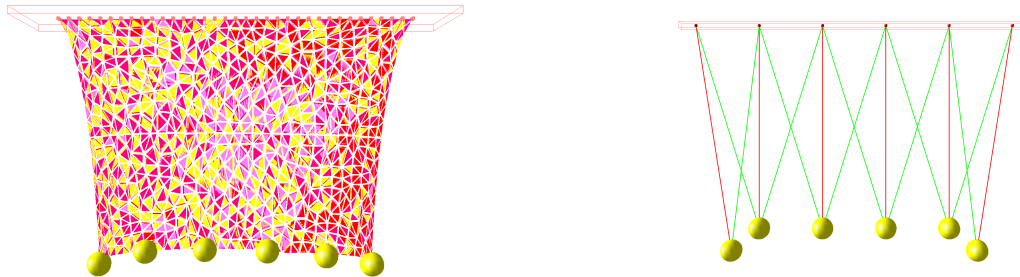


Figure 2.15: MSM stretching experiment description. A thin rectangle shape modeled by FE with Neo-Hookean material and the estimated Young's modulus was used as a reference model (left image). The MSM with diagonal springs were constructed and the parameters were assimilated from the FE model (right image).

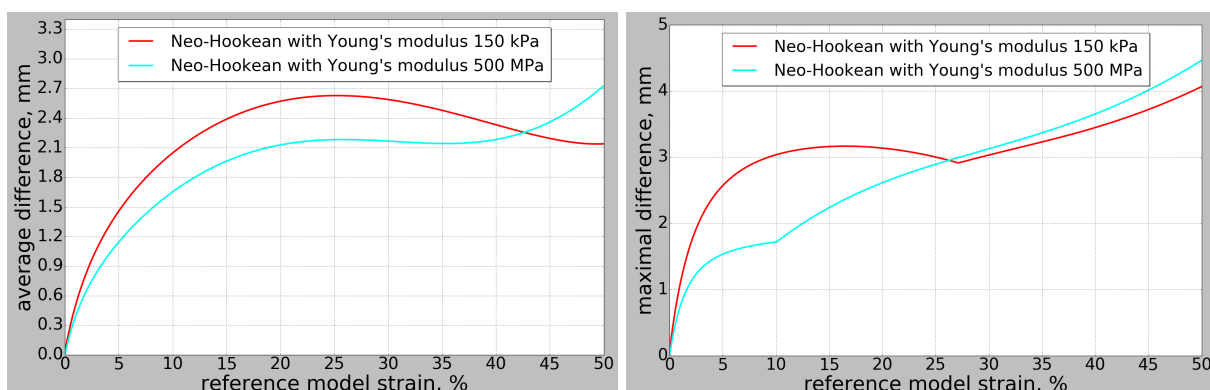


Figure 2.16: Comparison for stretching deformation. The average and maximal difference is computed for FEM with Neo-Hookean material and the indicated Young's modulus and MSM with assimilated springs coefficients.

values in between the considered limits. This indicates that the behavior of our mass-spring system is similar to a Neo-Hookean material.

We also compared the computation time between models. To show the performance of the Mass-Spring Model, we also added the hyperelastic Finite Element Method with StVK material, which is one of the fastest nonlinear FEM. The simulation time was computed for 1000 iterations, and the results are presented in Table. 2.2.

| Model type | Cubic MSM | St. Venant-Kirchhoff FEM | Neo-Hookean FEM |
|--|-----------|--------------------------|-----------------|
| Computation time for one iteration, ms | 0.847 | 24.119 | 28.138 |

Table 2.2: Stretching simulation computation time. For FEM, a mesh with 3700 elements were taken, and MSM was composed of 6 edge springs and 12 diagonal ones (Fig. 2.15).

The Mass-Spring Model that replaces the model with almost 4000 elements is 28.5 average times faster FEM with StVK material and 33 times faster the model with Neo-Hookean material. Therefore, it makes reasonable to replace complex models with simpler MSM to achieve a real-time performance.

As a result, we want to say that cubic springs are a good choice in the context of our work. Unlike liver model, we are not interested in exact positions of ligaments, and Mass-Spring Model is one of the fastest approaches that allows us to simulate the behavior of an object. In AR systems, real time is one of the crucial moments, which determines our choices for the specified problem. And, consequently, we stick to usage of the mass-spring system for the ligament behavior simulation.



DATA ASSIMILATION AND KALMAN FILTER



Contents

| | |
|---|-----------|
| 3.1 Overview of the parameters estimation procedures | 50 |
| 3.2 Kalman filtering approach and its extensions | 51 |
| 3.2.1 Kalman filtering approach | 51 |
| 3.2.2 Reduced-order unscented Kalman filter | 55 |
| 3.2.3 Ensemble transform Kalman filter | 58 |
| 3.3 Software module for data-driven simulation | 62 |
| 3.3.1 Prediction procedure | 64 |
| 3.3.2 Correction (Analysis) procedure | 65 |
| 3.4 Estimation using Kalman filtering | 67 |
| 3.4.1 Object stiffness estimation | 69 |
| 3.4.2 Estimation of boundary conditions | 72 |
| 3.4.3 Correction of boundary conditions | 78 |

As shown in the previous chapter, the liver attachment to surrounding organs has not been well studied, and, thus, it is difficult to construct a model that will describe their behavior. So, we need to identify them. During both open and laparoscopic surgery, the surgeons use cameras, which provide them with some information about liver shape and its modification. Thus the main idea is to identify BCs based on these movements, while performing various manipulations. In this chapter, we will give a brief overview of the available methods that allows us to perform such estimation. Then we will focus on the Kalman filtering approach, which we believe is a best choice under given restrictions.

3.1 Overview of the parameters estimation procedures

Generally, biomechanical characteristics of organs vary significantly for different persons. Consequently, to estimate model parameters, we can rely only on information obtained from specific patient i.e., intra-operative surgical images. However, they contain certain measurement errors. Moreover, they provide us only with limited range of scope. Therefore we need an approach that is able to deal with the appeared complications. We considered several possible directions that are presented further.

Neural networks. For estimation purposes, neural networks combined with deep learning algorithms are one of the most popular approaches nowadays. They are used in various domains of science: image processing, pattern recognition, and so on. Much literature is dedicated to this topic, for example [2, 75]. The general idea is to train a network on available data to predict a process behavior or a structure of the system in a similar scenario. But for training, some data is needed. In general, there has to be a huge amount of available data. Unfortunately, due to various personal information privacy rules, the description of the liver boundary conditions as well as intraoperative surgical recordings are unavailable publicly. Therefore this approach is hard to apply under current circumstances.

Variational data assimilation. Variational data assimilation is a family of approaches that are derived from the theory of optimal control [10]. The main idea of these approaches is to construct a cost function, which includes the model and real world measurements with some uncertainties. The solution of the problem is then obtained by gradient descent that minimizes the cost function. The main limitation of this approach is related to the context of our problem.

There are two main branches in the family: 3D-Var and 4D-Var.

In case of 3D-Var, also called the stationary case, the cost function is based on the covariance of the constructed system. Therefore, we need to know it, but in our context the covariance is strictly connected to the unknown stiffness, which we are trying to estimate. This limitation makes 3D-Var inapplicable for our estimation.

In 4D-Var (the nonstationary case), the data assimilation is based on observations that are acquired over a certain time interval, and the cost function is constructed for the sequence of process states. In order to obtain the gradient of the cost, we have to perform backward simulations. However, due to very limited available information, we want to use as many available real observations or measurements as possible, but the estimated data has to be used in surgical process, which provides us with the observations. Under this restriction, the way and the time we have to construct the cost function remain the open question. Of course, different solutions could be proposed, but Bayesian inference methodology described below allows us to update naturally the previous optimal estimation as soon as new data becomes available.

Bayesian inference. The Bayesian inference (BI) is a statistical inference that allows improving the estimation sequentially. In its essence, the Bayes' theorem is used to update the probability for a hypothesis when more evidence or information becomes available. BI incorporates the statistical noises of experimental devices and provides a statistical regularization, which makes inverse problems with limited observations solvable. Due to the nature of our process, for the correction, we decided to choose a Bayesian inference methodology.

There exist several implementations related to the Bayesian inference.

One group is based on the Monte Carlo method [206], which allows presenting complex non-Gaussian distributions by sampling the probability density functions. For example, this is valid for a Particle filter (Sequential Monte Carlo) [26]. These methods allow simulating various complex probabilistic distributions, but their main drawback is quite huge computational cost.

On the other side, one of the fastest implementations is the Kalman filter [92] and its extensions: unscented Kalman filter [91] with modifications and ensemble Kalman filter [52] that are able to process nonlinear systems. In Kalman filtering context, the estimated unknown data is called the *state* of the system. The probability of unknown data is presented as a Gaussian distribution with some mean and variance. The nice property of the Gaussian distribution is that it can be transformed through a nonlinear system with just a couple of sample points. For multidimensional space it can be reduced even further by combining information related to several unknowns into a single sample.

The different modifications of the general filtering approach are applied to solve problems in various domains, including biomechanics. For example, a reduced-order unscented Kalman filter is applied to estimate parameters for a vascular model [145] or a cardiac system of a specific patient [137, 146]. This work is also based on [180], where the authors proposed to use such a filter for BCs estimation.

Thus, the Kalman filtering method combines the possibility to work with uncertainty, the way to sequentially, and therefore fast, update the optimal estimations when new data becomes available, and the computational efficiency with potential to work in real time. So, among all possibilities, we decided to stick to the estimation correction based on a filtering approach.

3.2 Kalman filtering approach and its extensions

3.2.1 Kalman filtering approach

The Kalman filtering or linear quadratic estimation approach is one of the best-known methods for data assimilation in case of measurements with some noise or other inaccuracies. The unscented extension requires to compute neither inversion of a matrix that describes the system, nor the derivative of the system. This makes the filtering approach very attractive to the complex ill-posed problems, which usually have limited amount of available information.

In general, the Kalman filtering approach deals with a dynamic process of a system, which occurs during certain periods of time. We have an imperfect model that describes our system, and we want to estimate some unknown parameters or *state* of the system, which is not directly measurable. Instead of that, some noisy data is available that has a certain relationship with the state. This data is known as *observations*. Both state and observations are presented as random variables. They reflect the fact that we work with the hypothesis from theory of probability. Also, it is important to note that the process is discrete in time. In case of continuous one, it can be discretized. For a nonlinear system, the process with the state can be determined via mathematical equations as:

$$\begin{aligned} \mathbf{x}_{i+1} &= \mathcal{T}(\mathbf{x}_i, \mathbf{b}_i) + \mathbf{q}_i, & \mathbf{q}_i &\sim \mathcal{N}(0, \mathbf{Q}) \\ \mathbf{z}_{i+1} &= \mathcal{H}(\mathbf{x}_{i+1}) + \mathbf{v}_i, & \mathbf{v}_i &\sim \mathcal{N}(0, \mathbf{V}) \end{aligned} \quad (3.1)$$

here \mathbf{x}_i is the unknown state with size N_x , which we want to estimate, \mathbf{b}_i is some control sequence, \mathbf{z}_i is the observed data or measurements, \mathcal{T} is a transformation operator that describes the process flow, which in given work is a FE based simulation model, \mathcal{H} is an *observation operator* that describes a relationship between process and measurements, \mathbf{q}_i is a process noise, and \mathbf{v}_i is a measurement noise. The process and measurement noises are assumed to have a Gaussian distribution \mathcal{N} with zero mean and covariances \mathbf{Q} and \mathbf{V} , respectively.

In its essence, the Kalman filtering approach is a loop that contains two parts [245]. Before estimation, we take some preliminary approximation of our unknown state. The first part, called *prediction*, relates to the simulation of the next step using current estimation of unknowns. The second part performs the *correction*, also called *analysis*, for the unknowns based on the difference of modeled and real system behaviors. The difference describes the discrepancy between predicted and real observations and is called *innovation*.

For the prediction part, the next step of the simulation process is predicted based on a current approximation of unknowns:

$$\begin{aligned}\tilde{\mathbf{x}}_{i+1} &= \mathcal{T}(\mathbf{x}_i, \mathbf{b}_i) \\ \tilde{\mathbf{P}}_{i+1} &= \mathcal{T}_{cov}(\mathbf{P}_i, \mathbf{b}_i) + \mathbf{Q}\end{aligned}\quad (3.2)$$

where \mathbf{P}_i is the covariance for the unknown state, $\tilde{\mathbf{x}}_{i+1}$ is the predicted state, and $\tilde{\mathbf{P}}_{i+1}$ is the predicted covariance. The main issue here is the transferring of the random variables together with their covariances through the nonlinear system.

To deal with it, several modifications of Kalman filtering were proposed. The simplest filters are the extended Kalman filter [245] and the unscented Kalman filter (UKF) [89]. The extended Kalman filter assumes linearization of the process transformation operator. The calculations are then performed as for the linear system. The main drawback here is that this filter requires computation of the Jacobian operator. This is a complex step, and for some systems, where the transition model is not defined clearly, it is very hard to get. In addition to that, the linear approximation might be quite inaccurate, which leads to instabilities in estimation.

The UKF discretizes the probabilistic space of the unknowns. The propagation via a nonlinear system is performed through the so-called *transformation of uncertainty*, presented in Fig. 3.1. The main idea is to parametrize the Gaussian distribution using a set of samples called *sigma points*, which hold the mean and covariance information, but are easy to transfer through a nonlinear function [91]. The next state of the process is predicted for each of these sigma points, and then the predicted distribution is reconstructed based on their combination. Therefore, the equations 3.2 are transformed to:

$$\begin{aligned}\mathbf{x}_i^{\sigma_k} &= \mathbf{x}_i + \sqrt{\mathbf{P}_i} I^{(k)} \\ \tilde{\mathbf{x}}_{i+1} &= \mathcal{E}\left(\mathcal{T}(\mathbf{x}_i^{\sigma_k}, \mathbf{b}_i)\right) \\ \tilde{\mathbf{P}}_{i+1} &= \left(\mathcal{T}(\mathbf{x}_i^{\sigma_k}, \mathbf{b}_i) - \tilde{\mathbf{x}}_{i+1}\right) \left(\mathcal{T}(\mathbf{x}_i^{\sigma_k}, \mathbf{b}_i) - \tilde{\mathbf{x}}_{i+1}\right)^T + \mathbf{Q}\end{aligned}\quad (3.3)$$

where $\mathbf{x}_i^{\sigma_k}$ is a sigma point with index k , $I^{(k)}$ is a matrix of unitary sigma points, and \mathcal{E} is the statistical mean.

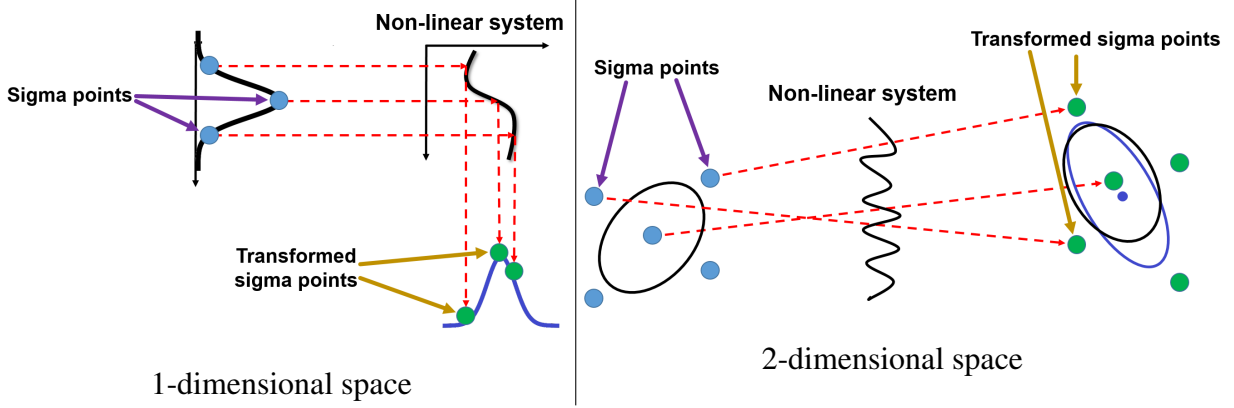


Figure 3.1: The transformation of uncertainty. To transform the Gaussian distribution through a nonlinear system a set of sigma points have to be selected in space. The new distribution is reconstructed based on the transformed points.

The unitary points describe the topological space used for distribution propagation. There are several main possibilities. The most accurate topology is *star sigma points*, which is defined such as:

$$I^{(k)} = \begin{cases} \sqrt{N_x} \mathbf{e}_k & 1 \leq k \leq N_x \\ -\sqrt{N_x} \mathbf{e}_{k-N_x} & N_x + 1 \leq k \leq 2N_x \\ 0 & k = 2N_x + 1 \end{cases} \quad (3.4)$$

where \mathbf{e}_k is a canonical vector in the state space and N_x is the unknown state size. Thus, star topology identifies $2N_x + 1$ sigma points. Without considering the last sigma point, this topology transforms to a set of *canonical sigma points*, which is defined as:

$$I^{(k)} = \begin{cases} \sqrt{N_x} \mathbf{e}_k & 1 \leq k \leq N_x \\ -\sqrt{N_x} \mathbf{e}_{k-N_x} & N_x + 1 \leq k \leq 2N_x \end{cases} \quad (3.5)$$

Also, the variant with the smallest amount of sigma points, necessary to transfer the probability distribution with a sufficient accuracy, is called *simplex sigma points*. In [90], the authors proposed to construct these points based on minimal skew (third order moment), which results in recursive construction procedure. One option is to use a set of points equidistant from the mean value [146]. So, now the matrix looks like:

$$\begin{cases} I^{(1)} = \begin{pmatrix} -\frac{\sqrt{N_x}}{\sqrt{2}} & \frac{\sqrt{N_x}}{\sqrt{2}} \\ \frac{\sqrt{N_x}}{\sqrt{2}} & \frac{\sqrt{N_x}}{\sqrt{2}} \end{pmatrix} \\ I^{(k)} = \begin{pmatrix} & & & 0 \\ & & & \vdots \\ & I^{(k-1)} & & 0 \\ \frac{\sqrt{N_x}}{\sqrt{k(k+1)}} & \frac{\sqrt{N_x}}{\sqrt{k(k+1)}} & \frac{\sqrt{N_x}}{\sqrt{k(k+1)}} & \frac{-k\sqrt{N_x}}{\sqrt{k(k+1)}} \end{pmatrix} \end{cases} \quad 2 \leq k \leq N_x \quad (3.6)$$

More information about unitary points generation is given in [90, 146].

Compared to an extended Kalman Filter, the unscented filter does not require to compute the Jacobian of the system. On the other hand, the nonlinear transformation has to be performed for every sigma point. It might not be an issue in case of simple systems; however, for a simulation with a Finite Element Method, it means that we have to compute a full simulation step for every sigma point.

Now, let us consider the correction step. Since model and observations in Kalman filtering context have uncertainties, we do not trust completely either the model, or the observations, but instead of that, we compute a conditional probability based on their expectations. The correction is computed by the minimization of an a posteriori state error covariance [10, 218]. Or, as it can be shown, it equivalently minimizes the sum of squared *Mahalanobis distances* (a distance metric to measure the distance between an element and a distribution). The distance is taken between current and already predicted model states and also between simulated and real observations. It is done with respect to the available covariances, as *cost function* \mathcal{J} shows [10, 218]:

$$\mathcal{J}(\hat{\mathbf{x}}) = \frac{1}{2}(\hat{\mathbf{x}} - \tilde{\mathbf{x}}_{i+1})^T \tilde{\mathbf{P}}_{i+1}^{-1}(\hat{\mathbf{x}} - \tilde{\mathbf{x}}_{i+1}) + \frac{1}{2}(\mathbf{z}_{i+1} - \mathcal{H}(\hat{\mathbf{x}}))^T \mathbf{V}^{-1}(\mathbf{z}_{i+1} - \mathcal{H}(\hat{\mathbf{x}})) \quad (3.7)$$

So, we look for the state $\bar{\bar{\mathbf{x}}}_{i+1}$, where the derivative of the cost function reaches zero. After performing mathematical operations, we get the next expressions for the updated state and then for its covariance [91]:

$$\begin{aligned} \tilde{\mathbf{z}}_{i+1} &= \mathcal{H}(\tilde{\mathbf{x}}_{i+1}) \\ \mathbf{P}_{\mathbf{zx}} &= cov(\tilde{\mathbf{x}}_{i+1}, \tilde{\mathbf{z}}_{i+1}) \\ \mathbf{P}_{\mathbf{z}} &= cov(\tilde{\mathbf{z}}_{i+1}, \tilde{\mathbf{z}}_{i+1}) + \mathbf{V} \\ \mathcal{K} &= \mathbf{P}_{\mathbf{zx}}(\mathbf{P}_{\mathbf{z}})^{-1} \\ \bar{\bar{\mathbf{x}}}_{i+1} &= \tilde{\mathbf{x}}_{i+1} + \mathcal{K}(\mathbf{z}_{i+1} - \tilde{\mathbf{z}}_{i+1}) \\ \bar{\bar{\mathbf{P}}}_{i+1} &= \tilde{\mathbf{P}}_{i+1} - \mathbf{P}_{\mathbf{zx}}(\mathbf{P}_{\mathbf{z}})^{-1}\mathbf{P}_{\mathbf{zx}}^T \end{aligned} \quad (3.8)$$

where $\tilde{\mathbf{z}}_{i+1}$ are the predicted measurements, *cov* is the cross covariance operator, $\mathbf{P}_{\mathbf{zx}}$ is the covariance for predicted estimations and observations, $\mathbf{P}_{\mathbf{z}}$ is the covariance for predicted observations, \mathcal{K} is the Kalman gain, $\bar{\bar{\mathbf{x}}}_{i+1}$ is the corrected state, and $\bar{\bar{\mathbf{P}}}_{i+1}$ is the corrected state covariance. The technical details are available in [89, 91].

Additionally, we can say that, at any iteration, the predicted unknowns contain some initial approximation and their correction for differences between observations, obtained from all previous iterations. Therefore, the essence of the Kalman filter is to find the solution that fits the best for all available measurements.

The general algorithm for the unscented Kalman filter is described in Alg. 1. It consists of an initialization part and a loop that contains prediction and correction steps. Inside both steps, there is an additional loop to predict state and compute observations separately for every sigma point. Commonly, the state contains object nodes and unknown stiffness parameters.

Algorithm 1: Unscented Kalman filter

```

1: Initialize data:
2: set  $\mathbf{x}_1$  - models positions and unknown parameters
3: set  $I^{(k)}$ ,  $\mathbf{P}_1$ ,  $\mathbf{Q}$ ,  $\mathbf{V}$  - initial filter parameters
4: get  $\mathbf{b}_1^{(c)}$  - control features at the initial step
5: set  $\mathcal{T} = \mathcal{T}(\mathbf{x}_1, \mathbf{b}_1^{(c)})$  - finite element model
6: for each simulation step  $i$  do
7:   Compute prediction phase:
8:   get  $\mathbf{b}_{i+1}^{(c)}$  - control features
9:    $\mathbf{x}_i^{\sigma_*} = \mathbf{x}_i + \sqrt{\mathbf{P}_i} I^{(k)}$  - generate sigma points
10:  for each sigma point  $k$  do
11:     $\tilde{\mathbf{x}}_{i+1}^{\sigma_k} = \mathcal{T}(\mathbf{x}_i^{\sigma_k}, \mathbf{b}_{i+1}^{(c)})$  - get result from deformation step
12:  end for
13:   $\tilde{\mathbf{x}}_{i+1} = \mathcal{E}(\tilde{\mathbf{x}}_{i+1}^{\sigma_*})$  - compute predicted state as mean of sigma points
14:   $\tilde{\mathbf{P}}_{i+1} = (\tilde{\mathbf{x}}_{i+1}^{\sigma_*} - \tilde{\mathbf{x}}_{i+1})(\tilde{\mathbf{x}}_{i+1}^{\sigma_*} - \tilde{\mathbf{x}}_{i+1})^T + \mathbf{Q}$  - compute predicted covariance
15:  Compute correction phase:
16:  get  $\mathbf{z}_{i+1}$  - observation features
17:  for each sigma point  $k$  do
18:     $\tilde{\mathbf{z}}_{i+1}^{\sigma_k} = \mathcal{H}(\tilde{\mathbf{x}}_{i+1}^{\sigma_k})$  - get predicted observation
19:  end for
20:   $\mathbf{P}_{\mathbf{z}\mathbf{z}} = (\tilde{\mathbf{x}}_{i+1}^{\sigma_*} - \tilde{\mathbf{x}}_{i+1})(\tilde{\mathbf{z}}_{i+1}^{\sigma_k} - \mathcal{E}(\tilde{\mathbf{z}}_{i+1}^{\sigma_*}))^T$  - compute cross covariance
21:   $\mathbf{P}_{\mathbf{z}} = (\tilde{\mathbf{z}}_{i+1}^{\sigma_k} - \mathcal{E}(\tilde{\mathbf{z}}_{i+1}^{\sigma_*}))(\tilde{\mathbf{z}}_{i+1}^{\sigma_k} - \mathcal{E}(\tilde{\mathbf{z}}_{i+1}^{\sigma_*}))^T + \mathbf{V}$  - compute observation covariance
22:   $\mathcal{K}_{i+1} = \mathbf{P}_{\mathbf{z}\mathbf{z}}\mathbf{P}_{\mathbf{z}}^{-1}$  - compute Kalman gain
23:   $\bar{\bar{\mathbf{x}}}_{i+1} = \tilde{\mathbf{x}}_{i+1} + \mathcal{K}_{i+1}(\mathbf{z}_{i+1} - \mathcal{E}(\tilde{\mathbf{z}}_{i+1}^{\sigma_*}))$  - compute corrected state
24:   $\bar{\bar{\mathbf{P}}}_{i+1} = \tilde{\mathbf{P}}_{i+1} - \mathbf{P}_{\mathbf{z}\mathbf{z}}\mathbf{P}_{\mathbf{z}}^{-1}\mathbf{P}_{\mathbf{z}\mathbf{z}}^T$  - compute corrected covariance
25: end for

```

The prediction step can be very costly in case when using a model with many degrees of freedom. Even for the simplex topological set, to transform the sigma points would require $M + 1$ simulations if M is the number of elements in the stochastic state vector (line 11 of the algorithm). With a deformable mesh of N nodes and K stiffness parameters, this would mean $3N + K + 1$ simulations. So, a simple FE mesh of only a few hundred nodes would be too time-consuming for a clinical application, as it would require more than several hundred simulations for each step of the assimilation process.

To solve this issue, we rely on filter improvements that are described in the next sections.

3.2.2 Reduced-order unscented Kalman filter

The unscented transformation is based on the fact that it is easier to sample a Gaussian distribution with known properties than to process an arbitrary nonlinear system. The issue here is that every sample point has to be transferred through a nonlinear model. Therefore, when we have a lot of unknowns that we are trying to estimate, many sigma points are generated, and the nonlinear transformation for them becomes computationally costly. In current context, the state

elements of the Kalman filter does not necessarily contain only stiffness values, but also hold positions (locations) of elements for an object in space, which dramatically increase the state size.

The idea to reduce the number of sigma points for better performance was already discussed in a certain number of papers. In [90], the authors propose to construct a set of sigma points the way to keep only the first statistical moments for the propagated data. The other ideas are dedicated to reduction of the state covariance matrix, so the modifications of the Kalman filter, namely reduced-rank square root Kalman filter [236] and, its variance, singular evolute interpolated Kalman filter [182] appeared. Later, a similar modification of unscented Kalman filter was proposed [146]. There, only essential part of unknowns is discretized by sigma points and goes through modeling process. All other parameters can be expressed the unique way through them, and are considered to be known based on estimations of the essential part. In [145], the authors apply this approach to estimate BCs, presented as penalty impacts, for an aorta vessel model. In the paper [180], the authors use the same approach to identify boundary conditions for a porcine liver. A very simple model was used, but the main point was to show that the boundary conditions estimated in such a way will improve the simulation accuracy.

To reduce the dimension of unknown space, the idea is to process the covariance matrices in the factorized form [146]. It is presented in the Fig. 3.2:

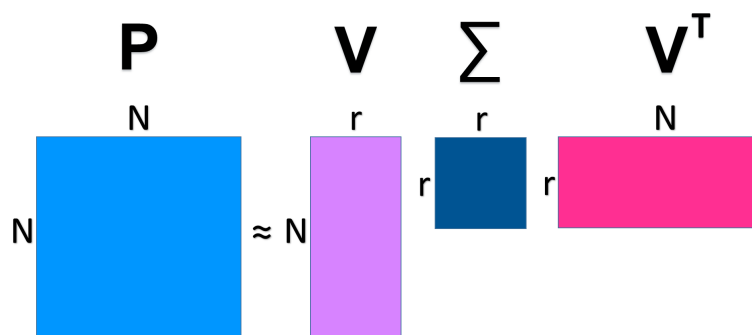


Figure 3.2: State covariance transformation to a reduce order. Since state covariance matrix is symmetric, it looks like matrix spectral, eigen, or VDV decomposition.

In filtering context, the decomposition for the covariance matrix can be written as:

$$\tilde{\mathbf{P}} = \mathbf{L}\mathbf{U}^{-1}\mathbf{L}^T \quad (3.9)$$

where \mathbf{U} is the reduced-order covariance, which is an invertible matrix, and \mathbf{L} is the transformation between reduced-order and normal states.

The final dimension depends on the selected components, which represent the system state. So, if the value r is smaller than the covariance size N , then after performing its spectral decomposition, the size of the factorized matrix will be less than the size of the source one. And thus, in filtering equations we could construct and transform sigma points, related only to this

reduced matrix. Taking this into account, the system of equations 3.3 can be now rewritten as:

$$\begin{aligned}
\mathbf{x}_i^{\sigma_k} &= \mathbf{x}_i + \mathbf{L}_i \sqrt{\mathbf{U}_i^{-1}} I^{(k)} \\
\tilde{\mathbf{x}}_{i+1} &= \mathcal{E} \left(\mathcal{T}(\mathbf{x}_i^{\sigma_k}, \mathbf{b}_i) \right) \\
\mathbf{L}_{i+1} &= \tilde{\mathbf{X}}_{i+1}^{\sigma_*} I^{(k)T} \\
\tilde{\mathbf{P}}_{i+1} &= \mathbf{L}_{i+1} \left(I^{(k)} I^{(k)T} \right)^{-1} \mathbf{L}_{i+1}^T + \mathbf{Q}
\end{aligned} \tag{3.10}$$

where $\tilde{\mathbf{X}}^{\sigma_*}$ is a matrix composed from sigma points. The index k is now related to the size of the reduced-order data \mathbf{U}_i . We should note that in general a deviation of state and its mean is used to update the reduced-order transformation matrix. But in case when sigma points are centered (with zero empirical mean), it is possible to update it using only transferred sigma points [146].

During filter correction step, initially, the updated reduced-order covariance is computed:

$$\begin{aligned}
\tilde{\mathbf{z}}_{i+1} &= \mathcal{H}(\tilde{\mathbf{x}}_{i+1}) \\
\mathbf{H}_{i+1} &= \tilde{\mathbf{z}}_{i+1} I^{(k)T} \\
\mathbf{U}_{i+1} &= \mathbf{1} + \mathbf{H}_{i+1}^T \mathbf{V}^{-1} \mathbf{H}_{i+1}
\end{aligned} \tag{3.11}$$

where \mathbf{H} is an additional matrix to process observations in reduced-order space. Then, the Kalman gain and corrected covariance are computed through the reduce-order data:

$$\begin{aligned}
\mathcal{K}_{i+1} &= \mathbf{L}_{i+1} \mathbf{U}_{i+1}^{-1} \mathbf{H}_{i+1}^T \mathbf{V}^{-1} \\
\bar{\bar{\mathbf{P}}}_{i+1} &= \mathbf{L}_{i+1} \mathbf{U}_{i+1}^{-1} \mathbf{L}_{i+1}^T
\end{aligned} \tag{3.12}$$

The final algorithm for the reduced-order unscented Kalman filter (ROUKF) is presented in Alg. 2. We have to note the given transformations are valid only for simplex sigma points; thus, for other topologies some modifications have to be performed.

The reduced-order method might significantly decrease the computation time since fewer simulations are required. In current work, for the best case, the reduced state is bounded up to the unknown parameters. However, we still need to keep a good approximation of the original space. Otherwise, the estimation process might be inaccurate.

Algorithm 2: Reduced-order unscented Kalman filter

```

1: Initialize data:
2: set  $\mathbf{x}_1$  - model positions and unknown parameters
3: set  $I^{(k)}$  - unitary sigma points
4: set  $\mathbf{P}_1, \mathbf{Q}, \mathbf{V}$  - initial filter parameters
5:  $\mathbf{P}_1 = \mathbf{L}_1 \mathbf{U}_1^{-1} \mathbf{L}_1^T$  - perform  $LUL^T$  decomposition of initial covariance
6: get  $\mathbf{b}_1^{(c)}$  - control features at the initial step
7: set  $\mathcal{T} = \mathcal{T}(\mathbf{x}_1, \mathbf{b}_1^{(c)})$  - finite element model
8: for each simulation step  $i$  do
9:   Compute prediction phase:
10:  get  $\mathbf{b}_{i+1}^{(c)}$  - control features
11:   $\mathbf{x}_i^{\sigma*} = \mathbf{x}_i + \mathbf{L}_i \sqrt{\mathbf{U}_i^{-1} I^{(k)}}$  - generate sigma points
12:  for each sigma point  $k$  do
13:     $\tilde{\mathbf{x}}_{i+1}^{\sigma k} = \mathcal{T}(\mathbf{x}_i^{\sigma k}, \mathbf{b}_{i+1}^{(c)})$  - get result from deformation step
14:  end for
15:   $\tilde{\mathbf{x}}_{i+1} = \mathcal{E}(\tilde{\mathbf{x}}_{i+1}^{\sigma*})$  - compute predicted state as mean of sigma points
16:   $\tilde{\mathbf{L}}_{i+1} = \tilde{\mathbf{x}}_{i+1}^{\sigma*} I^{(k)T}$  - updated transformation matrix to reduced space
17:   $\tilde{\mathbf{P}}_{i+1} = \tilde{\mathbf{L}}_{i+1} (I^{(k)} I^{(k)T})^{-1} \tilde{\mathbf{L}}_{i+1}^T + \mathbf{Q}$  - compute predicted covariance
18:  Compute correction phase:
19:  get  $\mathbf{z}_{i+1}$  - observation features
20:  for each sigma point  $k$  do
21:     $\tilde{\mathbf{z}}_{i+1}^{\sigma k} = \mathcal{H}(\tilde{\mathbf{x}}_{i+1}^{\sigma k})$  - get predicted observation
22:  end for
23:   $\tilde{\mathbf{H}}_{i+1} = \tilde{\mathbf{z}}_{i+1}^{\sigma*} I^{(k)T}$  - update transformation matrix to reduced observation space
24:   $\mathbf{U}_{i+1} = \mathbf{I} + \tilde{\mathbf{H}}_{i+1}^T \mathbf{V}^{-1} \tilde{\mathbf{H}}_{i+1}$  - compute reduced covariance
25:   $\mathcal{K}_{i+1} = \tilde{\mathbf{L}}_{i+1} \mathbf{U}_{i+1}^{-1} \tilde{\mathbf{H}}_{i+1}^T \mathbf{V}^{-1}$  - compute Kalman gain
26:   $\bar{\mathbf{x}}_{i+1} = \tilde{\mathbf{x}}_{i+1} + \mathcal{K}_{i+1} (\mathbf{z}_{i+1} - \mathcal{E}(\tilde{\mathbf{z}}_{i+1}^{\sigma*}))$  - compute corrected state
27:   $\bar{\mathbf{P}}_{i+1} = \tilde{\mathbf{L}}_{i+1} \mathbf{U}_{i+1}^{-1} \tilde{\mathbf{L}}_{i+1}^T$  - compute corrected covariance
28: end for

```

It is important to say that, initially, to estimate boundary conditions for a liver, the reduced version was proposed in the paper [180], where the authors estimate the properties of the attachments for the pig liver lobe.

3.2.3 Ensemble transform Kalman filter

Another possible option for parameter estimation is ensemble Kalman filter. Despite the fact that initially it was proposed as a derivation from the Monte Carlo approach, with some assumptions, it could be considered as a nonlinear Kalman filter with a reduced-order state. There exists several different variations for this type of filter. In this work, we decided to focus on the ensemble transform Kalman filter (ETKF).

The Ensemble Kalman filter was initially proposed in [52]. Its main applications are weather

forecasting, oceanography, and climate description systems, but it can also be used in various areas of science.

The main idea of the filter is to present the probability distribution as a set of state particles, which are called *ensemble members* (Fig. 3.3). Instead of propagating the state covariance through the nonlinear system using linearization or a set of sigma points, it is approximated by the Monte Carlo estimator. Here, we keep more information about the uncertainty of the system state using the fewer number of the sample elements, which increases the computational speed [10].

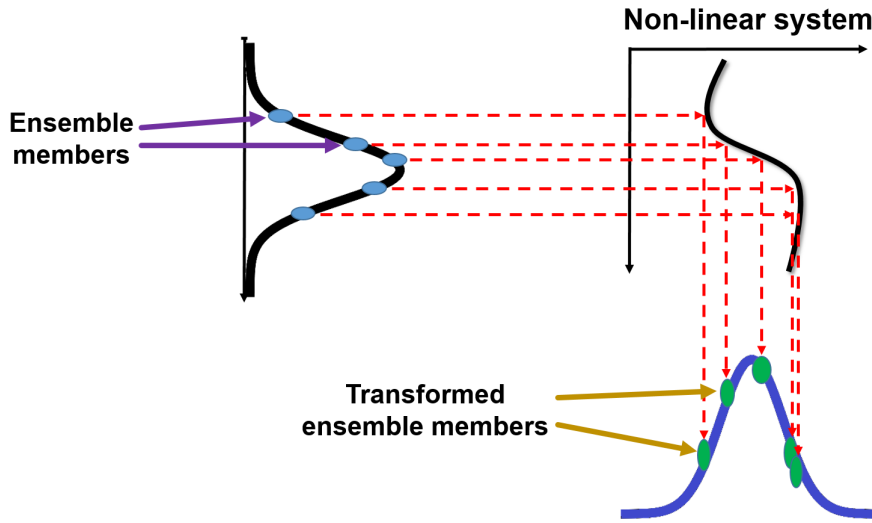


Figure 3.3: Transformation of ensemble members through a nonlinear system. The probability distribution is reconstructed using Monte Carlo estimator applied to the transformed members.

The other important difference is that, unlike uncertainty transformation processing, the ensemble particles are generated only once during initialization. So, during the prediction step, we just transfer the ensemble members through a nonlinear system and compute their mean and covariance [10, 84]:

$$\tilde{\mathbf{x}}_{i+1}^{enk} = \mathcal{T}(\mathbf{x}_i^{enk}, \mathbf{b}_i) \quad (3.13)$$

$$\tilde{\mathbf{x}}_{i+1} = \frac{1}{N_{en}} \sum_{k=1}^{N_{en}} \tilde{\mathbf{x}}_{i+1}^{enk}$$

$$\tilde{\mathbf{P}}_{i+1} = \frac{1}{N_{en} - 1} \sum_{k=1}^{N_{en}} \left(\tilde{\mathbf{X}}_{i+1}^{enk} - \tilde{\mathbf{x}}_{i+1} \right) \left(\tilde{\mathbf{X}}_{i+1}^{enk} - \tilde{\mathbf{x}}_{i+1} \right)^T$$

where \mathbf{x}_i^{enk} is an ensemble member, $\tilde{\mathbf{x}}_{i+1}^{enk}$ is a predicted member, $\tilde{\mathbf{X}}_{i+1}^{enk}$ is a matrix composed of predicted members, and N_{en} is their total number.

For the efficient computation of the analysis (correction) step, it is also performed for the ensemble members [83]. Therefore the state size has to be reduced up to the ensemble subspace:

$$\mathbf{x}_{i+1} = \tilde{\mathbf{x}}_{i+1} + \left(\tilde{\mathbf{X}}_{i+1}^{en} - \tilde{\mathbf{x}}_{i+1} \right) \boldsymbol{\omega} = \tilde{\mathbf{x}}_{i+1} + \tilde{\mathbf{X}}_{i+1}^{diff} \boldsymbol{\omega}_{i+1} \quad (3.14)$$

here $\tilde{\mathbf{X}}_{i+1}^{diff}$ shows the difference between ensemble members and their mean, ω_{i+1} describes state coefficients in ensemble subspace [10]. Inserting the obtained equation in the cost function, the equation 3.7 is now reformulated as [83]:

$$\mathcal{J}(\omega) = \frac{1}{2}(N_{en} - 1)\omega\omega^T + \frac{1}{2}\left(\mathbf{z}_{i+1} - \mathcal{H}(\tilde{\mathbf{x}}_{i+1} + \tilde{\mathbf{X}}_{i+1}^{diff}\omega)\right)^T \mathbf{V}^{-1}\left(\mathbf{z}_{i+1} - \mathcal{H}(\tilde{\mathbf{x}}_{i+1} + \tilde{\mathbf{X}}_{i+1}^{diff}\omega)\right) \quad (3.15)$$

To solve the equation, we assume that the observation operator is quasi-linear:

$$\mathcal{H}(\tilde{\mathbf{x}}_{i+1} + \tilde{\mathbf{X}}_{i+1}^{diff}\omega) = \mathcal{H}(\tilde{\mathbf{x}}_{i+1}) + \mathcal{H}(\tilde{\mathbf{X}}_{i+1}^{diff})\omega = \widetilde{\mathbf{z}}_{i+1} + \tilde{\mathbf{Z}}_{i+1}^{diff}\omega \quad (3.16)$$

where $\tilde{\mathbf{Z}}_{i+1}^{diff}$ is the observation difference related to the correspondent state difference. By minimizing the cost function, we get [83]:

$$\omega_{i+1} = \left[(N_{en} - 1)\mathbf{1} + \left(\tilde{\mathbf{Z}}_{i+1}^{diff} \right)^T \mathbf{V}^{-1} \tilde{\mathbf{Z}}_{i+1}^{diff} \right]^{-1} \left(\tilde{\mathbf{Z}}_{i+1}^{diff} \right)^T \mathbf{V}^{-1} (\mathbf{z}_{i+1} - \mathcal{H}(\tilde{\mathbf{x}}_{i+1})) \quad (3.17)$$

Finally, we generate the corrected ensemble members $\overline{\mathbf{x}}_{i+1}^{enk}$:

$$\overline{\mathbf{x}}_{i+1}^{enk} = \tilde{\mathbf{x}}_{i+1} + \tilde{\mathbf{X}}_{i+1}^{diff} \left(\omega_{i+1} + \sqrt{(N_{en} - 1) \left[(N_{en} - 1)\mathbf{1} + \left(\tilde{\mathbf{Z}}_{i+1}^{diff} \right)^T \mathbf{V}^{-1} \tilde{\mathbf{Z}}_{i+1}^{diff} \right]^{-1}} \right) \quad (3.18)$$

The transformation and correction of ensemble members are handled in a loop, like in a case of a general Kalman filter.

Since the covariance matrix is always approximated by a small number of ensemble members, an estimation error appears. Without additional steps, this error is accumulating, which results in filter divergence. Fortunately, there are two essential moments that have to be taken into account to avoid it. They are *localization* and *inflation* [10, 84].

Localization implies that to correct a state we need to use only local observations. However, the important moment is that, unlike huge systems, we estimate boundary conditions for the same organ we observe. So when the observations are moving under any external impact, the organ is deformed, and some boundary conditions are definitely active. Consequently we think that all observations are strongly correlated with estimated parameters and consider the case when all of them are local.

Inflation is an additional parameter that is added to ensemble members to prevent the filter divergence. It can also be considered as a way to take into account the process uncertainty, added after the prediction step. So, to make it similar to unscented Kalman filter, we decided to put it in an additive form after ensemble members prediction as an additional parameter ι_i , where $\mathcal{E}(\iota_i \iota_i^T) = \mathbf{Q}$.

The final algorithm is presented in Alg. 3. It is important to note that the number of ensemble members might be less than unknown state size, which might reach several thousand elements. However, it is not obvious whether we transfer the sufficient amount of information to reconstruct the probability distribution accurately.

Algorithm 3: Ensemble transform Kalman filter

```

1: Initialize data:
2: set  $\mathbf{x}_1$  - model positions and unknown parameters
3: set  $N_{en}$  - ensemble members amount
4: set  $\mathbf{P}_1, \mathbf{Q}, \mathbf{v}_1, \mathbf{V}$  - initial filter parameters
5: for each ensemble member  $k$  do
6:    $\mathbf{x}_1^{en_k} = \mathbf{x}_1 + \mathbf{P}_1 * \text{Gaussian\_rand}(0, 1)$  - generate ensemble members
7: end for
8: get  $\mathbf{b}_1^{(c)}$  - control features at the initial step
9: set  $\mathcal{T} = \mathcal{T}(\mathbf{x}_1, \mathbf{b}_1^{(c)})$  - finite element model
10: for each simulation step  $i$  do
11:   Compute prediction phase:
12:   get  $\mathbf{b}_{i+1}^{(c)}$  - control features
13:   for each ensemble member  $k$  do
14:      $\tilde{\mathbf{x}}_{i+1}^{en_k} = \mathcal{T}(\mathbf{x}_i^{en_k}, \mathbf{b}_{i+1}^{(c)})$  - get result from deformation step
15:   end for
16:    $\tilde{\mathbf{x}}_{i+1} = \frac{1}{N_{en}} \sum_{k=1}^{N_{en}} \tilde{\mathbf{x}}_{i+1}^{en_k}$  - compute prediction mean
17:   for each ensemble member  $k$  do
18:      $\tilde{\mathbf{x}}_{i+1}^{diff_k} = \tilde{\mathbf{x}}_{i+1}^{en_k} - \tilde{\mathbf{x}}_{i+1} + \mathbf{v}_{i+1}$  - compute perturbation with inflation
19:   end for
20:    $\tilde{\mathbf{P}}_{i+1} = \frac{1}{N_{en}-1} \tilde{\mathbf{X}}_{i+1}^{diff*} \left( \tilde{\mathbf{X}}_{i+1}^{diff*} \right)^T$  - compute predicted covariance of ensemble members
21:   Compute correction phase:
22:   get  $\mathbf{z}_{i+1}$  - observation features
23:   for each ensemble member  $k$  do
24:      $\tilde{\mathbf{z}}_{i+1}^{en_k} = \mathcal{H}(\tilde{\mathbf{x}}_{i+1}^{en_k})$  - get predicted observation
25:   end for
26:    $\tilde{\mathbf{z}}_{i+1} = \frac{1}{N_{en}} \sum_{k=1}^{N_{en}} \tilde{\mathbf{z}}_{i+1}^{en_k}$  - compute observations mean
27:   for each ensemble member  $k$  do
28:      $\tilde{\mathbf{z}}_{i+1}^{diff_k} = \tilde{\mathbf{z}}_{i+1}^{en_k} - \tilde{\mathbf{z}}_{i+1}$  - get observation perturbation
29:   end for
30:    $\mathcal{A}_{i+1} = \left[ (N_{en} - 1) \mathbf{1} + \left( \tilde{\mathbf{Z}}_{i+1}^{diff*} \right)^T \mathbf{V}^{-1} \tilde{\mathbf{Z}}_{i+1}^{diff*} \right]^{-1}$ 
31:    $\boldsymbol{\omega}_{i+1} = \mathcal{A}_{i+1} \left( \tilde{\mathbf{Z}}_{i+1}^{diff*} \right)^T \mathbf{V}^{-1} (\mathbf{z}_{i+1} - \tilde{\mathbf{z}}_{i+1})$  - compute state coefficients in ensemble space
32:    $\bar{\mathbf{x}}_{i+1} = \tilde{\mathbf{x}}_{i+1} + \tilde{\mathbf{X}}_{i+1}^{diff*} \boldsymbol{\omega}_{i+1}$  - compute corrected (analysis) state mean
33:    $\mathcal{Y}_{i+1} = \sqrt{(N_{en} - 1)} \mathcal{A}_{i+1}$ 
34:    $\bar{\mathbf{P}}_{i+1} = \frac{1}{(N_{en}-1)} \tilde{\mathbf{X}}_{i+1}^{diff*} \mathcal{Y}_{i+1} \mathcal{Y}_{i+1}^T \left( \tilde{\mathbf{X}}_{i+1}^{diff*} \right)^T$  - compute corrected (analysis) covariance
35:   for each ensemble member  $k$  do
36:      $\bar{\mathbf{x}}_{i+1}^{en_k} = \bar{\mathbf{x}}_{i+1} + \tilde{\mathbf{x}}_{i+1}^{diff_k} \mathcal{Y}_{i+1}$  - compute correction (analysis)
37:   end for
38: end for

```

All in all, we decided to perform experiments and compare the accuracy as well as performance for different variants of filters.

3.3 Software module for data-driven simulation

To perform stochastic simulations, we need a software that implements filtering approaches and enables estimating various characteristics of a biomechanical model. Additionally, the context of our work requires real-time execution. The framework SOFA [5], which is used to create various models for medicine, allows performing complex simulations in real-time context, including several types of physically based models. But unfortunately, there is no additional functionality that provides a data-driven simulation processing.

Therefore, to satisfy the considered requirements, a plugin called Optimus was implemented, which is currently freely available online [181]. The main purpose of the plugin is to provide a testing environment for data-driven physically based modeling. Since we intended to verify different types of estimators, the created architecture allows adding approaches with generic prediction-correction structure. However, currently only several nonlinear Kalman filtering approaches are implemented. The framework contains approximately 21k lines of C++ code and 1.5k lines related to additional Python scripts. Apart from other functionality, there are several examples to show how data-driven configuration scripts have to be created and benchmark tests to verify whether any proposed modification of code works according to the general processing workflow.

The main idea behind the plugin is to separate a state of the system or stochastic state, using in the Kalman filtering approaches, from a mechanical state, necessary for the physically based simulation. This way, we minimize the existing dependencies between the data assimilation code and SOFA software components, which describe the mechanical aspects of the problem. Therefore, on the one hand, we can easily add new methods based on the predictor-corrector scheme such as, for example, a particle filter or a sequential least square. And on the other hand, we can use various possibilities for physically based modeling as well as other functionality available in SOFA, including those from other plugins and extensions. It is worth noting that technically with Optimus it is possible to estimate or identify any quantity defined in the configuration data.

The overview of the plugin components is presented in Fig. 3.4. Roughly talking, they can be organized into four groups, depending on their main purpose.

The classes from the filtering group (yellow square in Fig. 3.4) implement nonlinear filtering approaches. The image depicts only main approaches, while some slightly modified filters, created mainly for testing purposes, are replaced by "three points" symbol. All filtering components are organized in one big hierarchy headed by abstract class called *StochasticFilterBase*, which defines the main filtering interface including prediction and correction declarations.

The components from observation handlers group (green square) define different methods to process noisy, mapped, or partially missing observations. Again, many options were implemented, so three most using approaches are shown. As in case for the filtering components, the observation handlers are organized in a big hierarchy with abstract class *ObservationManager*.

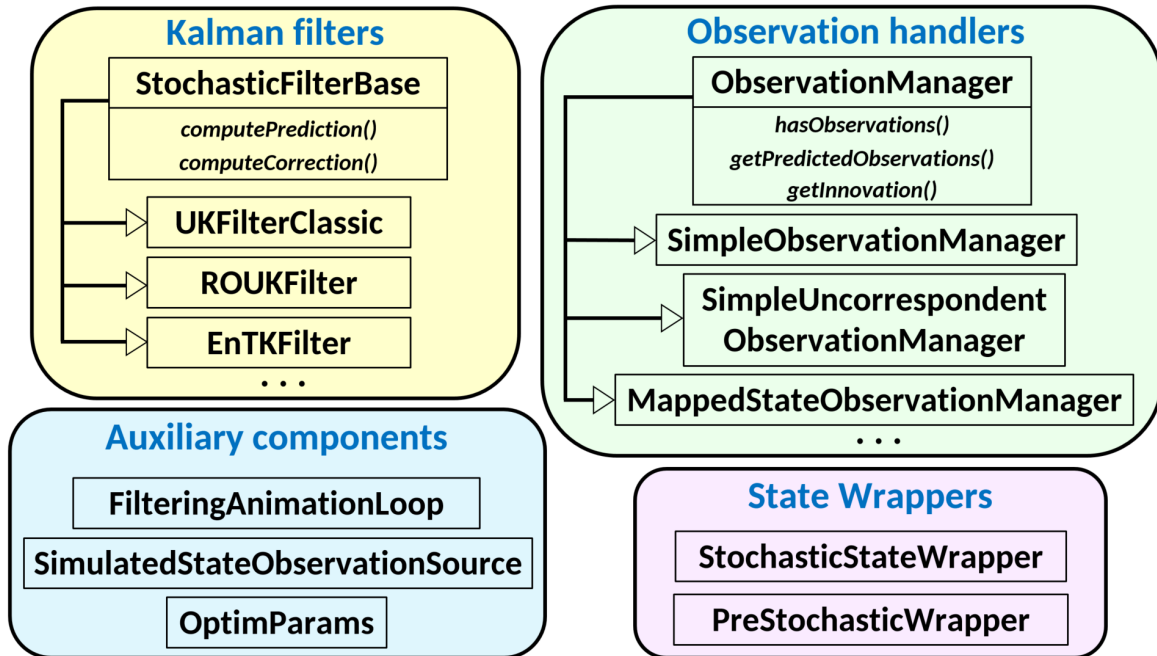


Figure 3.4: Overview of the components implemented in the Optimus plugin. They can be classified into four groups. Filtering group holds implementation of different Kalman filtering approaches. Observation manager group contains solutions to process the available observations. State wrappers group contains classes to transfer between filtering and mechanical states. Auxiliary components handle some additional issues to launch the data-driven process.

There are three main methods declared there. The first is verification whether there are available observations in the current step. If there are no observations, the correction step will be omitted. The second is getting predicted observations, which are required for some filtering approaches. The final method is computing innovation for the correction procedure. It is up to a specific implementation in the observation manager to decide whether the given observations are valuable enough to perform the correction step or they are just useless. But when the manager declares that it possesses observations, it has to compute the predicted observations and the innovation.

The third group is state wrapper components (violet square). These components transfer the stochastic state to the mechanical one together with the parameters and vice versa. There is no specific hierarchy here since finally the data transfer goes in a unique way.

Finally, the fourth group is presented by several components that add some additional functionality to execute the data assimilation process together with a possibility to take various model parameters for estimation.

Now, we will describe the workflow of the data assimilation process. The common SOFA configuration looks like a set of different components, each of which are responsible for a certain functionality. They are organized in a hierarchical structure that is usually called a *scene*. Primarily, an initialization is performed to extract all necessary data from the items and to create a physically based model. The simulation represents a loop *AnimationLoop*, in which a

modeling step is performed for a constructed system until the required outcome or any threshold condition is achieved.

In case of a data assimilation scene, the Optimus components described above are added to the *scene* in a specific order. The *AnimationLoop* is replaced with a data assimilation loop, implemented in the *FilteringAnimationLoop* class. Also, the filtering instance is put directly under the loop, and state wrappers are placed above the SOFA components. The place in hierarchy for the other Optimus components depends on the context of the problem.

The initialization part works like in a general case. The only essential difference is that plugin components are processed at the last moment since, to initialize a filter, they extract data from the created system.

The simulation now works with the *FilteringAnimationLoop* class where a prediction and correction steps are invoked (see Fig. 3.5). Thus, a single prediction and correction invocations correspond to one general simulation step in SOFA.

```

/// function called in every step of SOFA loop: calls initialization, prediction and
/// correction which must be implemented by the filter component
void FilteringAnimationLoop::step(const core::ExecParams* _params, SReal /**_dt*/) {

    actualStep++;

    // compute deterministic parts of simulation scene
    for (size_t i = 0; i < preStochasticWrappers.size(); i++)
        preStochasticWrappers[i]->step(_params, actualStep);

    // initialize internal data before main loop
    filter->initializeStep(_params, actualStep);

    // compute prediction
    filter->computePrediction();

    // compute correction
    filter->computeCorrection();

}

```

Figure 3.5: C++ example to illustrate the main idea of the *FilteringAnimationLoop* component.

A few words have to be said about the *PreStochasticWrapper*, noted in the code listing. During the simulation, the object under interest might interact with other objects, which we need to model as well. If we are not interested in estimating their state and, thus, want to keep outside the filtering process, we have to declare them as deterministic parts of the configuration by wrapping with the *PreStochasticWrapper*. The items wrapped by it behave like in the case of a general SOFA simulation.

All necessary calculations for the data-driven estimation are performed inside prediction and correction procedures.

3.3.1 Prediction procedure

In the prediction procedure, the filter computes prediction like in the Alg. 1, 2, and 3, given in Sections 3.2.1, 3.2.2, and 3.2.3. In fact, the mechanical components are hidden inside the transformation operator \mathcal{T} . This abstract layer is implemented inside the component called *StochasticStateWrapper*, which serves as a wrapper above all physical components under interest and

provides the correspondence between filtering and mechanical states. The filter computes prediction for sigma points by invoking the `computePerturbedStates` method of the wrapper. Inside it, the transformation is executed for every sigma point or ensemble member stored in a matrix, as Fig. 3.6 shows.

```
template <class FilterType> void UKFilterClassic<FilterType>::computePerturbedStates()
{
    EVectorX xCol(stateSize);
    int currentPointID = 0;

    for (size_t i = 0; i < sigmaPointsNum; i++) {
        // take a sigma point
        xCol = matXi.col(i);

        // transfer sigma point through stochastic state wrapper
        stateWrappers[0]->transformState(xCol, mechParams, &currentPointID);
        m_sigmaPointObservationIndexes[i] = currentPointID;

        // save the transfered point
        matXi.col(i) = xCol;
    }
}
```

Figure 3.6: C++ example to illustrate the transformation of sigma points inside the state wrapper.

The function `transformState` converts sigma point data to the mechanical state (Fig. 3.7), execute the animation step (like `step` function in the `AnimationLoop`), and perform back conversion. So, for the components placed below the `StochasticStateWrapper` in the hierarchy, this is just a single step of the simulation.

Here, the `OptimParams` component has to be noted, which participates in the conversion procedure (Fig. 3.7). In general, only object positions and velocities (if presented) are directly put to the correspondent SOFA components. But, depending on the problem, estimated parameters are usually presented in different components. The `OptimParams` class helps to create a link between filtering state and the parameters to update them directly before the animation step. In addition, it specifies the initial value and standard deviation for the estimated parameters. When needed, there is a possibility to create a set of `OptimParams` items to estimate several different parameters in the system.

Therefore, during the prediction step, for every sigma point a separate simulation is computed. Because of this, the prediction step is computationally expensive. And the SOFA framework, the purpose of which is to perform real-time simulations, is one of the best choices for such data estimation procedures.

After prediction, the filtering state is corrected inside the correction procedure.

3.3.2 Correction (Analysis) procedure

The correction procedure is almost exclusively performed in the filtering class, as given in the Alg. 1, 2, 3. The only component that, apart from the filter, participates in this process is `ObservationManager`. Its main responsibilities are to answer whether there are available observations, to provide the filter with predicted observations, and to compute innovations.

```

// function that sets SOFA state (position, velocity, parameters) from this->state (stochastic state, Eigen vector)
template <class DataTypes, class FilterType>
void StochasticStateWrapper<DataTypes, FilterType>::copyStateFilter2Sofa(const core::MechanicalParams* _mechParams,
                                                                    bool _setVelocityFromPosition) {
    typename MechanicalState::WriteVecCoord pos = mechanicalState->writePositions();
    typename MechanicalState::WriteVecDeriv vel = mechanicalState->writeVelocities();

    // transfer object positions if any
    if(estimatePosition.getValue()){
        for (helper::vector<std::pair<size_t, size_t> >::iterator it = positionPairs.begin();
            it != positionPairs.end(); it++) {
            for (size_t d = 0; d < posDim; d++) {
                pos[it->first][d] = this->state(it->second + d);
            }
        }
    }

    // transfer object velocities if any
    if(estimateVelocity.getValue()){
        for (helper::vector<std::pair<size_t, size_t> >::iterator it = velocityPairs.begin();
            it != velocityPairs.end(); it++) {
            for (size_t d = 0; d < velDim; d++) {
                vel[it->first][d] = this->state(it->second + d);
            }
        }
    }
    // if velocity is not estimated, it must be computed from the positions
    // at the beginning and at the end of the time step
    if (_setVelocityFromPosition) {
        if (velocityPairs.empty() && beginTimeStepPos.size() == pos.size()) {
            for (helper::vector<std::pair<size_t, size_t> >::iterator it = positionPairs.begin();
                it != positionPairs.end(); it++) {
                for (size_t d = 0; d < velDim; d++) {
                    vel[it->first][d] = (pos[it->first][d] - beginTimeStepPos[it->first][d])/this->getContext()->getDt();
                }
            }
        }
    }
}

sofa::simulation::MechanicalPropagateOnlyPositionAndVelocityVisitor(_mechParams).execute( this->gnode );
// let the OptimParams to extract the actual values of parameters from the filtering state
for (size_t opi = 0; opi < vecOptimParams.size(); opi++)
    vecOptimParams[opi]->vectorToParams(this->state);
}

```

Figure 3.7: C++ example to illustrate the data transferring from a sigma point to a mechanical state.

For now, the implementation works only with the cases where observations are attached to the object of interest. Therefore simulated observations are held in a separate component, which is mapped to the object position. To compute predicted observations for every sigma point or ensemble member, the *ObservationManager* extracts data from the *StochasticStateWrapper* and applies mapping (Fig. 3.8).

The real observations are extracted from a file where they were saved during data processing stage that is performed separately. To load them, a *SimulatedStateObservationsSource* component is used. So the *ObservationManager* just sends a request to the *SimulatedStateObservationsSource* about the data availability in the file (Fig. 3.9). The innovation is computed as a difference between predicted and real observations. After that, the observation manager returns all necessary data to the filter, and it continues correcting the stochastic state. Finally, after the correction procedure and before the next circle, the filter requests the *StochasticStateWrapper* to update the mechanical components with the corrected values.

```

template <class FilterType, class DataTypes1, class DataTypes2>
bool SimpleObservationManager<FilterType,DataTypes1,DataTypes2>::
    getPredictedObservations(int _id, EVectorX& _predictedObservation)
{
    Data<typename DataTypes1::VecCoord> predictedMasterState;
    Data<typename DataTypes2::VecCoord> predictedMappedState;

    typename DataTypes1::VecCoord& predictedMasterStateEdit = *predictedMasterState.beginEdit();
    typename DataTypes2::VecCoord& predictedMappedStateEdit = *predictedMappedState.beginEdit();

    predictedMasterStateEdit.resize(masterState->getSize());
    predictedMappedStateEdit.resize(mappedState->getSize());

    // get object positions from state wrapper
    stateWrapper->getActualPosition(_id, predictedMasterStateEdit);

    // apply mapping to get observations
    sofa::core::MechanicalParams mp;
    mapping->apply(&mp, predictedMappedState, predictedMasterState);

    // save observations in the output vector
    _predictedObservation.resize(this->observationSize);
    for (size_t i = 0; i < predictedMappedStateEdit.size(); i++)
        for (size_t d = 0; d < DataTypes1::spatial_dimensions; d++)
            _predictedObservation(DataTypes1::spatial_dimensions * i + d) = predictedMappedStateEdit[i][d];

    return true;
}

```

Figure 3.8: C++ example to illustrate the computation of predicted observations.

```

template <class FilterType, class DataTypes1, class DataTypes2>
bool SimpleObservationManager<FilterType,DataTypes1,DataTypes2>::hasObservations(double _time) {
    bool hasObservation;

    // get observations from file
    if(this->actualTime == 0){
        hasObservation = true;
    } else {
        hasObservation = observationSource->getObservation(this->actualTime, realObservations);
    }

    if (!hasObservation) {
        PRNE("No observation for time " << _time);
    }

    return hasObservation;
}

```

Figure 3.9: C++ example to illustrate the verification of observations availability.

A more detailed documentation together with several examples is available at [181] in the *doc* folder.

3.4 Estimation using Kalman filtering

In the final part of this chapter, we describe several experiments to demonstrate the potential of the Kalman filtering approach. It is important to note that despite the main goal of this work is

to estimate accurately the boundary conditions, the filtering methods can also be used to estimate other model parameters. For example, the constructed models might use incorrect material characteristics or physical measurements.

A typical experimental process consists of three parts (Fig. 3.10). During the first step, we generate synthetic data that describe shape deformation of the object and its position in space. The model is deformed according to an applied external impact. The deformation continues up to some final state. During the simulation, we store the tracks of the markers as well as the validation points. This data is used in the next parts of the experiment. In case of real data, this step is replaced by observed features, obtained from medical recordings.

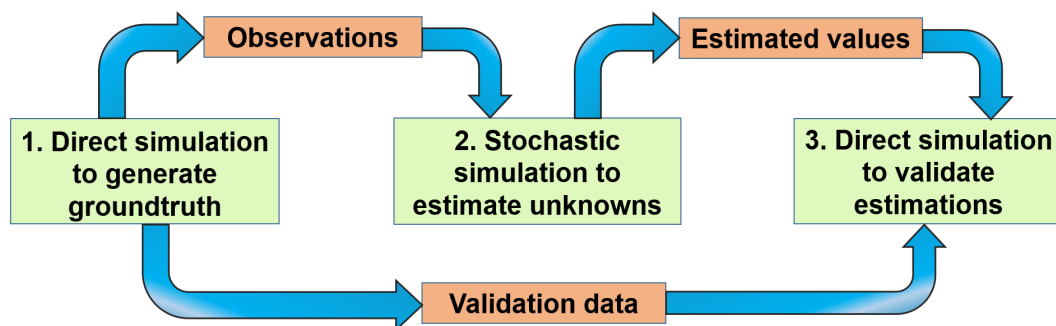


Figure 3.10: Overview of the experimental workflow. The process contains three steps. The first part relates to generating groundtruth observations and validation. Then, data assimilation process is performed to estimate the unknowns. Finally, the simulation based on estimated parameters is compared with the groundtruth one.

The second step is dedicated to Kalman filtering process. Based on available observations, the specified unknowns are estimated according to a selected Kalman filtering approach. In this process, the Young's modulus or elasticity of the selected domain is presented as unknown parameters with Gaussian distribution, as required by filter. The tracks, stored during the previous step, are used by filter as some observed features. As a result, some estimations of the unknowns are obtained. We also compute the time of the estimation process to compare its performance with the real-time requirements. All tests have been done on a computer with CPU Intel Core I7-6700K and 16 GB RAM.

Finally, a validation simulation is performed. In this step, the estimation is compared with the groundtruth data. For this, a set of selected validation points is used. In general, their positions are different from observations. Usually, they distributed more or less uniformly inside the mesh volume to approximate the deviations between deformations for the whole model. In case of real data processing, the groundtruth validation points are taken from CT images. For comparison, we generate the validation data based on the simulation with estimated parameters and compute the difference with the groundtruth points generated during the first step of the experiment. We then visualize the computed results for several estimated cases.

3.4.1 Object stiffness estimation

In the first experiment, we wanted to observe the potential of the nonlinear Kalman filtering methods in their ability to estimate properly the stiffness of a moving object. We decided to start from the Young's modulus estimation problem since this task is simple and it is easy to analyze the filtering behavior. For the experiment, we took a simple cylindrical object and applied a periodic force to simulate a continuous movement like in real surgical case. The overview of the experiment is illustrated in Fig. 3.11 left and its conditions are given in the list below:

- **Idea:** Young's modulus estimation
- **Domain geometry:** cylinder, 3 cm diameter, 18 cm length
- **Domain discretization:** 915 nodes and nearly 3000 finite elements
- **Material properties:** StVK material with Young's modulus equals to 8 kPa and Poisson's ratio to 0.49
- **Boundary conditions:** cylinder is fixed from both sides
- **Applied impact:** periodic force in indicated direction is applied
- **Observations:** 15 markers uniformly distributed on a part of the frontal surface (see Fig. 3.11)
- **Estimation method:** data assimilation using UKF, ETKF with 6 ensemble members, and ROUKF
- **Initial approximation:** Young's modulus is equal to 4 kPa and standard deviation to 350 Pa
- **Validation points:** 915 nodes of the mesh

For this experiment, we decided to observe the estimation process under ideal conditions. Thus, to avoid any errors caused by inaccurate modeling, we used the same discretization of the domain for generating the groundtruth and for data assimilation process.

It is also important to say that for all performed experiments we used groundtruth observations only during first 1600 iterations. But then we continued the process without them to show how the model with assimilated parameters will behave in case when observations are missing.

The results are presented in Fig. 3.11 right. They show that UKF as well as reduced versions are converging to the groundtruth Young's modulus, which indicates that for the ideal case the filters are able to find the correct solution. Also, the convergence rate for all versions of the filter is approximately the same.

Fig. 3.12 shows the average difference for validation points. They were computed for the groundtruth data and simulation with estimated parameters as well as initial approximation. The results show that if the Young's modulus is estimated correctly, the model behaves almost like the groundtruth one, while for the values taken as an initial approximation there is a certain difference in deformation.

Finally, the average and maximal relative errors with respect to deformation bounds of groundtruth data and the computation time are presented in Table. 3.1. According to these re-

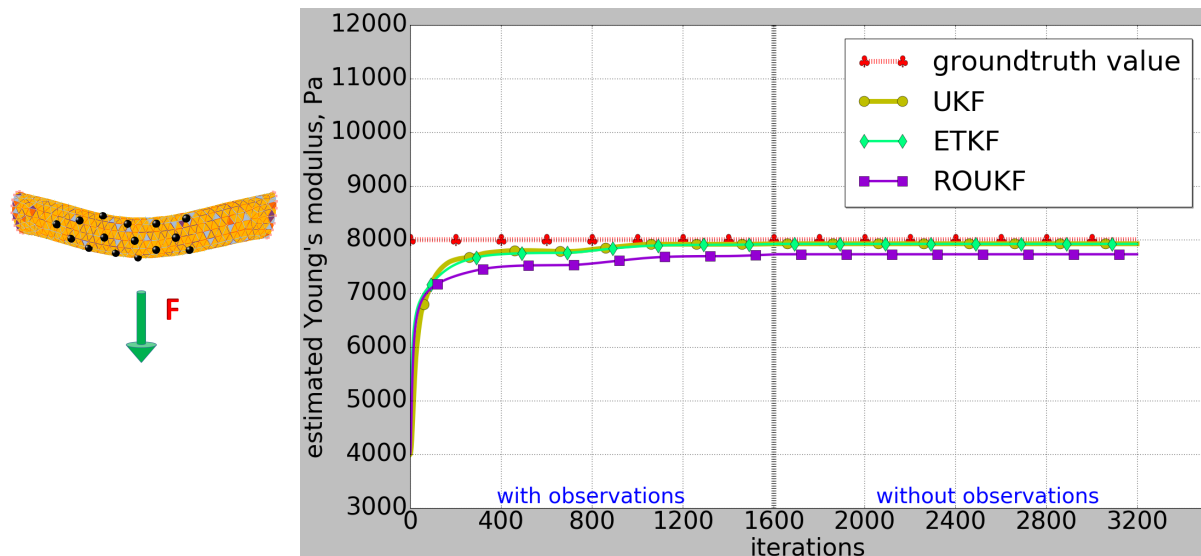


Figure 3.11: Left: Overview of the experimental model, attached observations, and applied force. Right: Estimation results of the Young's modulus for different filters. The meshes used to generate the groundtruth and to simulate the object deformations are the same.

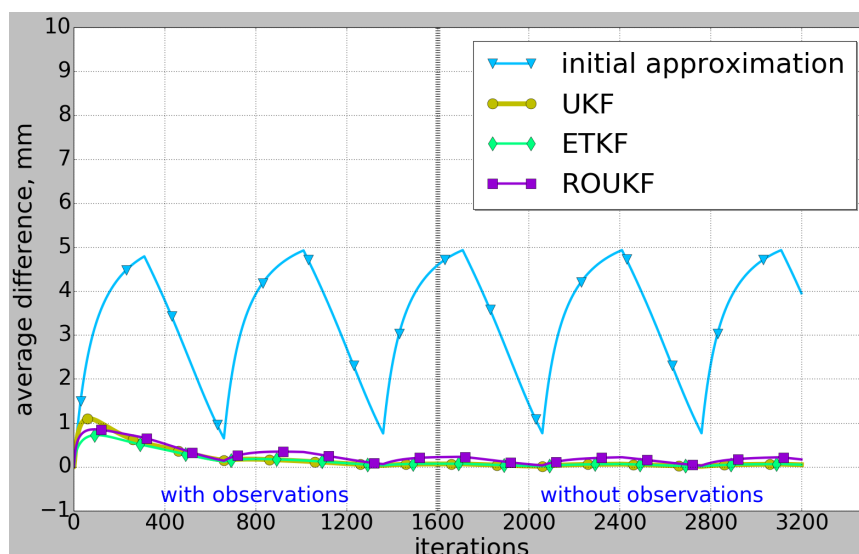


Figure 3.12: The average difference for validation points between groundtruth and simulation results obtained for initial and estimated Young's moduli. The meshes used to generate the groundtruth and to simulate the object deformations are the same.

sults, the estimation error for the reduced versions is less than 1 % bigger than for UKF, while the last one is 230 - 420 times slower. This can be explained by the fact that in UKF the filter state contains 915 model points, and therefore approximately 2700 sigma points are generated. Each of these points requires computing a simulation of deformation, so together it takes a lot

| Method | UKF | ETKF | ROUKF |
|---------------------------------------|---------|--------|--------|
| Average error wrt max. deformation, % | 0.135 | 0.143 | 0.48 |
| Maximal error wrt max. deformation, % | 0.282 | 0.303 | 1.0 |
| Computation time for one time step, s | 39.1899 | 0.1635 | 0.0915 |

Table 3.1: Young’s modulus estimation for the case when the meshes used to generate the groundtruth and to simulate the object deformations are the same. Estimation results relative comparison and computation time for different filters.

of time even for coarse meshes. In the next experiments, we are not going to consider the UKF anymore, because the huge computation time make it impossible to use in real-time context.

The second experiment was made to investigate the filter behavior in case of a more realistic scenario. To generate the deformation of an object for the groundtruth data more accurately, we used a denser domain discretization. We also took another hyperelastic material to express the fact that our simulation is different from the groundtruth. On the other hand, the estimation during surgical process has to be close to real time. So for data assimilation process we kept the same coarse model. The updated conditions for the experiment are presented in the list:

- **Idea:** Young’s modulus estimation
- **Domain geometry:** cylinder, 3 cm diameter, 18 cm length
- **Domain discretization: Groundtruth:** 11125 nodes and nearly 52000 finite elements, **Estimation:** 915 nodes and nearly 3000 finite elements
- **Material properties: Groundtruth:** Neo-Hookean material with Young’s modulus equals to 8 kPa and Poisson’s ratio 0.49, **Estimation:** StVK material with the same Poisson’s ratio
- **Boundary conditions:** cylinder is fixed from both sides
- **Applied impact:** periodic force in indicated direction is applied
- **Observations:** 15 markers uniformly distributed on a part of the frontal surface
- **Estimation method:** data assimilation using ETKF with 6 ensemble members and ROUKF
- **Initial approximation:** Young’s modulus is equal to 4 kPa and standard deviation to 350 Pa
- **Validation points:** 915 nodes of the coarse mesh

Fig. 3.13 depicts estimation results for reduced versions of filters. As in previous scenario, we estimated parameters during the first 1600 iterations and then removed the observations. According to the results, it is obvious that filters do not estimate the original value of the Young’s

modulus. Instead of that, they converge to some "effective" values. But with the approximated Young's modulus, the behavior of the coarser mesh, which is stiffer in general as shown in [211], is similar to the behavior of the groundtruth data. Also, both ROUKF and ETKF give very similar estimation results.

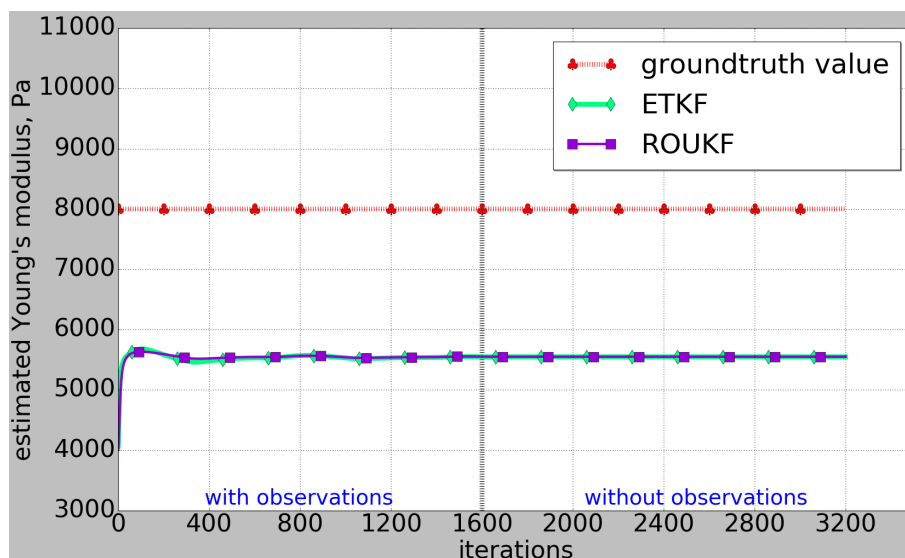


Figure 3.13: Estimation results of the Young's modulus for the reduced filter versions. The mesh used to generate the groundtruth is denser than the one used to simulate the object deformations.

The comparison for validation points is presented in Fig. 3.14. It is worth noting that, compared with the previous experiment, the difference between groundtruth and simulation based on estimated modulus is bigger. But, anyway, the model with estimated stiffness is more accurate than with initial one.

Table 3.2 shows the relative errors and computation time for both filters. Both ETKF and ROUKF have the same relative errors, but the second one is 44 % faster. This is explained by the fact that to estimate the Young's modulus in ROUKF only 2 sigma points are generated, while in ETKF 6 ensemble members are used. Of course, it is possible to reduce the number of the members up to 2, but then the result in ETKF becomes very dependent on each of them. And since a random element is included in the ensemble members generation, the result becomes unpredictable and sometimes quite unstable. Therefore ROUKF is more preferred for our estimations.

3.4.2 Estimation of boundary conditions

Unlike the Young's modulus identification, estimation of boundary conditions is a more complex problem. There are several reasons for this. First of all, it is an ill-posed problem since generally there exists an infinite number of BCs combinations, for which the considered object behaves the same way. Secondly, by observing only the part of the system and predicting properties outside the observed area, we are trying to solve an extrapolation problem. And in general

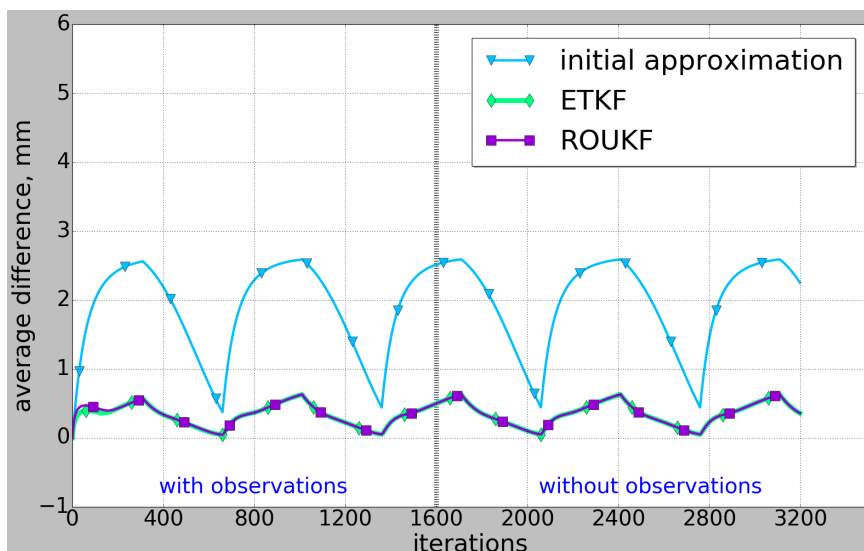


Figure 3.14: The average difference for validation points between groundtruth and simulation results obtained for initial and estimated Young's moduli. The mesh used to generate the groundtruth is denser than the one used to simulate the object deformations.

| Method | ETKF | ROUKF |
|---------------------------------------|--------|--------|
| Average error wrt max. deformation, % | 1.2 | 1.2 |
| Maximal error wrt max. deformation, % | 3.15 | 3.15 |
| Computation time for one time step, s | 0.1637 | 0.0902 |

Table 3.2: Young's modulus estimation for the case when the mesh used to generate the groundtruth is denser than the one used to simulate the object deformations. Estimation results relative comparison and computation time for the ETKF and ROUKF.

case, such type of problems do not have clear solutions. We can find only some statistical approximation. Finally, there is no well-known methodology to deal with boundary conditions. Usually, people model them by just fixing parts of the system in space and time, but for certain problems this is not an appropriate solution. Particularly, it relates to the case of liver modeling since the organ is attached to the other organs and tissues that are able to move under breathing and other impacts.

However, we want to note that even under these nontrivial conditions the Kalman filtering approaches are still able to perform some statistical estimations. Their results can improve the accuracy of the influence made by boundary conditions. As in previous experiments, we performed estimation during initial 1600 iterations and then continued the process to show the improvement of the model accuracy when there are no available observations.

In the first experiment, we wanted to verify how estimation of boundary conditions works in case of an ideal scenario. Therefore, as for the experiment with the Young's modulus estimation, we used the same domain discretization for the groundtruth generation and simulation process. The region of interest was represented by 3 cubic springs, which replaced fixed region (see Fig. 3.15 left). These springs connected the cylinder to some points fixed in space. The points were constructed by selecting several mesh points and their extrusion along the object normals. By boundary conditions estimation we consider estimation of the springs stiffness coefficients. The general conditions for the experiment are given below:

- **Idea:** spring stiffness estimation
- **Domain geometry:** cylinder, 3 cm diameter, 18 cm length
- **Domain discretization:** 915 nodes and nearly 3000 finite elements
- **Material properties:** StVK material with Young's modulus equals to 8 kPa and Poisson's ratio to 0.49
- **Boundary conditions:** cylinder is fixed from one side, 3 cubic springs with 3 cm length shared the common stiffness are attached to the other side. Groundtruth coefficients of springs are [4, 7, 20] N/m
- **Applied impact:** periodic force in indicated direction is applied
- **Observations:** 15 markers uniformly distributed on a part of the frontal surface (see Fig. 3.15)
- **Estimation method:** data assimilation using ETKF with 6 ensemble members and ROUKF
- **Initial approximation:** coefficients equal to [10, 10, 10] N/m and standard deviations equal to [4, 4, 4] N/m
- **Validation points:** 915 nodes of the mesh

The estimation results for all spring coefficients are presented in Fig. 3.15 right. They show that filters are not able to estimate the groundtruth values, while they still have a general tendency to converge with them. This can be explained by the fact that there exists many solutions for which the object behaves the same way, so the filters just use the most probable one as estimation result. Also, we can say that both versions of reduced filters converge to slightly different values.

Fig. 3.16 presents average difference between validation points positions for different estimated coefficients. Compared with the original stiffness, the estimations made by filters help to improve the modeling accuracy of the object deformation. Also, despite the fact the filters find slightly different values, the behavior of the object for the estimated values is almost the same.

Table 3.3 shows the maximal and average relative error with respect to cylinder deformation and computation time results. According to the results, the difference between estimations does not exceed 1 %, but the reduced-order unscented Kalman filter is approximately 24 % faster.

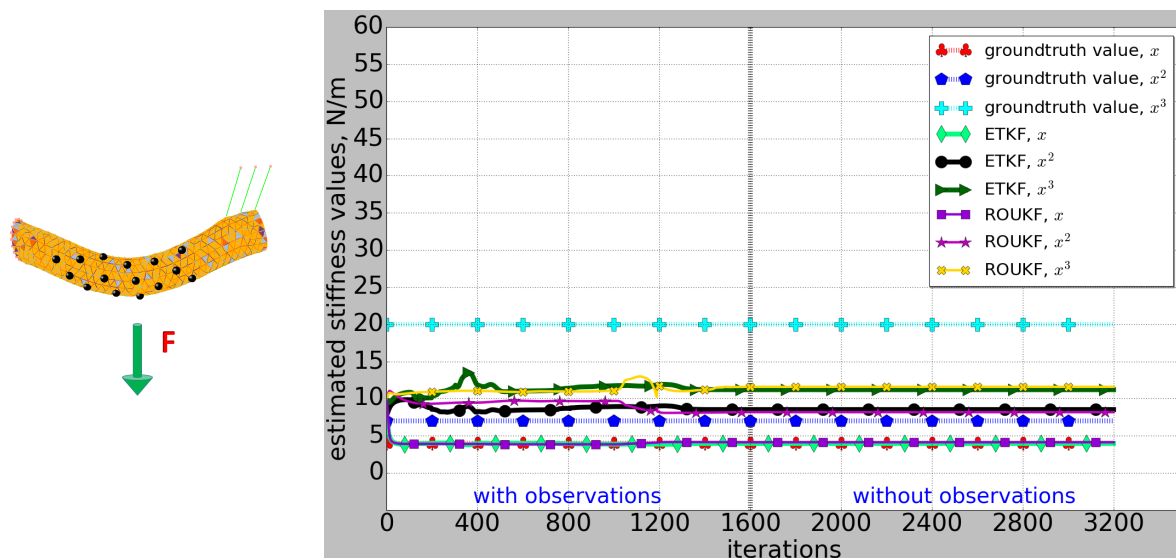


Figure 3.15: Left: Overview of the experimental model, attached springs, observations, and applied force. Right: Estimation results of cubic springs coefficients for different types of filters. The meshes used to generate the groundtruth and to simulate the object deformations are the same.

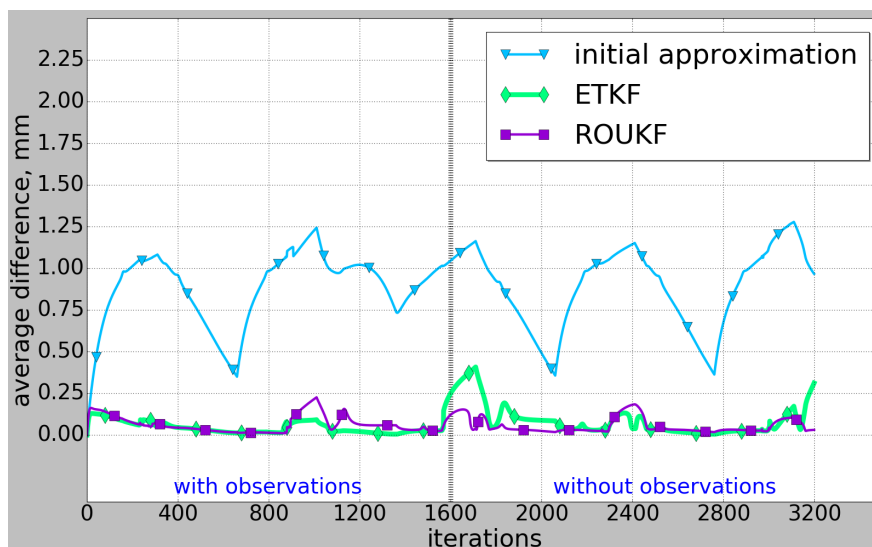


Figure 3.16: The average difference for validation points between groundtruth and simulation results obtained for initial and estimated spring coefficients. The meshes used to generate the groundtruth and to simulate the object deformations are the same.

The next experiment was dedicated to a more realistic scenario. To generate groundtruth data, we took a denser mesh discretization. However, we did not change the hyperelastic material, since springs attached to a Neo-Hookean model caused some instabilities during the sim-

| Method | ETKF | ROUKF |
|---------------------------------------|--------|--------|
| Average error wrt max. deformation, % | 0.55 | 0.31 |
| Maximal error wrt max. deformation, % | 5.56 | 4.61 |
| Computation time for one time step, s | 0.1876 | 0.1425 |

Table 3.3: Springs coefficients estimation for the case when the meshes used to generate the groundtruth and to simulate the object deformations are the same. Estimation results relative comparison and computation time for the ETKF and ROUKF.

ulation process. Like in previous case, we estimated the parameters of springs. The conditions of the experiment are given further:

- **Idea:** spring stiffness estimation
- **Domain geometry:** cylinder, 3 cm diameter, 18 cm length
- **Domain discretization: Groundtruth:** 11125 nodes and nearly 52000 finite elements, **Estimation:** 915 nodes and nearly 3000 finite elements
- **Material properties:** StVK material with Young's modulus equals to 8 kPa and Poisson's ratio to 0.49
- **Boundary conditions:** cylinder is fixed from one side, 3 cubic springs with 3 cm length shared the common stiffness are attached to the other side. Groundtruth coefficients of springs are [4, 7, 20] N/m
- **Applied impact:** periodic force in indicated direction is applied
- **Observations:** 15 markers uniformly distributed on a part of frontal surface (see Fig. 3.15)
- **Estimation method:** data assimilation using ETKF with 6 ensemble members and ROUKF
- **Initial approximation:** coefficients equal to [10, 10, 10] N/m and standard deviations equal to [4, 4, 4] N/m
- **Validation points:** 915 nodes of the coarse mesh

The estimation results are presented in Fig. 3.17. Unlike the previous experiment, filters converged to completely different values. However, in both cases, the estimated values improve the simulation accuracy. The modeling difference is decreased from 4 mm up to approximately 2.3 mm, as Fig. 3.18 shows.

Table 3.4 depicts the relative error and computation time for both filters. In this experiment, the estimation result of the ETKF is a bit better than the result made by the ROUKF, but the latter one is, as usual, faster averagely on 26%.

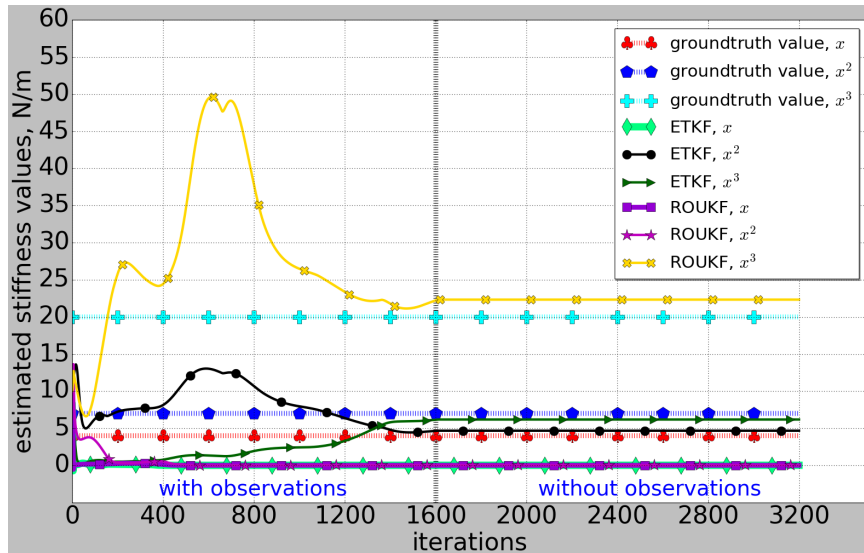


Figure 3.17: Estimation results of cubic springs coefficients for the ETKF and ROUKF. The mesh used to generate the groundtruth is denser than the one used to simulate the object deformations.

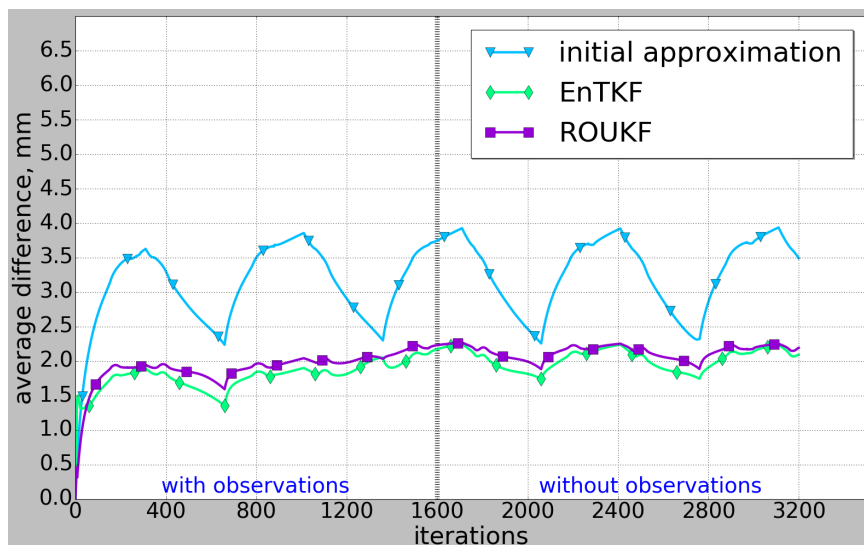


Figure 3.18: The average difference for validation points between groundtruth and simulation results obtained for initial and estimated spring coefficients. The mesh used to generate the groundtruth is denser than the one used to simulate the object deformations.

The performance of the ROUKF is explained by the fact that to estimate three unknown parameters, in case of the simplex topology, we need to generate only 4 sigma points for every simulation step. And for the ensemble transform Kalman filter we still keep 6 ensemble members.

If we reduce the number of ensemble members, the performance of the ETKF will increase,

| Method | ETKF | ROUKF |
|---------------------------------------|-------|-------|
| Average error wrt max. deformation, % | 3.05 | 3.07 |
| Maximal error wrt max. deformation, % | 6.08 | 6.14 |
| Computation time for one time step, s | 0.192 | 0.142 |

Table 3.4: Springs coefficients estimation for the case when the mesh used to generate the groundtruth is denser than the one used to simulate the object deformations. Estimation results relative comparison and computation time for the ETKF and ROUKF.

but this might lead to stability problems. The reason for this is still a subject for research, but we guess that for a few ensemble members the filter becomes too dependent on each of them. Therefore, if one of them is converging to a completely different result, which is typical for ill-posed problems, the statistical result also deviates. In the end, this might cause instabilities in the simulation.

3.4.3 Correction of boundary conditions

In the final experiment, we wanted to investigate the estimation process in case of BCs modification. During surgical procedure, surgeons often cut ligaments or perform other operations that modify the model topological structure. So it would be nice to observe how the estimation process reacts to such changes.

For this experiment, we took the same model as the one used for boundary conditions estimation. The main difference is that at some moment we performed the emulation of a "cutting procedure". In it, the shared stiffness of the springs was set to a zero value. Then the deformation process continued up to the final stage.

Therefore, there are three modeling segments. For the first 1600 iterations, the estimation went like in previous cases, then we performed "cutting", and during next 1600 iterations we kept performing estimation for the modified system. Finally, we eliminated observations to watch how accurately the corrected model behaves. The conditions of the experiment are given below:

- **Idea:** spring stiffness estimation in case of a "cutting procedure"
- **Domain geometry:** cylinder, 3 cm diameter, 18 cm length
- **Domain discretization: Groundtruth:** 11125 nodes and nearly 52000 finite elements, **Estimation:** 915 nodes and nearly 3000 finite elements
- **Material properties:** StVK material with Young's modulus equals to 8 kPa and Poisson's ratio to 0.49

- **Boundary conditions:** cylinder is fixed from one side, 3 cubic springs with 3 cm length shared the common stiffness are attached to the other side. Groundtruth coefficients of springs are [4, 7, 20] N/m
- **Applied impact:** periodic force in indicated direction is applied. After 1600 iterations, a "cutting procedure" is performed, where the stiffness of springs is set to zero
- **Observations:** 15 markers uniformly distributed on a part of frontal surface (see Fig. 3.15)
- **Estimation method:** data assimilation using ETKF with 6 ensemble members and ROUKF
- **Initial approximation:** coefficients equal to [10, 10, 10] N/m and standard deviations equal to [4, 4, 4] N/m
- **Validation points:** 915 nodes of the coarse mesh

The estimation results are presented in Fig. 3.19. For the better view, a cubic scale was applied to the estimated parameters. Looking at them, we might say that the estimation process quickly recognizes changes in the system and modify the estimated parameters accordingly.

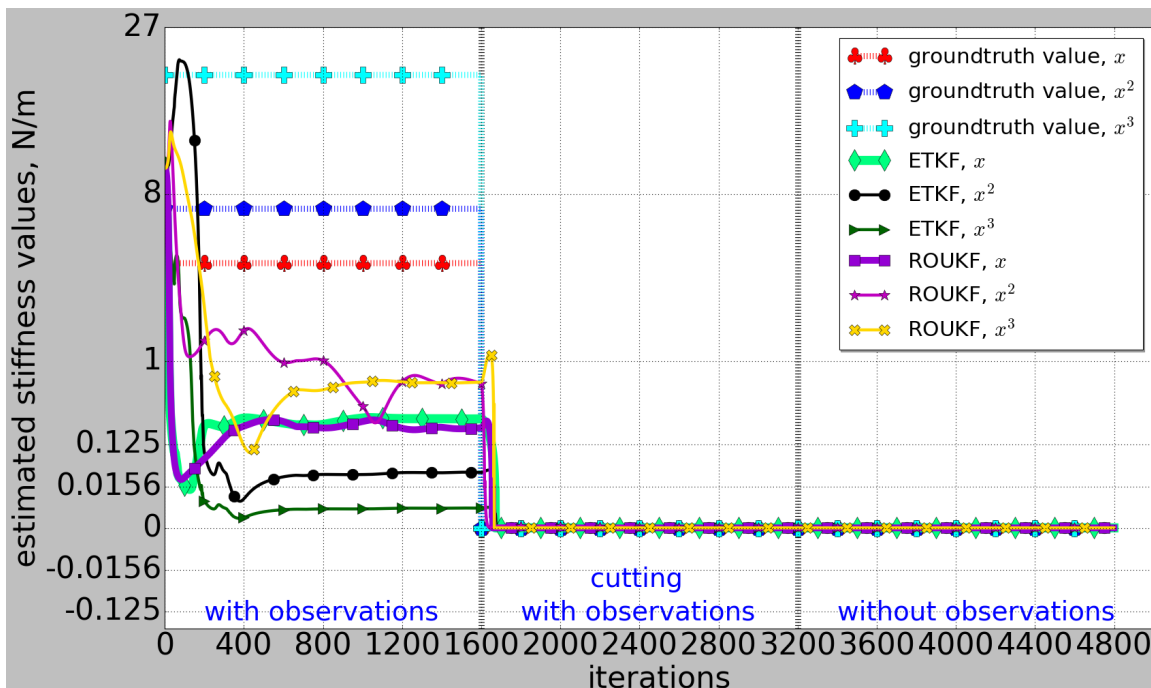


Figure 3.19: Estimation results of springs coefficients after the "cutting procedure" for the ETKF and ROUKF. The mesh used to generate the groundtruth is denser than the one used to simulate the object deformations.

Fig. 3.20 shows the difference for validation points in cases when parameters were corrected after the "cutting procedure" and were originally estimated without it. To show the differences more clearly, the difference distances are presented in a squared scale.

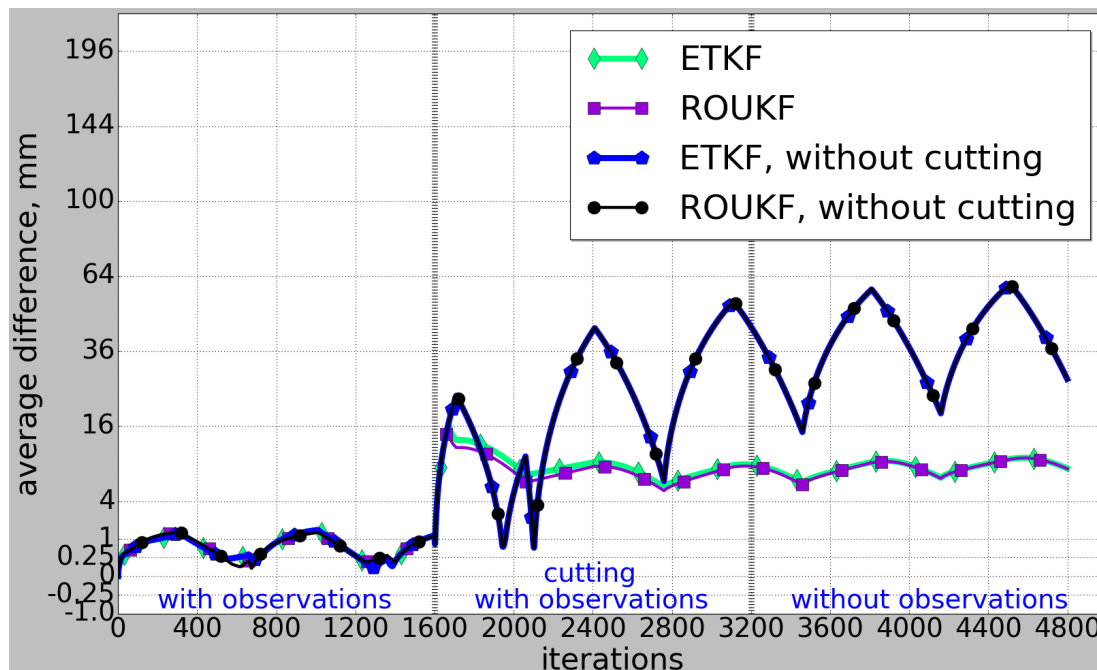


Figure 3.20: The average difference for validation points between groundtruth and simulation results. The initially estimated coefficients are compared with ones estimated after the "cutting procedure". The mesh used to generate the groundtruth is denser than the one used to simulate the object deformations.

It is worth noting that parameters correction after the cutting procedure will sufficiently improve the accuracy of the model. However, the difference between estimations after cutting and groundtruth values is still big. This can be explained by the difference between meshes used for the groundtruth generation and the estimation process. After cutting, these differences cannot be compensated anymore by correcting spring parameters, which finally results in a bigger error, compared with results from other experiments.

Table 3.5 shows the relative difference for cases when estimations are corrected after the

| Method | ETKF | ROUKF | ETKF without cutting | ROUKF without cutting |
|--|-------|-------|-------------------------|--------------------------|
| Average error wrt max. deformation, % | 8.93 | 8.26 | 33.21 | 33.30 |
| Maximal error wrt max. deformation, % | 20.30 | 18.60 | 80.91 | 81.01 |

Table 3.5: Springs coefficients estimation for the case when the mesh used to generate the groundtruth is denser than the one used to simulate the object deformations. Estimation results relative comparison for the ETKF and ROUKF in cases when spring parameters are corrected after the "cutting procedure" and not.

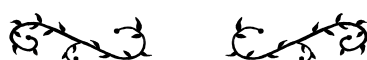
"cutting procedure" and are not. As in other experiments, both versions of filters show approximately the same accuracy. The small average errors compared with maximal ones can be described by the fact that a certain part of simulation of the object happens near rest shape position, where differences in simulations are small.

As a result of these experiments, it is worth saying that filtering approach can be used not only to estimate the initial parameters, but also to correct them in time. This correction can be done for various objects, and the deformation does not need to be necessarily simple and straightforward.

All in all, based on the all performed experiments we decided to stick to the ROUKF approach as giving more stable results, while taking less computation time.



ESTIMATION OF LIVER BOUNDARY CONDITIONS



Contents

| | | |
|------------|---|-----------|
| 4.1 | Boundary conditions identification | 84 |
| 4.2 | The estimation approach | 85 |
| 4.3 | Statistical atlas construction | 86 |
| 4.3.1 | Overview of the available approaches | 86 |
| 4.3.2 | Large deformation diffeomorphic metric mapping | 89 |
| 4.3.3 | Atlas construction procedure | 91 |
| 4.4 | Ligaments properties initialization | 92 |
| 4.5 | Preconditioning for data assimilation procedure | 93 |
| 4.5.1 | Conjugate gradients and system preconditioning | 94 |
| 4.5.2 | Preconditioned data assimilation process | 95 |
| 4.5.3 | Experiments with preconditioning | 96 |
| 4.6 | Constructed process for boundary conditions estimation | 99 |

In the previous chapter, we presented the intention to estimate boundary conditions for the liver using a nonlinear Kalman filter. But this approach goes with certain prerequisites. Firstly, we need to have an initial approximation of the liver boundary conditions, which, in general, is not available for a specific patient. Secondly, the estimation has to be done close to real time, since it is based on observations obtained during surgery for the biomechanical model used in the same surgery. In this chapter, we will give a general outline and the details of our process, which holds the possible solutions for the given issues. We will present a statistical atlas to construct an initial approximation of ligaments position on the liver model. We will also show how the initial spring parameters are approximated based on an available constitutive law. Finally, the preconditioning approach that allows accelerating the simulation process will be described.

4.1 Boundary conditions identification

The idea of identifying BCs is not new. In the area of biomechanics, it has been already addressed by a few authors.

Some approaches attempt to specify BCs based on statistics obtained from real or generated synthetic data. The work of Hu et al. [82] is dedicated to registration method using a statistical motion model for image-guided prostate interventions. In this work, a Finite Element patient-specific model is built using preoperative MRI data. To create a statistical motion model, which is needed for further registration process, a set of deformations with randomly sampled material properties and boundary conditions are computed and evaluated statistically. However, for such analysis a good description of object boundaries is needed, including available possibilities for their positions and biomechanical properties.

The authors from [87] describe a training database of liver deformations obtained from medical data. As available data, the authors have both preoperative and intraoperative CT images, from which they generate models and their displacements, as well as information about abdominal pressure and boundary conditions. When preoperative images for a new liver are obtained, the similarities between this data and database models are computed. The intraoperative model together with BCs are then computed as a weighted average of displacements observed in database models. However, first of all, this method requires a database of intraoperative CT images, which is hard to get. It also models boundary conditions as zero displacement constraints, which does not take into account their properties and a possibility of modification over time.

Plantefevé et al. [187] proposed a method for modeling boundary conditions in deformable anatomical structures using a statistical atlas. The atlas gathers information about the connective structures attached to the organ. Then it is transferred to a patient-specific anatomy using a physically based registration technique. The resulting BCs are computed based on the mean position and variance of the ligaments. While this approach already shows a benefit in integrating this information in the model, it has certain limitations. First of all, atlas construction takes into account only rigid transformations, while the liver shape of different patients might vary significantly. It also implicitly implies knowing the correspondences between liver meshes, which in general is not the case. Finally, it is neither truly patient specific, nor models the parameters of boundary conditions correctly.

Another general way to determine BCs is to apply various estimation algorithms. In [169], the authors estimate what they call compliance boundary conditions. Recently, this approach was validated for several scenarios [170]. The authors suppose that the observed area of an object is attached to the remaining part of it with aligned springs. These springs create an additional restriction for the deformation, whose properties could be estimated by comparing unconstrained and constraint motions. Anyway, for this approach, in the case of a volumetric object, a sufficient number of internal observations need to be made and, moreover, they have to be accurate enough.

In [178], the authors try to estimate BCs based on two deformed configurations of a liver. Matching the shapes gives a constraint that was applied to get the transformation between both configurations. The researchers present boundary conditions as Lagrangian multipliers with unknown parameters, and shape matching allows estimating them. Unfortunately, two different

shapes are necessary for this approach, whereas, in common, for diagnosis purposes surgeons take only one.

In [112], an inverse simulation method is proposed. The authors present BCs as forces with unknown intensities. To obtain a solution for unknowns, they apply traction and solve a gap minimization problem between simulated and observed positions. Here, it is implicitly supposed that the model is accurate enough to simulate the real object. But in the context of AR, the accuracy is often waived in favor of performance. The observed positions also have to be measured with high accuracy.

In case, when observations are obtained through a laparoscopic camera, only several dozens of features can be detected on the liver. These features are usually located on the visible part of the liver, so almost no information about boundary positions are given. Moreover, the number of parameters that can be estimated is highly dependent on the amount of given data. Therefore the methods described above are not applicable in this context. However, in our work, we follow the idea of the statistical atlas, which is still relevant since the available information can provide us with some initial approximation related to BCs.

4.2 The estimation approach

In this section, we describe the main steps of our method. We keep an important focus on patient-specific modeling and the possibility to deploy our method in a clinical context without the need for additional equipment. Fig. 4.1 depicts an overview of our solution.

In general, all connective tissues have nonlinear properties, and, thus, we use a nonlinear model to simulate the liver BCs. The model of the boundary conditions has to take into account the anatomy of a specific patient. The essential moment here is that in general the available information is limited to a preoperative model and intraoperative images, which cannot indicate us the exact locations of boundary conditions. Therefore, we try to identify them in two steps. Initially we create a generalized initial approximation of the BCs. Then we correct it using an extension of the Kalman filter. The approximation is based on patient-specific data obtained from a laparoscopic camera or some other modality used during surgical intervention.

For the initial approximation, the general idea is to use a statistical atlas from available datasets (like in [187]) and stress-strain curve data from the literature. Despite the fact that an atlas cannot give the exact properties of boundary conditions for a certain patient, it still helps to restrict the variability of locations and their characteristics.

In order to correct BCs with the available data, our Kalman filtering approach takes into account that we have some uncertainty both in the numerical solution and the intraoperatively obtained information. Therefore we combine the benefits of both initial approximation and correction steps to make fast and robust estimation in a patient-specific context.

The Kalman filter was described in detail in the Chapter 3; therefore, we will now focus on constructing the initial approximation of the BCs for the liver and the optimization process.

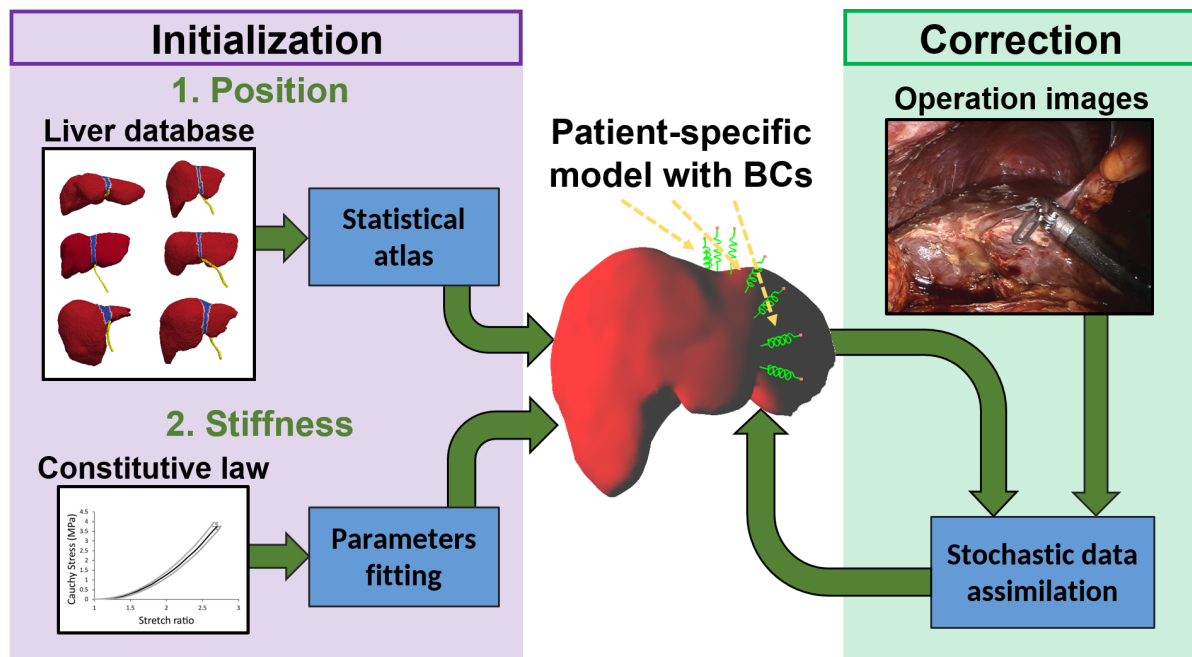


Figure 4.1: General identification process of BCs. The BCs are presented as cubic springs with unknown parameters. The estimation workflow contains two steps. 1 - Initial approximation based on statistics from liver database and constitutive laws from the literature. 2 - Correction based on intraoperative images. Ligament constitutive law taken from [246].

4.3 Statistical atlas construction

4.3.1 Overview of the available approaches

The prediction of unknown information that is based on statistics is quite a general approach, used in various domains of science. Yet, it has particular importance for medical imaging and biomechanics where, in general, images can provide us only with limited amount of information. The general idea is to create a database with detailed models or images and to segment all necessary anatomical details using the most accurate techniques and help of experts. From that, a statistical atlas is constructed, which can be used to describe approximately the anatomy for a general patient.

For example, the statistical atlases are quite often used in brain image analysis [48, 51, 194], where they help to detect various brain deviations that cause diseases. Another domain covers bones research [16, 128] where atlases help to perform their segmentation in medical images.

Since the liver is a common target for surgery, there is also a certain number of works related to atlas creation for liver models. There are two general types of regarded atlases: probabilistic and statistical ones.

Liver probabilistic atlases are commonly used for liver segmentation. In general, for every single space element, the probabilistic atlas shows its probability to be a part of an organ. For

example, in [45], the authors describe probabilistic atlas for the liver and, in [171, 172], for the whole abdominal area. Based on available segmented images, they compute the probability for every voxel to be inside the liver. It is taken as a number of images where it is inside the organ over their whole number. For the mean, a set of voxels with probability more than 50 % is taken, and diapason from 0 % to 100 % shows the variance of organ shapes.

In [21], a probabilistic atlas is created by computing probability using Gaussian mixture models. In [44], the authors propose to compute the combination of the distance function mean and the coverage function mean computed on images. In [252], to obtain an atlas, a superposition for all images is computed using the Procrustes method, and then they are mapped via optimal diffeomorphic deformation.

Probabilistic atlases are good for processing images to perform, for example, organ segmentations, but not so useful to work with shapes. However, in the context of our work, we are dealing with three-dimensional biomechanical model. Consequently, liver BCs have to be segmented on three-dimensional organ shapes. So, the statistics will be computed based on meshes. In this case, a statistical atlas is a more appropriate solution.

Statistical atlases or statistical shape models can be presented as a set of three-dimensional shapes with a mean and variations. Commonly, a reference frame is selected, on which all available shapes are registered. Then, for shape deviation analysis, a principal component analysis (PCA) is used.

A certain number of research works are dedicated to the construction of a statistical atlas for the liver. In [107], a liver statistical shape model is described. Nevertheless, the correspondences between shapes are computed using manually annotated features. They might be difficult to find on liver surfaces and require a lot of work for a database with hundreds of models. In [254], the authors construct both probabilistic and statistical models. They use CT images of the whole abdominal area and align them based on the spinal column. After that, they compute the modes of the liver variations. But the result gives rather variations of the liver location in the abdominal area than variations of the liver shape.

Some works perform attempts to improve model database registration and determination of markers correspondence. In [127, 129], the modification of an iterative closest point (ICP) approach is used to register all models on a selected reference frame. Then a set of uniformly distributed points over the shapes are generated, and correspondence between these points is constructed. Finally, a principal component analysis is performed for the correspondences. In the paper [59], the researchers use the minimal description length approach to find correspondent points and carry out a generalized Procrustes analysis for models registration. In addition, they improve the atlas by classifying database models into different populations. A separate atlas is constructed for every population. For analysis, the correspondent atlas is selected according to the target model classification. Unfortunately, the registration for correspondence searching is based on rigid transformations, which does not take into account the variability of the liver shape.

In [108], the main idea is to split liver surface on several patches and to compute correspondences for every segment. To segment patches, a bundle of bound points has to be marked manually, connected by the shortest geodesic path computed later. Since all liver meshes have different topologies, the idea is to map every segment onto a plane and find correspondences on

this plane. The main drawback of this approach is that atlas construction still requires manual work because the patches of borders have to be marked.

In [100, 101, 232], the researchers propose another method to generate a liver atlas. Using the marching cube algorithm, they construct triangulated meshes. Then they use non-rigid registration taken from [29] to find the correspondent points and PCA to obtain the variations. The proposed approach does not require a cloud of correspondent points for every shape and allows performing a non-rigid registration. However, it is not obvious whether this approach allows handling large variations between organs shapes. Also, the final result relates to the analysis of the liver shape itself, but not to its anatomical structures.

Several other papers try to improve the accuracy of the statistics. In [201], in order to compute the anatomical variability, the authors use a principal factor analysis [3] instead of generally used PCA. In [76], the key moment is to construct a special Lie group, based on mesh triangle positions, and compute mean and statistics using mathematical transformations of this group. In [163], a multi-level statistical shape models are proposed. The idea is to organize all organs and parts hierarchically starting from an abdominal area. To link the regions in a simple model, the authors introduce the so-called adhesive constraints. In [164], the authors extend their approach to compute statistics for many organs. But all proposed solutions still require the presence of correspondences between elements, over which the statistics will be computed.

An interesting idea is given in [228]. The researchers try to construct a combination of statistical and probabilistic atlases. Together with the statistical model creation, a set of features is also extracted, and later they are used as additional conditions. To get the range of available values, a mean and a variance are identified for the features. Lastly, for a test data, a correlation between observed features and feature statistics is computed. Based on the result, the set of available features can be taken into account, which will simplify elements recognition and segmentation.

All in all, the papers described above mainly deal with the liver shape, which helps to analyze the variations and execute segmentation in medical images. However, the main idea of the current work is to identify the boundary conditions around the liver, but not the variation of its shape.

A more relevant work is described in [187]. The authors discussed the construction of an atlas for BCs and its registration on a target model. Initially, they register a set of correspondent points on each mesh from the database. This step is not described in detail, but in [186], the authors specify that the registration was done with a biomechanical model. To create a set of correspondent points, a selected liver shape is scaled and non-rigidly registered on all other meshes. From this clouds a generalized Procrustes analysis is performed to align all models in a common reference frame. Then, like in other methods, a PCA is executed for each structure of the interest, including boundary conditions, to compute the principal modes of the deformation. The final step of the approach is atlas to patient registration and boundary conditions projection. To find the BCs of a patient model, the authors use a biomechanically based registration on a preoperative data. This registration is guided by dynamically weighted iterative closest point.

The main issue of the described approach is that it is difficult to find a set of correspondent points, necessary for further processing, since liver parenchyma shape varies a lot from one patient to another. A biomechanically based registration allows compensating for object defor-

mations, but not for anatomical differences in liver structures such as, for example, lobes. So this method is quite questionable for the case when a liver with a specific shape has to be registered to another one. Also, the alignment of models is based on rigid transformation; therefore, PCA analysis includes variations not only for ligament positions, but also for the liver parenchyma. Finally, the authors analyzed the positions of several anatomical landmarks, but no investigation was performed to compare ligaments positions.

Consequently, we propose a new approach for atlas construction related to boundary conditions, which we will describe hereafter.

4.3.2 Large deformation diffeomorphic metric mapping

The proposed approach is based on a solution that is called *large deformation diffeomorphic metric mapping* (LDDMM). The main idea is to present an object mesh deformation or displacement as a diffeomorphic transformation, independent on the mesh topology, and express shape variations through its properties. In this way, for any shape, the transformation parameters define bundle of shapes that are diffeomorphic to the original one. The similarity between shapes can be measured as amount of deformation (displacement) needed to transfer one shape to another. One possible instance of such a construction is to do a parametrizing with a set of control points [19, 47]. The modifications of the object shape are then reconstructed by interpolation functions based on the control points.

So, in the interpolation framework, the displacement $\mathbf{v}(s) = \Phi(s) - s$ of any transformation Φ and simple element s can be presented as a sum of radial functions based on control point positions:

$$\mathbf{v}(s) = \sum_{i=1}^{N_p} \Gamma(s, g_i) w_i \quad (4.1)$$

where g_i is a set of control points, N_p is their amount, w_i is a set of weight coefficients, which are called *moments*, and Γ is an *interpolation kernel*. The interpolation kernel is a scalar function that takes any pair of values. In [47], a Gaussian kernel is used, with $\Gamma(x, y) = \exp(-\|x - y\|^2 / \eta^2)$. The kernel width η controls the size of the generated deformation patterns.

The set of displacements, constructed in such a way, forms a space where the inner product for elements can be defined:

$$\langle \mathbf{v}_a, \mathbf{v}_b \rangle_{\mathbf{v}} = \sum_{i=1}^{N_p^a} \sum_{k=1}^{N_p^b} \Gamma(g_i^a, g_k^b) (w_i^a)^T w_k^b \quad (4.2)$$

To handle the large displacements, the idea is to use an instantaneous velocity field instead of the displacement one. Considering the control points g_i and moments w_i as trajectories, the velocity field at any instant l for an element s can be written as:

$$\mathbf{v}_l(s) = \sum_{i=1}^{N_p} \Gamma(s, g_i(l)) w_i(l) \quad (4.3)$$

for $l \in [0, l_{max}]$. For every positive l , the equation given above forms a diffeomorphism of the initial object. More details are given in [47].

The same equation can also be applied to specify the control points. In case of the moments w_i , the possible solution might be to choose trajectories that minimize the kinetic energy along the path. Therefore the equations that drive the motion are now given as:

$$\begin{cases} \dot{g}_k(l) = \sum_{i=1}^{N_p} \Gamma(g_k(l), g_i(l)) w_i(l) \\ \dot{w}_k(l) = -\sum_{i=1}^{N_p} w_k(l)^T w_i(l) \nabla \Gamma(g_k(l), g_i(l)) \end{cases} \quad (4.4)$$

A set of points can be combined in a single vector, and then the system can be solved using some numerical method. It is important to note that the flow of transformation is now entirely parametrized by the initial positions of control points $g_i(0)$ and initial moments $w_i(0)$.

To compare two meshes, some metric is needed. Among approaches that do not require correspondences, there exist a metric based on varifolds [24]. The basic idea behind a varifold is to present smooth curves or surfaces as distributions of basic curvatures in an extended space, where they could be compared with specific metrics. Particularly, the distance between two meshes can be computed as:

$$\begin{cases} dist_V(\mathcal{M}_1, \mathcal{M}_2)^2 = \|\mathcal{M}_1 - \mathcal{M}_2\|_V = \langle \mathcal{M}_1, \mathcal{M}_1 \rangle_V + \langle \mathcal{M}_2, \mathcal{M}_2 \rangle_V - 2\langle \mathcal{M}_1, \mathcal{M}_2 \rangle_V \\ \langle \mathcal{M}_a, \mathcal{M}_b \rangle_V = \sum_i \sum_k \Gamma_V(e_i^a, e_k^b) \frac{((n_i^a)^T n_k^b)^2}{\|n_i^a\| \|n_k^b\|} \end{cases} \quad (4.5)$$

here \mathcal{M} is a surface of a mesh, e is a center point of a mesh face, and n is its normal. Γ_V is a kernel of a varifold, which is usually a Gaussian function.

The varifold metric is too general and does not allow identification of the transformation between objects in a unique way. To perform a non-rigid registration, one possible way is to use the distances based on diffeomorphisms to measure the transformation increment. For two objects \mathcal{M}_1 and \mathcal{M}_2 , let Φ^{diff}_l where $l \in [0, l_{max}]$ be a diffeomorphic transformation such that $\Phi^{diff}_0(\mathcal{M}_1) = \mathcal{M}_1$ and $\Phi^{diff}_{l_{max}}(\mathcal{M}_1) = \mathcal{M}_2$. Then the difference between two meshes can be defined as the length of the trajectory with the minimal kinetic energy:

$$dist_{\Phi^{diff}}(\mathcal{M}_1, \mathcal{M}_2)^2 = \sum_{i=1}^{N_p} \sum_{k=1}^{N_p} w_i(0)^T \Gamma(g_i(0), g_k(0)) w_k(0) = \mathbf{w}_0^T \Gamma(\mathbf{g}_0, \mathbf{g}_0) \mathbf{w}_0 \quad (4.6)$$

where $\mathbf{g}_0 = \mathbf{g}(0)$ and $\mathbf{w}_0 = \mathbf{w}(0)$ are the vectors constructed on the sets of control points and moments, such that $\Phi^{diff}_{l_{max}, \mathbf{w}_0}(\mathcal{M}_1) = \mathcal{M}_2$ [47]. The initial moment, in its turn, is selected the way to minimize the varifold distance between transformed and target shapes:

$$\mathbf{w}_0 = \underset{\mathbf{w}}{\operatorname{argmin}} [dist_V(\Phi^{diff}_{l_{max}, \mathbf{w}_0}(\mathcal{M}_1), \mathcal{M}_2)]^2 \quad (4.7)$$

Thus, the distance between varifolds is used to find the deformed shape that is close to the target one. And the correspondent diffeomorphic distance depicts the amount of the transformation.

Finally, for a set of shapes registered non-rigidly, a statistically average shape is defined as a Fréchet mean, which minimizes the shape variance: $\mathcal{M}_{atlas} = \underset{\mathcal{M}}{\operatorname{argmin}} [\sum_i dist_{\Phi^{diff}}(\mathcal{M}, \mathcal{M}_i)]^2$.

Since the computation of the diffeomorphism looks for the minimization of the varifold metric, the final solution is obtained as a combination of two minimization problems:

$$\mathcal{M}_{atlas} = \operatorname{argmin}_{\mathbf{g}_0[\mathbf{w}_0^i]} \left[\sum_{i=1}^{N_{meshes}} \left(\operatorname{dist}_V [\Phi_{l_{max}, \mathbf{w}_0^i}^{diff}(\mathcal{M}_{atlas}), \mathcal{M}_i]^2 + (\mathbf{w}_0^i)^T \Gamma(\mathbf{g}_0, \mathbf{g}_0) \mathbf{w}_0^i \right) \right] \quad (4.8)$$

The mean shape construction and non-rigid registration procedures, depicted above, are used to generate statistics for ligament positions.

4.3.3 Atlas construction procedure

To construct the atlas for ligaments, the main idea is to perform a non-rigid registration, which is based on diffeomorphic metric mapping, of the liver models with segmented ligaments on the average shape. This allows determining a probability distribution of the ligaments location on a human liver (see Fig. 4.2). The average shape is the most natural choice for statistics computation since it does not have any particular anatomical deviations.

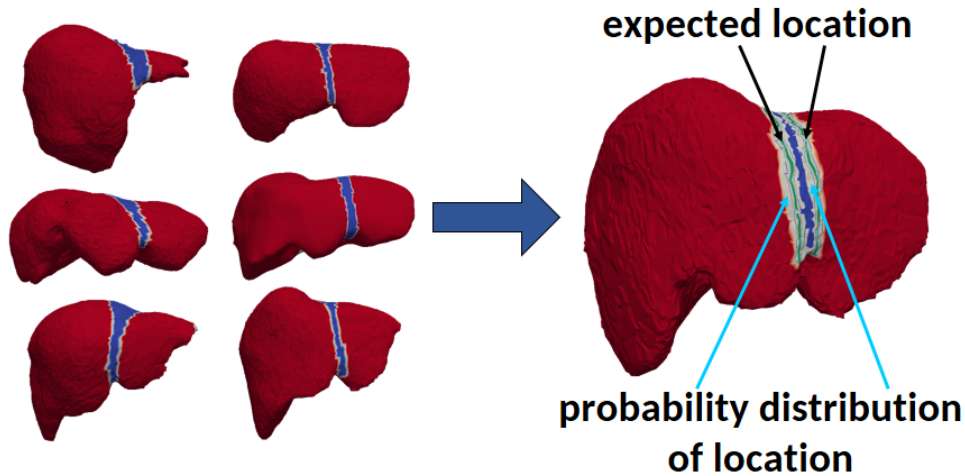


Figure 4.2: Statistical atlas construction for ligaments location. A set of models (left part of image) with segmented ligaments (white strips between red and blue) are registered on a mean liver model (right part of image). A registered statistical atlas has expected ligament location (green strips) and a range of probable locations (white region).

Consequently, the first step of the construction procedure is to construct an average shape (Fig. 4.3). Here, we use a shape that is computed as a Fréchet mean for the database shapes. Then, we register every liver shape from the database on the mean shape and extract segmented data. This procedure allows us to have all segmented ligaments (and other data) on the single surface.

Since ligament positions were segmented as liver mesh edges, their extraction looks such as a set of piecewise linear curves. To get statistics, we need to compute the mean and variance

for the available curves (Fig. 4.3). In this case, the general idea is to deal with all curves in a parametric form. There, the curve shape is presented as a function of an independent parameter $l \in [0, l_{max}]$: $x = x(l), y = y(l), z = z(l)$ in a three-dimensional space. So, the next step is to generate a discretization for the l and to compute statistics for the points obtained from the set of discrete values l_i . Unfortunately, we do not have any analytical parametrization of the curves. Therefore we decided to intersect the curves with a set of planes. Depending on the segmented ligaments, the planes have different spacial configuration. The statistics are then computed for the points of intersections in each of these planes.

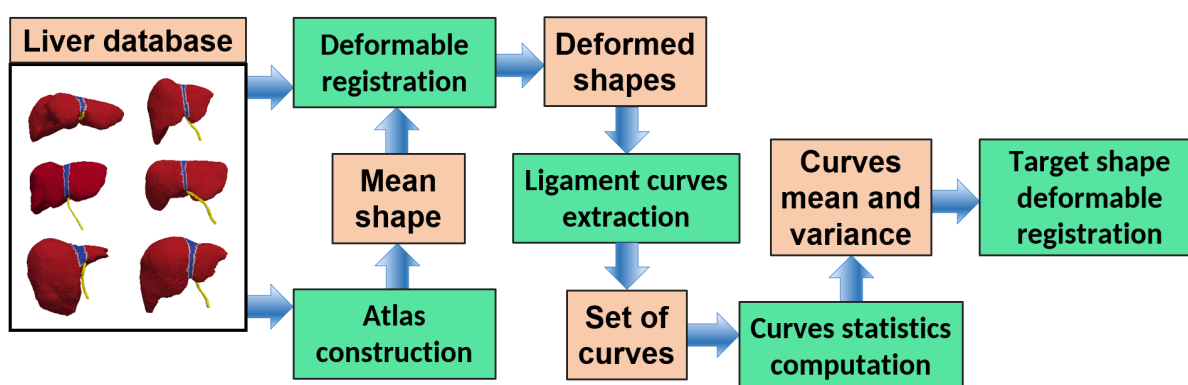


Figure 4.3: Main steps of the atlas construction process and its registration on the target model. Green rectangles depict actions, while red ones describe correspondent data.

Currently, for simplicity, we compute mean and variance in Euclidean space, but generally it is also possible to compute statistics in geodesic space related to the liver mean shape curve, which might be more accurate. Since the Euclidean statistical result might not be on the shape surface, the results are projected on it. The projected point is selected as a point on liver mesh with a minimal distance to the given point.

The final step is statistical data registration on the target shape. To attach the statistical data to the mean shape, a barycentric interpolation is generated. This mapping describes liver mesh positions and interpolation values to obtain the positions of the curve points. After that, the mean shape is non-rigidly registered on the target shape, and barycentric interpolation is used to obtain statistics for the target.

4.4 Ligaments properties initialization

Apart from the positions, the model of BCs is also dependent on mechanical parameters. Of course, as initial values, it is possible to put some casual numbers, but a more natural way is to rely on the available biomechanical data. So, we propose to take the stress-strain curves from [246], measured for porcine tissues, and use them to obtain the initial approximation. The general workflow is presented in Fig. 4.4.

Our idea contains two steps. During the first step, we select a referent hyperelastic model with a strain energy density function Ψ and identify its parameters C_{mat} , based on least squares

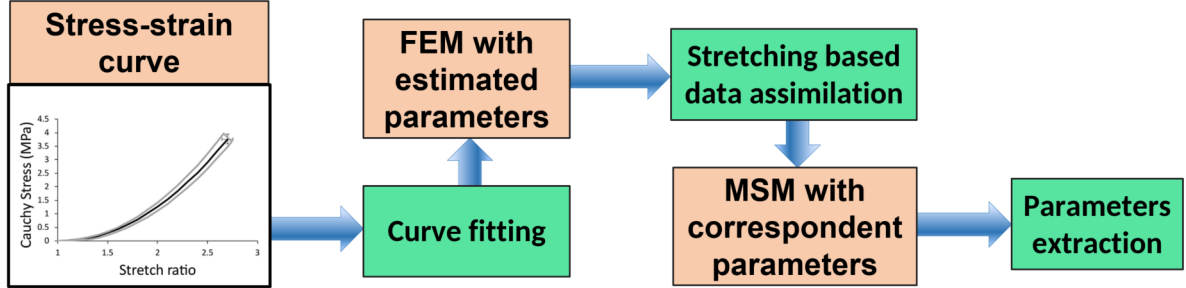


Figure 4.4: Main steps of the elastic parameters initialization process for ligaments. Green rectangles depict actions, while red ones describe correspondent data. Ligament constitutive law taken from [246].

curve fitting approach. The fitting process finds hyperelastic material parameters in such a way that the model behavior matches the best way a series of data points extracted from stress-strain measurements. Or, in other words, we are trying to minimize the difference \mathbf{e} :

$$\mathbf{e} = \sum_{j=1}^N [\tau_j - \tau(\lambda_j, \Psi, C_{mat})]^2 \longrightarrow \min \quad (4.9)$$

where τ_j and λ_j are stress and stretch values obtained from the measured data, $\tau(\cdot)$ is a result of a simulation. It computes a uniform stress for a particular strain based on a selected hyperelastic material with the strain energy density function Ψ and given parameters C_{mat} . There exist many algorithms that perform such optimization. One possible way is to use the Levenberg-Marquardt approach [116], available in the FEBio software [132].

The second step is to determine the parameters for the mass-spring system the way that it will behave such as the given hyperelastic FEM. Unfortunately, there is no mathematical expression to compute parameters under the required constraints. However, we can use the Kalman filtering approach to approximate the spring parameters. For data assimilation process, we took a brick constructed from hyperelastic elements and a set of springs. We then performed stretching deformation for data assimilation, comparing only bounds of both models, like it was done in Section 2.3.4. The filter estimates spring parameters in a way to minimize the difference for the compared positions. The estimated parameters are then considered as an approximation of the material properties and are taken for the correction procedure.

4.5 Preconditioning for data assimilation procedure

The correction procedure for BCs, given in Fig. 4.1, is performed using the Kalman filtering approach, as it was described in the Chapter 3. As far as new intraoperative data becomes available, the unknown parameters are updated, based on a prediction-correction scheme. Consequently we have their best estimation with respect to the available observations and the computation complexity of the procedure.

However, in the context of biomechanical modeling, the prediction step can be time consuming, since physically based simulation has to be executed for every sample point of the filter. The ROUKF or ETKF allow us to reduce the size of the unknown state. This results in fewer simulations for every prediction step. But anyway, if we estimate a huge number of parameters, the prediction will be still time consuming. To accelerate the simulation further, we propose to augment it with a technique called *preconditioning*.

4.5.1 Conjugate gradients and system preconditioning

During simulation process, an object deformation is computed to achieve an equilibrium between external impact and internal stress. The correspondent mechanical problem is commonly solved iteratively with an optimization method until system convergence. Regardless the type of optimization, at each iteration a linear system is constructed, and by solving it we obtain the closer result to the final solution.

In case of the Euler solver, which is used in our work, the linear system is described by equation 2.16 from Section 2.1.4.2. It can be equivalently written in a general form. So we have the system with an unknown vector \mathbf{y} , a transformation or system matrix \mathbf{A} , and a vector of free or known values \mathbf{d} :

$$\mathbf{A}\mathbf{y} = \mathbf{d} \quad (4.10)$$

The left part of this equation contains the stiffness matrix for the whole mechanical system. So, if the body mesh is quite dense, or we have a lot of interacting objects, the matrix is large. But to solve the system, a transformation matrix has to be inverted, which might be time consuming. Therefore there is a need in a way to solve the considered system fast and effectively.

In this case, we can stick to Conjugate gradient method (CG) [77] to solve it. In theory of elasticity, stiffness and mass matrices are symmetric and positive definite, so it is likely to be the fastest method to find a solution for the constructed equations. In CG, we iteratively get closer to the exact solution \mathbf{y} by minimizing the difference, also called *error*, between the correct solution and the current result $\mathbf{y}_{(k)}$: $\mathbf{e}_{(k)} = \mathbf{y}_{(k)} - \mathbf{y}$. The error for the iteration i depends on the initial error in accordance with the inequality:

$$\|\mathbf{e}_{(i)}\|_{\mathbf{A}} \leq 2 \left[\left(\frac{\sqrt{\kappa} + 1}{\sqrt{\kappa} - 1} \right)^i + \left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^i \right]^{-1} \|\mathbf{e}_{(0)}\|_{\mathbf{A}} \Leftrightarrow \delta \|\mathbf{e}_{(0)}\|_{\mathbf{A}} \quad (4.11)$$

where $\|\mathbf{e}_{(0)}\|_{\mathbf{A}}$ is the initial error and κ is a *condition number* of the matrix \mathbf{A} . The condition number for a matrix shows the relationship between its maximal and minimal eigenvalues. The maximum number of required iterations can also be expressed through the matrix condition number:

$$i_{max} \leq \left\lceil \frac{1}{2} \sqrt{\kappa} \ln \left(\frac{2}{\delta} \right) \right\rceil \quad (4.12)$$

More details about the Conjugate gradient method analysis can be found in [216, 217].

Anyway, the important moment we want to precise here is that the CG convergence rate and, thus, the whole performance is very dependent on the condition number of the system matrix.

So, it would be useful to find a way to decrease its value. One existing possibility for this is called preconditioning.

The first preconditioner to solve a linear system was proposed in [223]. There, the authors attempt to approximate the inverse of the symmetric matrix \mathbf{A} by a circulant matrix \mathbf{C} . If the matrix \mathbf{A} has a lot of small values on the most outer diagonals, the multiplication $\mathbf{C}^{-1}\mathbf{A}$ is very close to the unique matrix [240]. Thus, the essence of the preconditioning approach is to solve the system

$$\mathbf{C}^{-1}\mathbf{A}\mathbf{y} = \mathbf{C}^{-1}\mathbf{d} \quad (4.13)$$

instead of equation 4.10, where the condition number of the multiplication follows the inequality $\kappa(\mathbf{C}^{-1}\mathbf{A}) \ll \kappa(\mathbf{A})$. In fact, this multiplication performs system transformation to a more regular space, which allows CG converging in a fewer iterations. However, there are a lot of restrictions that have to be applied for the matrix \mathbf{C} . For example, the multiplication does not have to add new solutions to the system or eliminate existing solutions from it. In the real-time context, there is one more limitation to this technique, the computational overhead does not have to exceed the amount of time saved by increasing the CG convergence rate.

4.5.2 Preconditioned data assimilation process

For real-time deformable object simulations, to find the preconditioning matrix, several approaches have been proposed, starting from taking the diagonal of the matrix and up to approximation of the matrix inversion. Among others, an interesting idea was proposed in [34]. The authors notice that during the simulation process between any two iterations the assembled system matrix \mathbf{A} undergoes small perturbations. Therefore they take the inversion of the first iteration matrix, which they obtained using Cholesky factorization, and use it for a certain number of following simulation steps. Their results show the performance improvement up to several times.

Therefore, to accelerate the assimilation process further, we decided to adopt the solution proposed in [34]. The main idea in our work is to compute the matrix inversion for the first

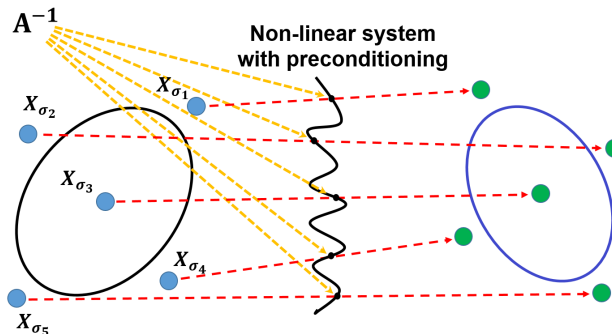


Figure 4.5: Preconditioning for the transformation of uncertainty or the ensemble members transformation. The idea is to compute an inverse matrix for one of sigma points or an ensemble member and use it as a preconditioning in transformation for all other points.

sigma point or ensemble member. And then we could use it to compute simulations for all other sigma points (ensemble members) at a given iteration, as Fig. 4.5 depicts.

The nature of data assimilation process does not differ much from the general simulation. Indeed, the difference between simulation for all such sigma points is small, since every sigma point is obtained by performing some perturbations from original values. This perturbation affects only a subset of the process state and parameters, and, thus, the preconditioning transformation gives almost a unique matrix. So, despite the fact that we use stochastic approach, the deformation difference between simulation iterations remains small. Moreover, it is still possible to use the preconditioning in a similar way as it was used in [34].

The modified algorithm for the reduced-order unscented Kalman filter is presented in Alg. 4 and for the ensemble transform Kalman filter in Alg. 5.

Algorithm 4: Reduced-order unscented Kalman filter with preconditioning

- 1: **Initialize filter data()** - initialize data like in the general case
 - 2: $\mathbf{C} = \mathbf{A}_{(\mathcal{T})}^{-1}$ - compute initial preconditioning for finite elements
 - 3: **set** $\mathcal{T} = \mathcal{T}(\mathbf{C}, \mathbf{x}_1, \mathbf{b}_1^{(c)})$ - preconditioned finite element model
 - 4: **for** each simulation step i **do**
 - 5: **Compute prediction phase:**
 - 6: **get** $\mathbf{b}_{i+1}^{(c)}$ - control features
 - 7: $\mathbf{x}_i^{\sigma_k} = \mathbf{x}_i + \mathbf{L}_i \sqrt{\mathbf{U}_i^{-1} I^{(k)}}$ - generate sigma points
 - 8: $\mathbf{C}_{i+1} = \mathbf{A}_{(\mathcal{T})_{i+1}}^{-1}$ - recompute preconditioner if needed
 - 9: **for** each sigma point k **do**
 - 10: $\tilde{\mathbf{x}}_{i+1}^{\sigma_k} = \mathcal{T}(\mathbf{C}_{i+1}, \mathbf{x}_i^{\sigma_k}, \mathbf{b}_{i+1}^{(c)})$ - get result from deformation step
 - 11: **end for**
 - 12: $\tilde{\mathbf{x}}_{i+1} = \mathcal{E}(\tilde{\mathbf{x}}_{i+1}^{\sigma_*})$ - compute predicted state as mean of sigma points
 - 13: $\mathbf{L}_{i+1} = \tilde{\mathbf{x}}_{i+1}^{\sigma_*} I^{(k)T}$ - updated transformation matrix to reduced space
 - 14: $\tilde{\mathbf{P}}_{i+1} = \mathbf{L}_{i+1} (I^{(k)} I^{(k)T})^{-1} \mathbf{L}_{i+1}^T + \mathbf{Q}$ - compute predicted covariance
 - 15: **Compute correction phase()** - the same as in the general case
 - 16: **end for**
-

We also want to point out here that, despite the fact we modify the linear system, the CG still converges to the similar final solution. Thus the preconditioning does not influence significantly on the process of parameters estimation. However, since in case of BCs the estimation solution is not unique, we finally might converge to a completely different result.

4.5.3 Experiments with preconditioning

Now, let us perform an experiment to see how the preconditioning improves the estimation process.

For this experiment, we took the similar setup like for boundary conditions estimation (see Fig. 3.15 left), that is the cylinder, which is fixed from one side and attached by springs from another. However, the preconditioning shows significant improvement when we use it for many

Algorithm 5: Ensemble transform Kalman filter with preconditioning

-
- 1: **Initialize filter data()** - initialize data like in the general case
 - 2: $\mathbf{C} = \mathbf{A}_{(\mathcal{T})}^{-1}$ - compute initial preconditioning for finite elements
 - 3: **set** $\mathcal{T} = \mathcal{T}(\mathbf{C}, \mathbf{x}_1, \mathbf{b}_1^{(c)})$ - preconditioned finite element model
 - 4: **for** each simulation step i **do**
 - 5: **Compute prediction phase:**
 - 6: **get** $\mathbf{b}_{i+1}^{(c)}$ - control features
 - 7: $\mathbf{C}_{i+1} = \mathbf{A}_{(\mathcal{T})i+1}^{-1}$ - recompute preconditioner if needed
 - 8: **for** each ensemble member k **do**
 - 9: $\tilde{\mathbf{x}}_{i+1}^{en_k} = \mathcal{T}(\mathbf{C}_{i+1}, \mathbf{x}_i^{en_k}, \mathbf{b}_{i+1}^{(c)})$ - get result from deformation step
 - 10: **end for**
 - 11: $\tilde{\mathbf{x}}_{i+1} = \frac{1}{N_{en}} \sum_{k=1}^{N_{en}} \tilde{\mathbf{x}}_{i+1}^{en_k}$ - compute prediction mean
 - 12: **for** each ensemble member k **do**
 - 13: $\tilde{\mathbf{x}}_{i+1}^{diff_k} = \tilde{\mathbf{x}}_{i+1}^{en_k} - \tilde{\mathbf{x}}_{i+1} + \mathbf{v}_{i+1}$ - compute perturbation with inflation
 - 14: **end for**
 - 15: $\tilde{\mathbf{P}}_{i+1} = \frac{1}{N_{en}-1} \tilde{\mathbf{X}}_{i+1}^{diff*} \left(\tilde{\mathbf{X}}_{i+1}^{diff*} \right)^T$ - compute predicted covariance of ensemble members
 - 16: **Compute correction phase()** - the same as in the general case
 - 17: **end for**
-

identical simulations. Therefore we considered a scenario where springs do not share a common stiffness, and these parameters were estimated separately. So we had 3 springs, each of which had 3 polynomial parameters, 9 values in common. The conditions of the experiment are described in the following list:

- **Idea:** spring stiffness estimation with preconditioning
- **Domain geometry:** cylinder, 3 cm diameter, 18 cm length
- **Domain discretization: Groundtruth:** 11125 nodes and nearly 52000 finite elements, **Estimation:** 915 nodes and nearly 3000 finite elements
- **Material properties:** StVK material with Young's modulus equals to 8 kPa and Poisson's ratio to 0.49
- **Boundary conditions:** cylinder is fixed from one side, 3 cubic springs with 3 cm length, all coefficients estimated separately, are attached to the other side. Groundtruth coefficients for every spring are [4, 7, 20] N/m
- **Applied impact:** periodic force in indicated direction is applied.
- **Observations:** 15 markers uniformly distributed on a part of the frontal surface (see Fig. 3.15)
- **Estimation method:** data assimilation using ETKF with 6 ensemble members and ROUKF with and without preconditioning
- **Initial approximation:** coefficients equal to [10, 10, 10] N/m and standard deviation equal to [4, 4, 4] N/m

- **Validation points:** 915 nodes of the coarse mesh

Fig. 4.6 shows the average difference for validation points computed for preconditioned and simple simulations. According to the obtained results, the estimation with preconditioned simulation is slightly worse than the original one. But after 1600 iterations, where we performed initial parameter estimation, the difference between simulation results does not exceed 0.5 mm.

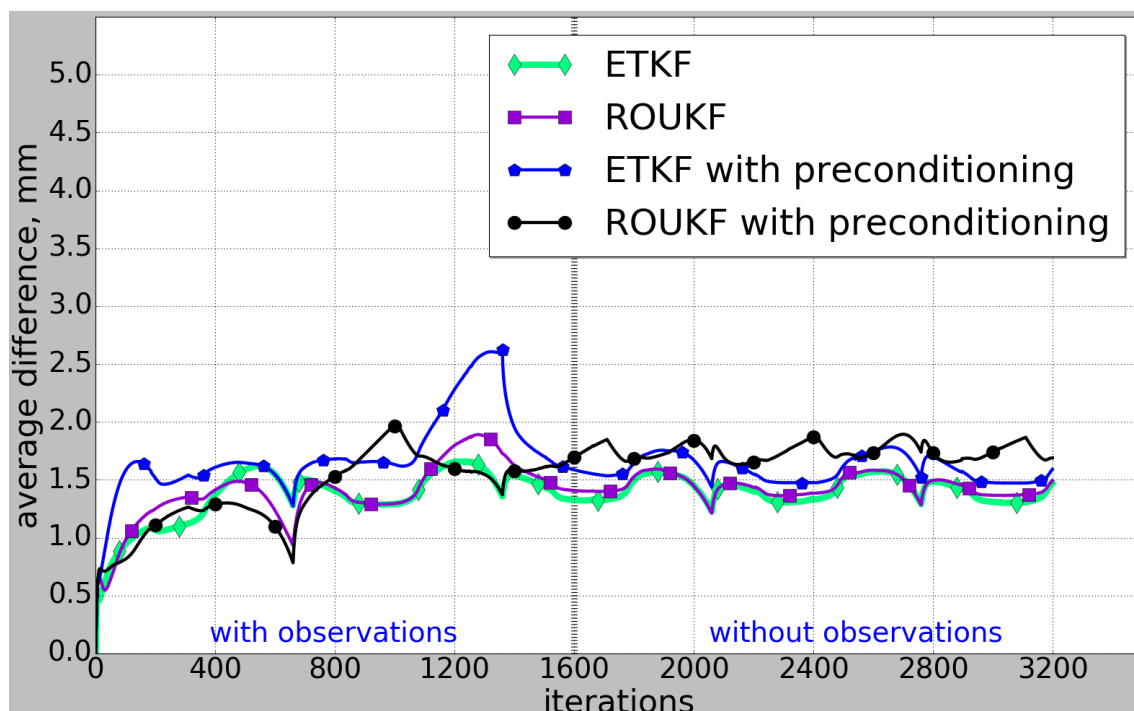


Figure 4.6: The average difference for validation points between groundtruth and simulation results. The results are computed for the ETKF and ROUKF with and without preconditioning. The mesh used to generate the groundtruth is denser than the one used to simulate the object deformations.

Table 4.1 depicts the relative error and computation time for the all performed simulations. The results show that preconditioned estimation is less precise with up to 3 % loss in accuracy. On the other side, it decreases the computation time on approximately 25 %. As usual, the ROUKF shows slightly better results, which makes it preferable for using in the parameter estimation process.

The obtained results show that, regardless expectations, the preconditioning does not provide a very significant computational improvement for our system. However, it is generally known that it works very efficiently for large systems, where it helps to reduce the complexity of a matrix inversion. Because of real-time constraints, we use quite coarse meshes; therefore, the advantage of this technique might be negligible. Anyway, it still provides us with some improvement related to the computation process.

| Method | ETKF | ROUKF | ETKF with preconditioning | ROUKF with preconditioning |
|---------------------------------------|--------|--------|---------------------------|----------------------------|
| Average error wrt max. deformation, % | 2.25 | 2.56 | 3.55 | 2.69 |
| Maximal error wrt max. deformation, % | 7.84 | 8.28 | 8.24 | 11.48 |
| Computation time for one time step, s | 0.2562 | 0.2301 | 0.1948 | 0.1703 |

Table 4.1: Springs coefficients estimation for the case when the mesh used to generate the groundtruth is denser than the one used to simulate the object deformations. Estimation results relative comparison and computation time for the ETKF and ROUKF with and without preconditioning.

4.6 Constructed process for boundary conditions estimation

The main idea of the work is to estimate boundary conditions around the liver for a more precise simulation, but the properties of surrounding organs, particularly ligaments, are not well known. However, during the surgical operation, the position of the liver is caught by the camera or some other modality. This requires to have a combined model of the liver and its attachments. Only in that condition can we estimate BCs, while observing the liver.

Since the aim of the work is to provide a predictive simulation in the context of augmented reality, it implies additional constraints for the estimation process. First of all, real-time simulation is a crucial element in augmented surgery to provide a model compatibility with high quality visual rendering. A typical frame rate for visual display is in the range of 20-60 Hz. Thus it is not possible to simulate all the biomechanical complexity of the liver. We have to simplify the models to optimize the computational efficiency.

The ability to handle soft tissue cutting and suturing has also a primary importance. The impact of such operations in terms of tissue modeling is considerable since it implies changing tissue topology over time. The cost of such a topological change depends largely on the chosen geometric representation, but also on the numerical method that is adopted to compute tissue deformation.

Finally, we are estimating properties of tissues that are not directly visible. Imaging devices are installed only in operable area. Surgeons usually cut all visible ligaments to be able to get access to the liver parenchyma, and therefore there are no attachments. Consequently, we have to extrapolate the observed information in order to perform some estimations. Also, during manipulations, the camera records different parts of the liver, so the areas of interest can be visible partially and discontinuously.

To perform the numerical simulation for the liver, we rely on the FEM method based on Saint Venant-Kirchhoff material. The boundary conditions around the liver model are presented as a set of nonlinear springs, attached to the liver and some fixed points in space. It is impor-

tant to point out that springs do not simulate the behavior of real boundaries, but only their contributions to the elastic properties of the particular areas on the liver surface.

Generally, the whole liver surface has to be covered with springs, but fortunately the statistical atlas will provide us with domains, which are more likely to be attached. Thus it is sufficient to generate springs only in these specific areas. Since there is no specific description for liver ligaments, the best option is to perform initial parameters approximation based on stress-strain curves available in literature [246]. Also, not knowing the exact information about the ligaments length for a specific case, it is possible to take an average of the measurements [118].

The observations that are required to correct the model prediction and estimate the unknown parameters are typically obtained during surgery. As we noted, it is difficult to observe the ligaments; consequently, we mainly rely on features of the liver surface. These features follow the liver deformation behavior and can be tracked by the medical equipment during the surgical procedure. So it is quite natural to consider the liver deformation modeling as a dynamic process that can be managed by the Kalman filtering approach. The stiffness values of boundary springs are presented as unknowns that have to be estimated and tracked markers as observations. By performing the data assimilation process, we will find the more accurate estimation of the boundary conditions.

One essential point here is a combination of the process modeling with the filtering loop. It is presented in Fig. 4.7. We propose to perform it like in the description below. During the

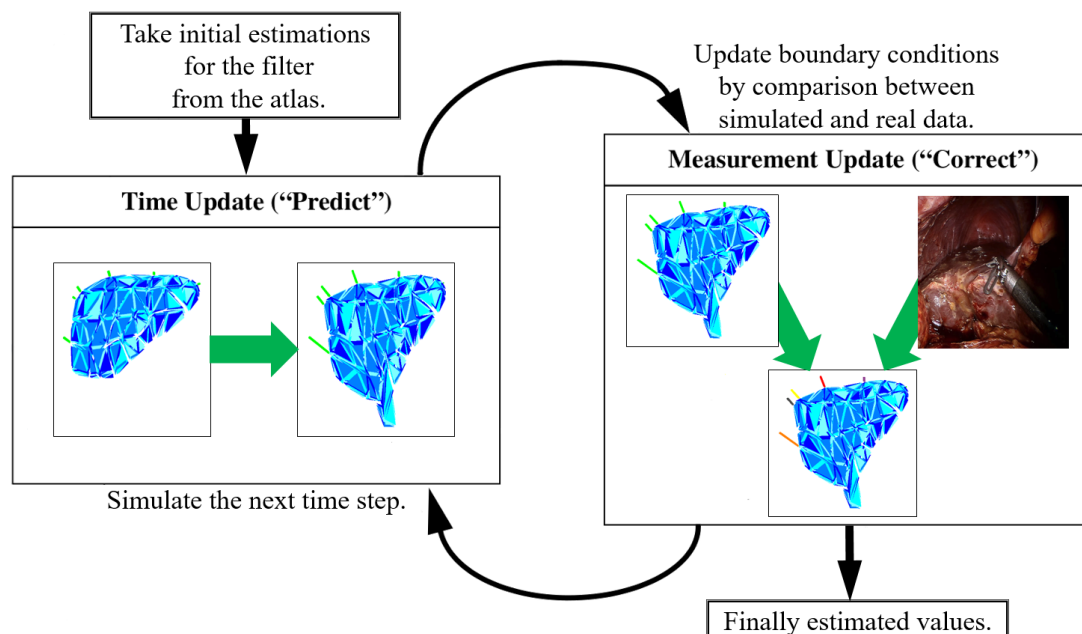


Figure 4.7: Stochastic process workflow for estimation of boundary conditions. In the prediction step, the deformation related to next video frame is modeled based on given springs stiffness estimation. In the correction step, the values are corrected based on difference between simulated observations and real data.

prediction step, the deformation related to the next video frame is simulated using the given

estimation of the spring parameters. And the transformation of uncertainty allows us to use the nonlinear models in the simulation. During the correction step, the simulated features are compared with the data obtained from real recordings. The correction is performed based on the probability distributions comparison. Therefore the system takes into account the fact that extracted data can be noisy. The process repeatedly continues until the end of the procedure. This way, we perform estimation based on all available information related to the surgery. However, if, for any reason, the observations are very noisy or missing, we simulate liver deformation based on currently estimated parameters.

The preconditioning for sigma points will help us to reduce the computational time for the prediction step and, therefore, increase estimation performance. However, we still do not process data in real time. One possible solution is to perform a delayed parameter estimation in a separate system process. The simulation improvement will go slower, but, after certain period of time, the model will be more accurate. Also, in case of force feedback absence, to perform the deformation of the liver and to compare the simulation with real data it is possible to extract two sets of points from medical images. The set of control points is responsible for forming the model deformation, and the set of observation points is used to correct estimation of unknowns. The authors in [180] describe this in more details.



ESTIMATION EXPERIMENTS WITH HUMAN LIVERS



Contents

| | |
|---|------------|
| 5.1 Estimation based on synthetic observations | 103 |
| 5.2 Atlas construction results | 107 |
| 5.3 Experiments with human liver | 111 |
| 5.4 Discussion | 117 |

In this last chapter, we want to present the experimental results that depict how the proposed ideas specify the ligaments positions and improve the simulation accuracy. Initially, we will consider a simplified scenario with observations generated synthetically. Then, we will focus on the experiments with data obtained from real human livers and try to cover the complex scenarios. We will show the results related to the statistical atlas and then stochastic estimation process based on the reduced-order unscented Kalman filter.

5.1 Estimation based on synthetic observations

The idea of this experiment is to investigate how the estimation of BCs using the Kalman filtering approach improves the simulation accuracy. The experiment is based on observations generated synthetically, for which the liver was taken from a segmented CT scan of a patient. Here, we decided to focus only on the estimation process and did not to take into account the contribution made by the statistical atlas. Therefore we processed a model where the ligaments locations were defined by an expert.

For simplicity, we decided to consider only the falciform ligament. This ligament is close to a front surface of the liver, and therefore, compared with other attachments, it has a larger influence on the visible part of the organ.

The reference (groundtruth) simulation was computed with the help of FEBio software [132]. This framework allows using various hyperelastic models for the both volumetric liver model and membrane ligament. Fig. 5.1 depicts an overview of the constructed model.

For the liver we generated a tetrahedron model with approximately 14300 nodes and 73000 elements. An Ogden material was used for it with parameters taken from [127] ($\alpha_1 = 0.88$, $\mu_1 = 16.47$ kPa, $G = 7.21$ kPa).

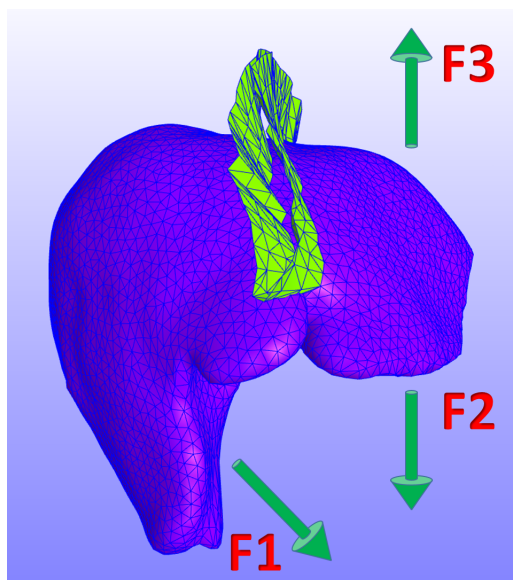


Figure 5.1: Numerical simulation mimicking a manipulation of the liver, from which observations were extracted.

The mesh of the ligament (shell elements) was generated by the segmentation nodes extrusion in the direction of the liver surface normals. Fixed constraints were applied to the furthest nodes from the liver, and bilateral constraints were used to couple the liver and ligament models. We decided to take 3 cm as an extrusion distance. According to [118], the falciform ligament has a width of several centimeters (1.1 - 4.9 cm), and therefore 3 cm is some sort of an average for its size. A Neo-Hookean material was used for the ligaments with the Young's modulus equals to 200 kPa and Poisson's ratio to 0.48.

To generate a representative deformation of a surgical manipulation, we applied nonconstant periodic loads to the right and left lobes of the liver (see Fig. 5.1). To the right lobe, only the downside load was applied since there is a diaphragm upside, and thus the lobe cannot move in that direction. To the left lobe, we applied two sequential forces in opposite directions.

We also attached 14 markers that present some features on the liver observed during the operation procedure (Fig. 5.2 left). We saved their tracks and used the savings as known information in the data assimilation process.

The estimation procedure was implemented using SOFA framework [5]. Since we intend to use the constructed liver model for navigation purposes during surgery, it has to be coarser to process deformations in real time. Therefore for data assimilation process we generated a mesh that contains 2150 nodes and approximately 7500 elements. The ligaments were also extruded along the liver surface normal as two layers of non-linear springs (Fig. 5.2 right).

The springs parameters were inferred as discussed in the Section 4.4. As a reference model, we used a hyperelastic FEM with the Neo-Hookean material. We estimated the material mod-

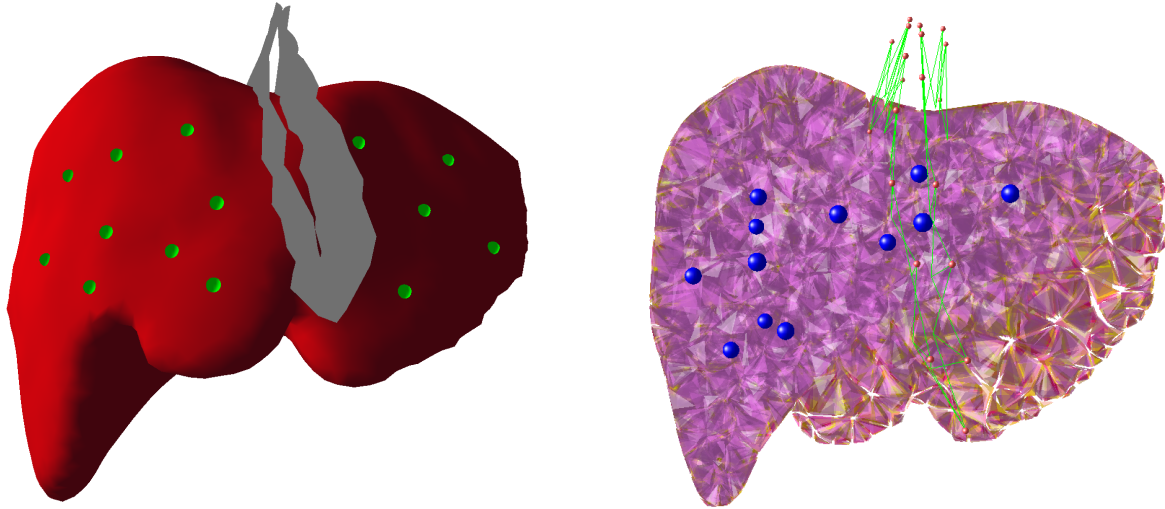


Figure 5.2: Left: Overview of generated observations attached to the liver anterior surface. Right: liver model created for AR with generated validation points. To observe the internal part, the liver model is made transparent.

ulus from a stress-strain curve available for a porcine peritoneum [246] (Fig.7a, x direction). For the process, we used a parameter optimization module available in FEBio [132], and we got 403 kPa as a result. Then, taking the hyperelastic finite element model as the groundtruth, we performed initial approximation of spring parameters using the data assimilation method (ROUKF).

To correct parameters in the patient-specific context, we used a data assimilation approach based on the preconditioned reduced-order unscented Kalman filter. This method is implemented as an open plugin of SOFA framework, and is available online [181]. To optimize further the assimilation process, we split all springs onto a set of regions and shared the same stiffness parameters for each of them. Such parameters processing is just an optimization to speed up the estimation of the BCs, and it can be adjusted depending on the amount of data or available time.

To validate our estimation, we generated 12 internal validation markers that are uniformly distributed inside the liver model (Fig. 5.2 right). To compare the marker positions, we used the naive approach where the falciform ligament is presented as a set of fixed nodes. During the validation deformation, we computed the difference between groundtruth and results for the markers in case of the estimated data and the naive approach. For validation points, we got the average difference presented in Fig. 5.3 and the maximal one presented in Fig 5.4.

The results show that for boundary conditions estimated with our method, the mean error in the deformation is only 2.1 mm (± 0.9 mm), while it is 5.9 mm (± 3.6 mm) when using predefined constant BCs. Similarly, the maximal error is only 3.2 mm (± 1.6 mm), while it is 9.9 mm (± 6.3 mm) for fixed boundaries. Therefore for the largest deformation, the error is reduced by approximately 67 %, as shown in the validation images.

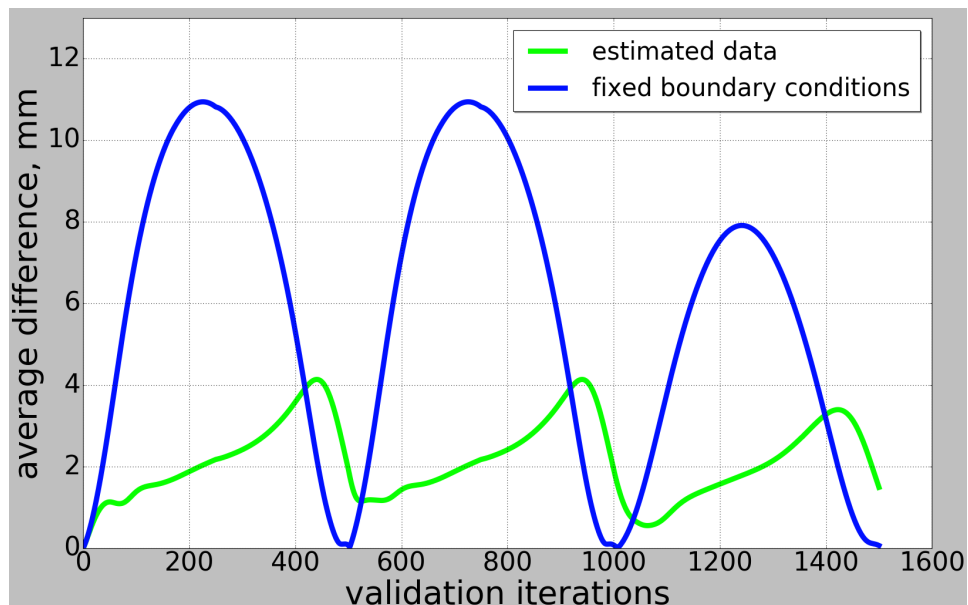


Figure 5.3: Average difference for validation points between groundtruth and simulation results with estimated BCs (green curve) and between groundtruth and simulation with fixed BCs (blue curve). The x-axis shows iterations of the validation process.

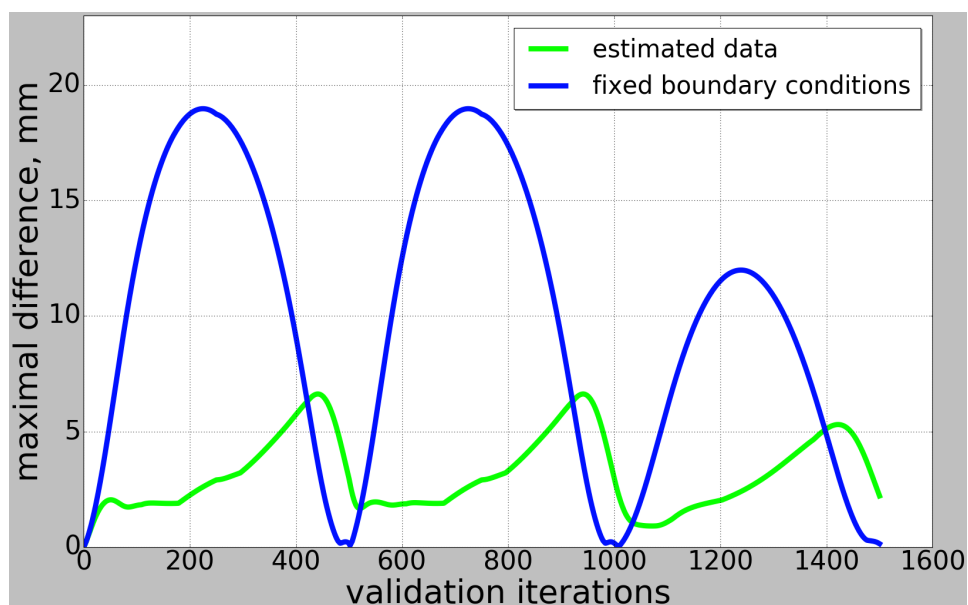


Figure 5.4: Maximal difference for validation points between groundtruth and simulation results with estimated BCs (green curve) and between groundtruth and simulation with fixed BCs (blue curve). The x-axis shows iterations of the validation process.

We also calculated time required by the data assimilation process to perform 1400 iterations. For the preconditioned system, the average time per iteration achieves 1.32 seconds.

As a result, it is worth saying that the estimated boundary conditions essentially improve the simulation accuracy of the biomechanical model. The differences in deformation results presented here are bigger than the ones obtained for the experiments in the Chapter 3. But it can be explained by the fact that we do not only estimate parameters of BCs, but also replace the fixed boundary by a mass-spring system, which simulates the environment more accurately. However, without data assimilation process it is not obvious how the parameters of springs can be specified. Also, the data assimilation process is not real time, but when set as a background process, it allows correcting spring parameters in a reasonable time.

5.2 Atlas construction results

In this section, we will focus on a statistical atlas construction experiment needed to indicate the liver BCs. Since in general, for a particular patient, there is no information about their positions, it makes sense to initialize them from some statistics. The statistical atlas can be constructed from a set of 3D annotated liver shapes. Its computation and target mesh registration workflow, described in the Chapter 4, allows us to determine a probability distribution for the location of ligaments and other anatomical landmarks.

Now, we describe the experiment performed to validate the atlas generation idea. For this, we took 15 liver shapes created using anatomies of real patients. Several ligaments, including falciform, coronary, left and right triangular ones, were segmented by an expert on each of the shapes. We took 14 liver meshes, presented in Fig. 5.5, as a dataset to construct the statistics, and 1 mesh (Fig. 5.6) to validate the accuracy of the atlas. Here, we computed statistics for falciform, coronary, and both triangular ligaments.

Before any processing, there is a need to align liver meshes. For a common reference, we matched their bounding box centers, but the orientation alignment was done manually. The reason for that is the difficulties to find any matching parameters for a correct orientation. The anatomical landmarks also did not work well because of their huge variations. Anyway, in real scenario it could be done by hand since we need to construct this atlas only once, and then we will reuse it for any patient. In addition to that, we did not change the shapes scale, which we hope results in a more accurate registration for ligaments.

The average shape construction and database registration on the average shape was done based on the large deformation diffeomorphic metric mapping method (see Chapter 4), according to the workflow given in Section 4.3.3 (Fig. 4.3). For these procedures we used Deformetrica software [19], available under a free non-commercial license.

The next step consists in statistics creation for the registered curves. However, it might be difficult since there is no available parametrization, based on which they can be compared. As a solution, we decided to intersect the curves with a set of planes.

The falciform ligament has attachments with anterior and superior surfaces of the liver (see Section 2.2.3), the shapes of which are rather round. Therefore we performed intersections with a set of radial planes. Each plane went through the liver center and was parallel to its transverse direction. The intersection points for all the registered curves are presented in Fig. 5.7.

On the other side, the coronary and triangular ligaments lie close to the liver posterior sur-



Figure 5.5: The liver meshes used to construct the statistical atlas for ligaments, anterior view. White strips (between red and blue) represent annotated layers of falciform ligaments. Yellow meshes are round ligaments.

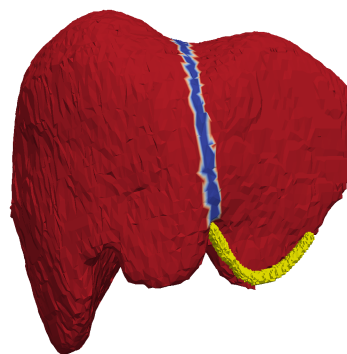


Figure 5.6: The liver mesh selected to validate the statistical atlas, anterior view. The two layers of the falciform ligament are segmented as white strips (between red and blue), and yellow mesh is a round ligament.

face (Section 2.2.3), whose shape is flatter. Therefore we created a set of parallel planes, perpendicular to the liver transverse direction. The intersection results are presented in Fig. 5.8.

The intersection points give us a possibility to compute statistics inside each of the plane. Roughly talking, the statistics has to be computed on the average shape, but we just computed

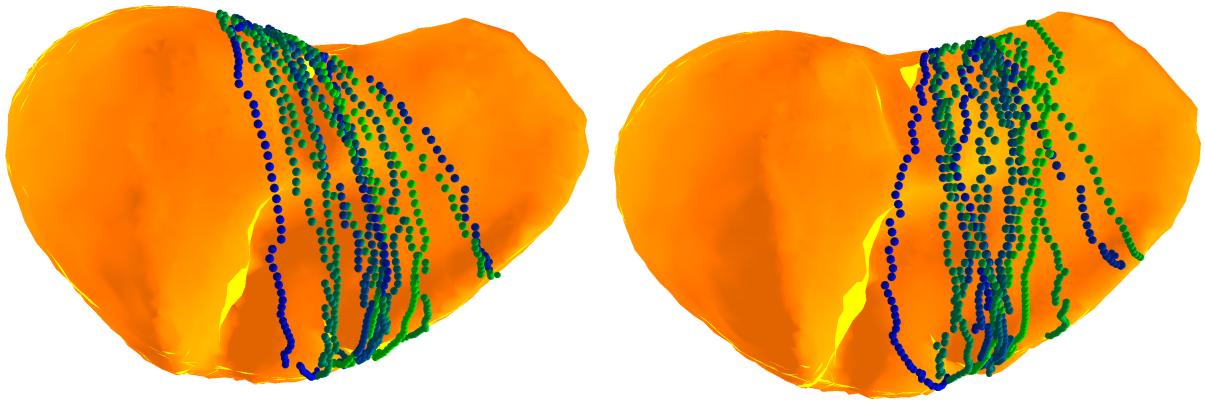


Figure 5.7: The annotated right layers (left image) and left layers (right image) of the falciform ligament taken from liver dataset and registered on the average shape. Each curve is marked as a set of points with a color from green-blue color map.

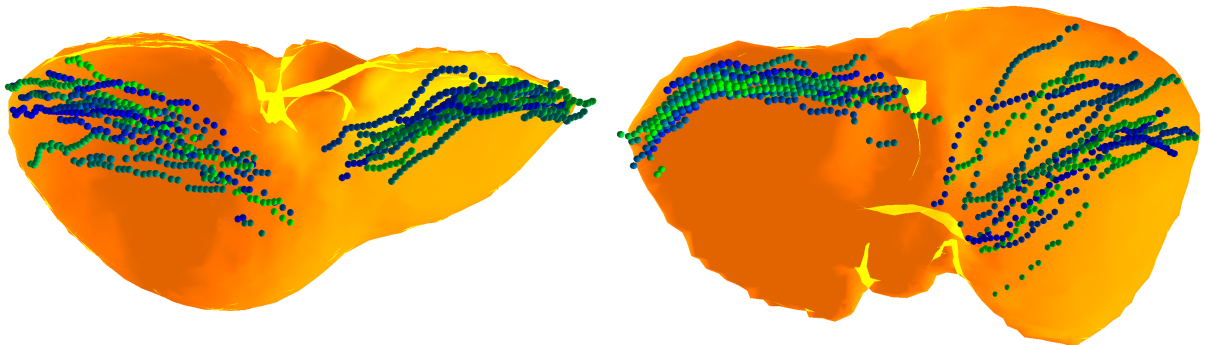


Figure 5.8: The annotated upper layers (left image) and lower layers (right image) of the coronary ligament with left and right triangular ones taken from liver dataset and registered on the average shape. Each curve is marked as a set of points with a color from green-blue color map.

the Euclidean metric in space and then performed its projection on the shape using minimal distances. Additionally, we computed statistics for the all available intersections. So if there were fewer than 14 intersections in a given plane, we computed the mean and standard deviation values anyway.

Finally, the average mesh together with the statistics was registered on a target model based on the same LDDMM approach. To transfer the computed results, we created barycentric coordinates for mean and standard deviation positions in the average shape subspace. After target registration, we calculated them, to restore the coordinates of the registered statistics. For the validation model, the registration results for the faciform ligament are presented in Fig. 5.9 and for the coronary as well as the triangular ligaments in Fig. 5.10.

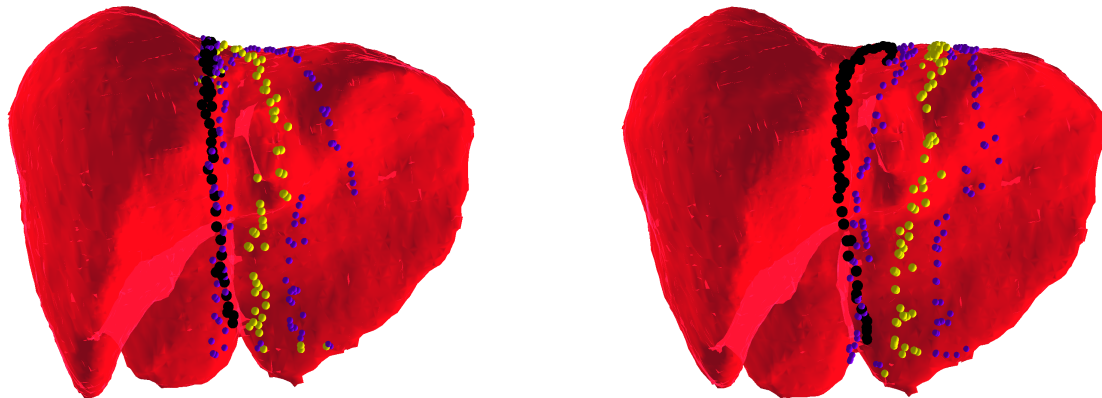


Figure 5.9: Registration results overview for the right (left image) and left (right image) falciform ligament layers on the validation model. The mean (yellow dots) and standard deviation (violet dots) are superposed with manual segmentation (black dots).

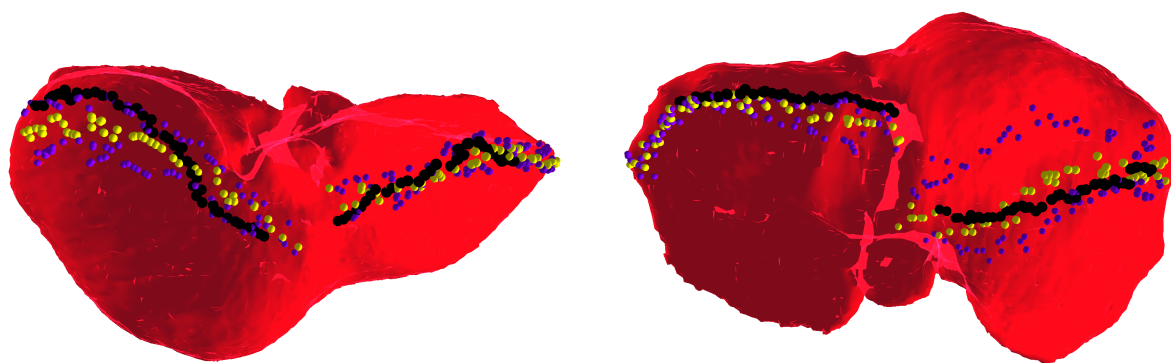


Figure 5.10: Registration results overview for the upper (left image) and lower (right image) coronary layers, together with left and right triangular ligaments, on the validation model. The mean (yellow dots) and standard deviation (violet dots) are superposed with manual segmentation (black dots).

To assess the accuracy, we computed the Target Registration Error (TRE) between registered averages and the curves that were segmented by the expert. For comparison, we used the same plane intersections technique as described above. We counted the difference only for planes where both compared curves have intersections. The results are presented in Table 5.1.

We should note that there is a certain difference between statistical average and segmented curves, particularly for the falciform ligament. This might be caused by several reasons. First of all, there is a huge anatomical deviation both in ligament positions and liver parenchyma shapes. It especially affects the falciform ligament because it lies on a smooth surface far from sharp segments. However, in general, such segments are used as referent elements for the accurate registration. Likewise, we used only 14 livers for the statistics, which cannot provide us with a very confident result. Also, since the initial alignment was manual, it might be inaccurate.

| Ligament | Average difference, mm | Maximal difference, mm |
|---|------------------------|------------------------|
| Falciform right layer | 14.7 | 26.9 |
| Falciform left layer | 23.6 | 36.1 |
| Coronary upper layer from left side and part of left triangular | 2.9 | 9.6 |
| Coronary lower layer from left side and part of left triangular | 5.3 | 12.0 |
| Coronary upper layer from right side and part of right triangular | 11.1 | 16.7 |
| Coronary lower layer from right side and part of right triangular | 6.5 | 12.1 |

Table 5.1: The Target Registration Error between registered statistical average and manual segmentation for the given ligaments on the validation model.

Anyway, according to our experience, the proposed approximation is the best we know for initializing the ligaments positions. For any constructed atlas, there will always be a certain deviation between real positions and statistical results. Consequently we expect we can still compensate it by estimating slightly different stiffness parameters.

5.3 Experiments with human liver

In our final experiment, we would like to demonstrate the results obtained for the liver of the real human. Simulations based on synthetic data allows us to investigate the estimation process under simplified conditions. But only experiments with real data show the whole complexity of the considered problem and depict how the proposed solution attempts to solve it.

For this experiment, a human cadaver was taken (woman, 89 years old, the weight is approximately 80 kg) from the Institute of Normal Anatomy in Strasbourg. The cadavers are stored frozen, so before the experiment a 72 hours defrosting was done. The defrosting process was held under room temperature and standard conditions.

The overview of the operating field is presented in Fig. 5.11. In this setup, we emulated an open surgery procedure.

The various liver manipulations were done by a surgeon using common surgical tools. Several laparoscopic tackers were attached to the liver surface to form visible observations that can be tracked by a camera (Fig. 5.12 left). Their positions were recorded by an Intel RealSense D435 RGB-D camera (Fig. 5.12 right), which was placed over the surgical table (not visible in Fig. 5.11). This camera, in addition to an RGB sensor, has an infrared emitter and two infrared sensors, which allow us to obtain 3D positions for the pixels selected on the RGB image.

For validation, we used 9 internal fiducials, which were inserted in different parts of the liver parenchyma to be more or less uniformly distributed inside the organ. We acquired CT scans

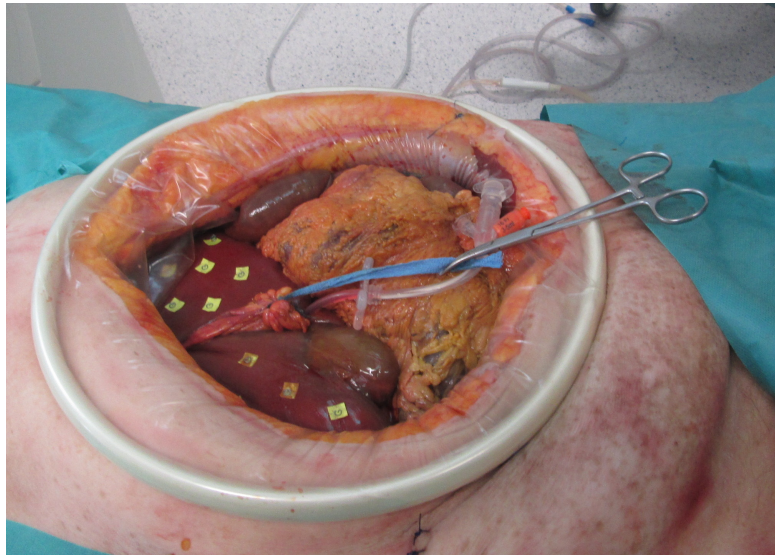


Figure 5.11: Overview of the operating field for the human liver experiment. To track the liver movements, a set of laparoscopic tackers were attached to its surface. To simplify segmentation, we added colored papers to the tackers. The manipulations were recorded with an Intel RealSense D435 RGB-D camera.



Figure 5.12: Left: Example of a laparoscopic tacker, which is generally used to attach artificial materials to the tissues. Right: Overview of an Intel RealSense D435 RGB-D camera. It has a color sensor to obtain an RGB image and an infrared (IR) projector with two IR sensors to reconstruct the scene depth.

during the experiment, including several for the deformed organ, to reconstruct the shape of the liver as well as the markers positions.

After the manipulation experiment, the recordings were post-processed. We constructed the finite element liver model with spring attachments, fulfilled the data assimilation process, and validated our estimations.

The liver shape was reconstructed from a CT scan, holding its undeformed shape. Using CGAL library [190] integrated in SOFA, a 3D mesh was constructed. We generated a mesh

that contains 2350 nodes and approximately 7700 elements. Together with the liver organ, we segmented a set of superficial tackers and internal fiducials, detected by CT. Their relative positions are presented in Fig. 5.13.

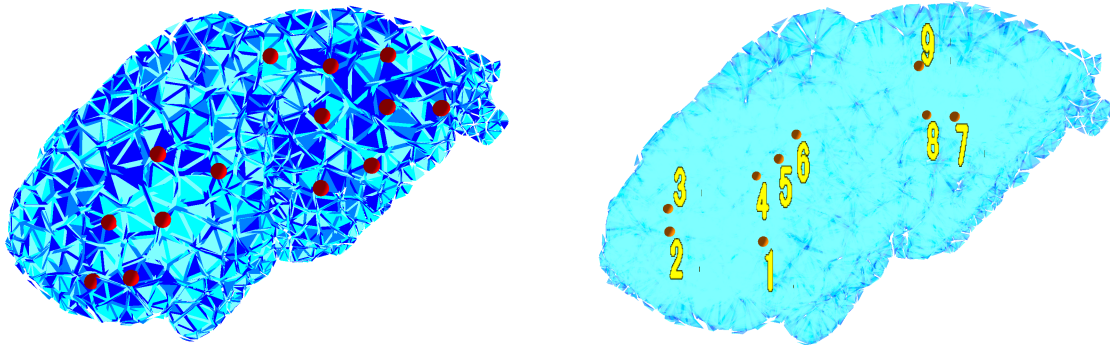


Figure 5.13: Left: overview of the reconstructed laparoscopic tackers on the liver surface. Right: overview of the reconstructed deep fiducials labeled with numbers. To observe the internal parts, the liver model is made transparent.

The statistical atlas, needed to initialize the ligaments positions, was generated from 15 segmented liver shapes (Fig. 5.5 and 5.6), according to the workflow described in Section 5.2. The statistical average was then registered on the human liver mesh. The registration results are presented in Fig. 5.14 and 5.15.

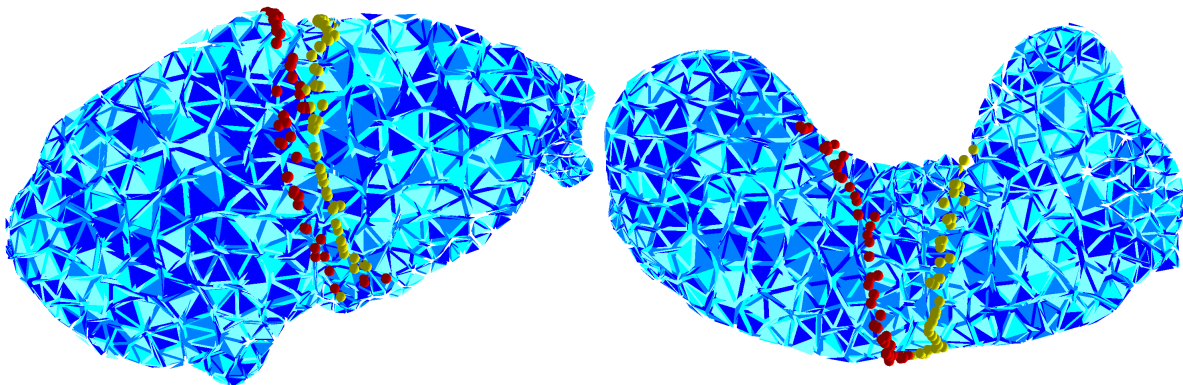


Figure 5.14: Statistical average registration result for the falciform ligament on the human liver shown in superior (left image) and anterior (right image) views. The layers colored differently to clarify their registered positions.

Based on registered point cloud, a mass-spring system representing ligaments was created. We manually annotated a set of points, the shape of which looks like a curve. The points were extruded along the mesh surface normals, and fixed constraints were applied to the furthest nodes. The distance of the extrusion was equal to the average falciform ligament size (approximately 3 cm), noted in [118]. Finally, we generated the springs. The result model with generated mass-spring system is presented in Fig. 5.16.

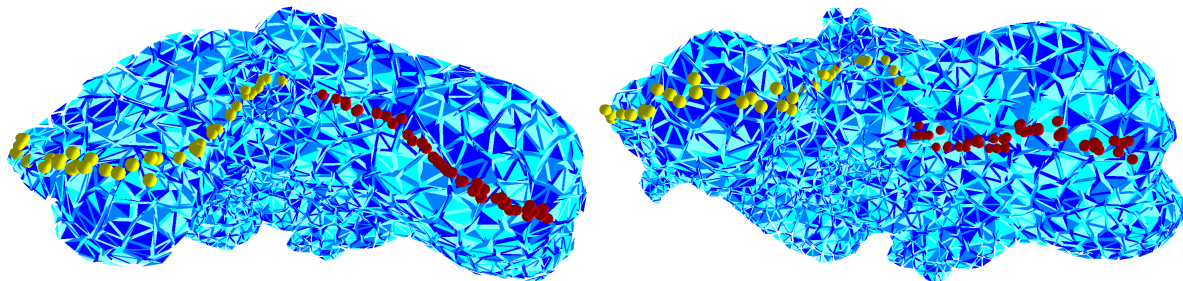


Figure 5.15: Statistical average registration result for the upper (left image) and lower (right image) layers of the coronary ligament and both triangular ligaments on the human liver. The layers colored differently to clarify their registered positions.

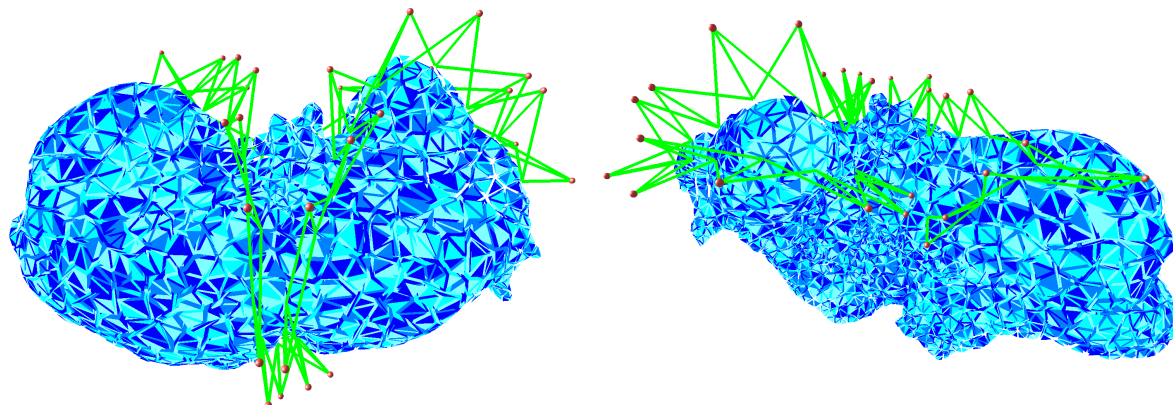


Figure 5.16: Finite element model with the constructed mass-spring system shown in the superior (left image) and posterior (right image) views.

The observations for data assimilation process were taken from liver manipulation recordings. To extract the spatial information for visible points, we had to segment them on image sequence. Unfortunately, the modern approaches are mainly based on convolutional neural networks [229], which require already processed data for training. In our case, we did not have it. Therefore, without having any other appropriate solution, we segmented sets of points manually. After segmentation, to extract markers trajectories, we had to find points correspondences between neighbor frames. Currently, we used an approach based on the minimal distances. The idea is to compute the squared distance matrix for the two sets of points, and then couple them by selecting pairs with the minimal available distance from the uncoupled subset.

To perform the data assimilation step, the constructed model had to be registered on intraoperative view. The registration was based on the superficial tackers. Since the same tackers were recorded by the RGB-D camera, we had their spatial positions for intraoperative data. With two available point clouds, the registration procedure can be performed based on iterative closest point approach.

To simulate the liver behavior, we used a hyperelastic Finite Element Method with StVK ma-

material. Unfortunately, we did not have a possibility to measure the liver Young's modulus, so we took it as 15 kPa, which is an approximate average for the measurements done in the literature (for example, in [168]). However, in general case, it is possible to estimate the modulus using, for example, ultrasound elastography. We also took Poisson's ratio as 0.49 since liver tissues are almost incompressible. The spring parameters were inferred according to the description, given in Section 4.4. As a reference stress-strain curve, we used data from porcine peritoneum [246], (Fig.7a, x direction). We estimated the Young's modulus for the reference Neo-Hookean model using the parameter optimization method available in FEBio [132] (we got 403 kPa). Then, we performed initial approximation of spring parameters using the ROUKF.

The data assimilation was performed with the preconditioned ROUKF. After that, we computed a Target Registration Error to validate the corrected model and process computation time.

For validation, we used another CT scan where the liver was fixed in the deformed state. The deformation was obtained by putting a roller piece under it. With these conditions, the modified liver shape differed from the reconstructed one. The difference between the CT scans is presented in Fig. 5.17. From the second CT we extracted only surface tackers and internal fiducials.

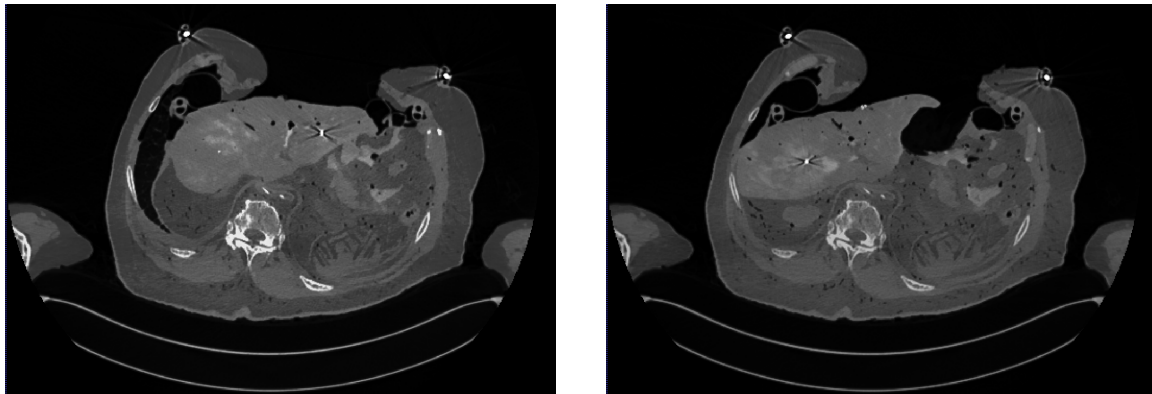


Figure 5.17: Visual comparison of two CT scans for the liver organ in its rest shape (left image) and deformed shape (right image).

We supposed that two CT scans are aligned, so we can directly compare the two sets of points. Also, for comparison, we mapped surface markers from both configurations and compared the position of internal fiducials. Roughly talking, it is not very accurate since the biomechanical model deformation is based on mechanical parameters, which only approximate the behavior of real tissues. At the same time, we just created matches between surfaces regardless the model behavior. Anyway, without having any force feedback in the system, this comparison is the best idea we have and still can give us some general results related to model accuracy.

We considered three scenarios:

1. Only blood vessels, as BCs visible on a CT scan, are modeled. They are placed as a fixed region.
2. The known BCs are modeled including blood vessels and ligaments obtained from the

statistical atlas. The blood vessels are placed as a fixed region, while the parameters of springs are initialized based on the porcine peritoneum stress-strain curve.

3. The known BCs are modeled including blood vessels and ligaments obtained from the statistical atlas. The blood vessels are placed as a fixed region, and parameters of springs are estimated based on the reduced-order unscented Kalman filter with preconditioning.

The Target Registration Error for all three scenarios is presented in Fig. 5.18. Since only 9

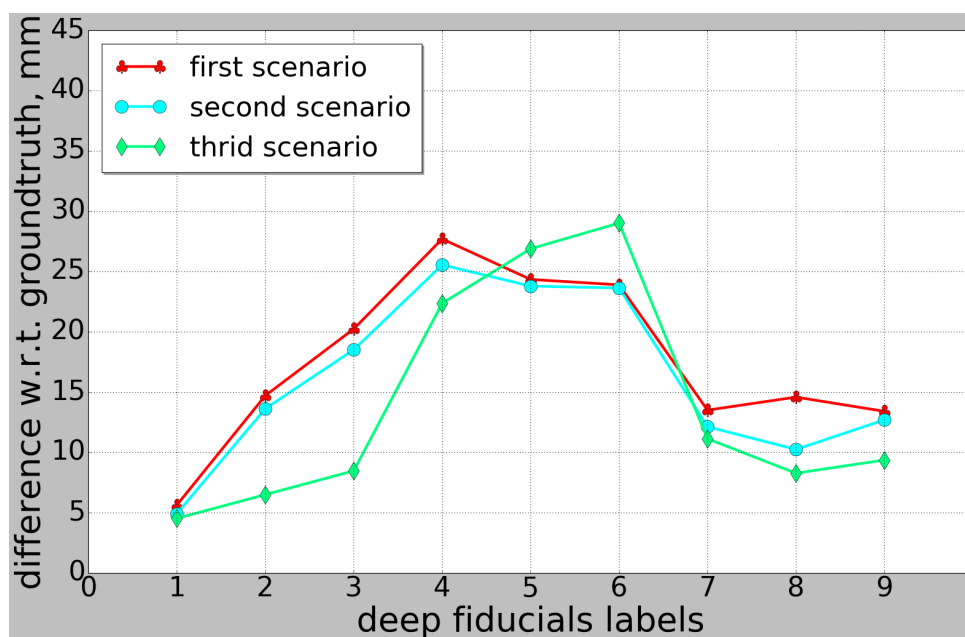


Figure 5.18: TRE for the internal fiducials between the groundtruth and the simulation results in three scenarios. In the first scenario, only blood vessels as fixed BCs are modeled. In the second one, a mass-spring system is added with parameters estimated from pig peritoneum stress-strain curve. In the third one, a mass-spring system is added with spring parameters estimated by the preconditioned ROUKF. The fiducials labels are correspondent to ones in Fig. 5.13 right.

internal markers were inserted, we show the difference separately for each of them. The average difference for the first case is 17.5 mm. For the second case related to the initial approximation, it is 16.1 mm, and for the last one 14 mm. For data assimilation process with preconditioning, the computation time for one iteration takes 1.97 seconds on average.

The results show that for the case with added BCs there is a certain improvement in simulation accuracy. The estimated BCs give further improvement for several deep fiducials, while the positions of others become less accurate. Anyway, the average difference decreases further.

It is worth saying that in this experiment we expected to observe much more accurate simulation for the last two cases. The reasons why we do not have so much improvement might be different. First of all, we map superficial points, therefore, forcing the system to behave according to the observed part of the liver, but not to its biomechanical properties. Then it might also be due to unilateral BCs, which we do not consider at all, but they still have a huge influence

on the simulation process. In addition, we observe too little data with too big noise, which does not allow us to assimilate a lot of information. It is also worth noting that in general case we could track the surgical tool outside the abdominal area with an optical system and, therefore, simulate its influence more accurately. But without having this information, we just used some of the observations to control the deformation, which also appends additional error.

5.4 Discussion

This work is dedicated to the problem of BCs identification for the more accurate liver behavior simulation. The corrected simulation can be used in augmented reality system to reveal the internal parts of the organ, which would help surgeons to perform quicker and safer procedures.

To correct values of the boundary conditions, the nonlinear Kalman filter has been employed. The initial approximation is made based on a statistical atlas and a constitutive law available in the literature. This statistic is not always available in practice. But it is possible to perform the necessary segmentation and other measurements for several samples manually, and the results can already be used as an initial approximation.

To perform correction, the filtering approach uses data obtained from surgical recordings. As observed markers, any trackable points on the liver surface can be used, such as standard or reflective markers or marks burned with a surgical tool. An important issue is to observe and track their motion; otherwise, nothing can be learned about the behavior of the organ.

The boundary conditions are presented as nonlinear springs with unknown parameters. Together with the liver nodes, they form the state estimated by the filter. The mass-spring system is computationally fast and allows handling topological changes without complex mesh restructuring. Of course, it does not simulate the behavior of boundary conditions accurately. However, we are mainly interested in predicting the deformation of the liver organ but not its surrounding tissues. On the other side, we reduce the complexity of the model, while keeping simulation of liver deformations with sufficient accuracy. The results show that by estimating BCs this way, the simulation error decreases significantly.

Both liver and ligaments as well as other human organs and tissues have nonlinear properties. The unscented Kalman filter and its extensions are an interesting choice since they handle the nonlinearity without computing the system derivatives, which is not trivial for the hyper-elastic Finite Element method. The essential challenge, however, is that this transformation requires to perform the simulation for every generated sigma point, which is time consuming. On the other hand, the model has to simulate the liver in real time because surgeons have to observe directly the results of their manipulations. Therefore we focus on the reduced filters, which enable us to decrease essentially the computational cost, while keeping approximately the same level of estimation accuracy.

It is also important to point out that the nonlinear Kalman filter can update the estimation in time. If surgeons perform certain modifications of BCs, such as ligament cutting, it could be corrected by the upcoming data assimilation. The results show that the filtering approach quickly reacts to such changes and is able to estimate the modified parameters.

The idea to precondition the computation of the sigma points can give us some improvement

in computation speed, but a frame rate (20-60 Hz) suited for human vision cannot be achieved yet. Anyway, filters allow obtaining intermediate results, which could be used to improve the estimation for the AR system in a workflow scenario. In this case, a possible solution is to put the filtering approach in a separate background process, which updates the real-time simulation as far as a new estimation is ready. It will perform estimation with a certain delay, but, with the lapse of time, the real-time simulation will get the updated parameters, and this will improve its accuracy.

One of the main advantages for the filtering approach is its uncertainty nature. It incorporates the observation errors of intraoperative images and takes into account that the liver model is inaccurate. Therefore, we could use a simplified but less accurate model to be close to real time and be sure our model is processed properly. Also, together with expected values, the filters give us estimation statistics, which we could use as an additional information for the investigations, such as exploring the dependency of the estimated values on various system parameters.



On the other hand, the filtering approach requires the certain knowledge about system uncertainty, since it has parameters that have to be tuned appropriately. In ideal case, the measurement and model noise have to be known, but in context of our work it depends on so many factors that we cannot estimate them easily. A possible way to avoid this might be to use estimation with adaptive parameters tuning, but there is no well-defined methodology related to this domain.

In addition to that, the Gaussian distribution might not be the best probability distribution to estimate stiffness parameters. However, we have to keep using it since non-Gaussian distributions are computationally expensive and cannot be applied for the real-time systems.



The estimation process is dependent on the initial stiffness value and its covariance. The choice for the initial stiffness is not trivial. It cannot be negative due to unrealistic behavior, and it also has to be small enough since stiff objects are almost solid, and therefore there is little information for data assimilation. The idea is to take the softest values available in the literature, which are reasonable and relate to the available data. Ideally, in this case, it will provide us with the most relevant information.

Looking at the covariance, for small values the process might take too much time to converge, but for big ones sigma points will get too different simulations, which might cause huge deviations and, finally, divergence.

Despite all drawbacks and limitations, we still think the proposed method is the best possible solution to estimate BCs, taking into account the given constraints and general knowledge absence about patient specific description of liver boundaries.



CONCLUSION



In this work we proposed a new method to estimate BCs based on reduced-order unscented Kalman filtering approach. The boundary conditions are presented as springs with unknown parameters. The filter estimates them using patient-specific information obtained during surgery. The initial approximation of BCs is taken from a statistical atlas, constructed from a set of liver models with segmented ligaments, and an experimental constitutive law using the parameters fitting approach.

We believe the proposed solution offers a great potential. The atlas registration allows focusing only on the specific areas of interest and does not take into account other regions. The data assimilation process helps us to correct the initial approximation according to data obtained from a specific patient, which is a huge benefit. Generally, the anatomy of every person has specific variations, and this makes it different from a statistical average. Moreover, filtering process allows us to perform parameters correction in close to real-time context. This is an essential point for the cases when the recorded data has to be processed during the procedure, or there is no additional time for calculations.

Our main area of interest is the simulation of soft tissues in the clinical context. However, the proposed approach could be used in various areas. Finite element method is one of the most popular methods to solve systems of partial derivatives. Preconditioning is a common approach to improve the convergence for linear systems. Kalman filtering is also a general approach for estimation of some unknowns in a dynamic process. Thus, the main restrictions of the proposed approach are the presence of dynamics in a considered event and the existence of observations that track it.

We are trying to estimate the liver boundary conditions, but, commonly, we could estimate various parameters of the constructed model, created for any organ or tissue. The parameters might be either time dependent or not. Also, in our work, we mainly focus on cameras, but, in general case, other medical modalities could be used, such as ultrasound probes, magnetic trackers, or infrared sensors.

In addition to that, we guess such approach has a primary importance for medical robotic systems. Unlike humans, robots process only the data they observe, and, therefore, a possibility to correct constantly the procedure according to the obtained information will definitely increase their positive outcome.

There are possibilities for further improvement of our approach. The main issue is that,

despite the data assimilation procedure is close to real-time performance, it is still not there. So, it would be nice to improve further the computational time. The sigma points can be easily parallelized since they process independent simulations. Moreover, it is possible to perform it on a GPU, which in general is faster. The narrow place is CPU to GPU transformation, but we need to transfer only stochastic state, while all other data related to biomechanical simulation could be kept in the GPU.

In addition, it would be interesting to add a mechanism to estimate unilateral constraints (contacts) using the Kalman filter. Unlike ligaments and other attachments, these constraints are active only when liver contacts surrounding organs and tissues, so for their estimation a special protocol is required. A possible solution might be to select a certain video sequence, where surgeons perform specific manipulations to activate all contacts, and use a Kalman filter to estimate only their characteristics.

Another possibility is the improvement of the filtering approach itself. Data assimilation is composed of two classes. They are statistical, which includes various versions of Kalman filters, and variational, which were obtained from optimal control theory. Each of them has its own advantages and drawbacks. In recent investigations, researchers are trying to combine them to keep advantages from both classes. As a result, new hybrid variational-statistical methods appear. In the domain of biomedical modeling and particularly liver BCs estimation, they might give some advantage for the accuracy over performance rate.

It is also possible to look at how the observations are obtained. Currently a general optical tracking approach is used. The RGB-D camera is a recent development that allows reconstructing three-dimensional positions based on selected image pixels. But in this case, instead of just selecting features, it is possible to segment the visible part of the object surface, reconstruct the point cloud related to it, and then create points matching between frames. This will help to avoid putting any markers on the surface. But the main question, which remains here, is the creation of a reliable matching between point clouds.

Consequently, in the future work, it would be important to focus the research on the indicated remarks.



Summary in french



Introduction

Le foie est un des plus grands organes du corps humain. Il joue un rôle fondamental dans de nombreuses fonctions vitales, notamment la détoxification de divers métabolites, la synthèse des protéines et la production de substances biochimiques nécessaires à la digestion et à la croissance. Par conséquent, les maladies du foie peuvent entraîner de graves problèmes qui se traduisent par diverses anomalies et une réduction de l'espérance de la vie. Il est particulièrement vulnérable au cancer primaire du foie, qui est le cinquième cancer le plus fréquent dans le monde et la troisième cause de décès par cancer. L'organe hépatique souffre également d'autres métastases, causées par des maladies telles que le cancer colorectal.

Avec la transplantation, le traitement typique du cancer pour le foie est la résection chirurgicale. Les chirurgiens enlèvent complètement les tumeurs, tout en préservant autant de tissus sains que possible. Actuellement la tendance principale est d'utiliser des procédures plus sûres avec moins de complications, comme les traitements mini-invasifs. Cependant, pendant ces procédures, il n'y a pas de vue directe de la zone à opérer. C'est pourquoi les chirurgiens utilisent diverses modalités d'imagerie médicales, telles que des caméras laparoscopiques 3D, qui n'offrent qu'un champ de vision limité. Pour améliorer cette situation, une solution possible est d'utiliser un modèle qui simule le comportement de l'organe réel. Adapté à la partie visible, il prédit les positions des parties invisibles et montre les emplacements des tumeurs initialement reconstruites, formant ainsi une réalité augmentée pour l'organe considéré.

Le foie est un organe mou, qui change de forme pendant la chirurgie, mais aussi en raison des mouvements respiratoires et cardiaques. Fournir une vue augmentée du foie pendant la chirurgie nécessite donc de relever plusieurs défis: le modèle 3D de l'anatomie virtuelle doit être recalé sur l'organe réel en utilisant uniquement des données de surface partielles; il doit suivre le mouvement réel des tissus en temps réel; et il doit fournir une estimation de l'emplacement des structures internes avec une bonne précision.

Pour atteindre ces objectifs, différentes solutions ont été proposées. Certaines méthodes proposent d'utiliser des modalités d'imagerie médicales peropératoires comme la tomographie par ordinateur ou l'imagerie par résonance magnétique. D'autres essaient de reconstruire progressivement la région abdominale. Mais les approches principales sont soit des méthodes basées sur l'intensité, qui tentent de fusionner les données avec l'image peropératoire, soit des méthodes basées sur la physique, qui décrivent la déformation de l'organe avec plus ou moins de

précision. Lors d'une chirurgie hépatique, le principal système visuel reste la vision directe (en chirurgie ouverte) ou à travers une caméra (dans les procédures laparoscopiques). Par conséquent, seule une petite partie de la surface du foie est observée, à partir de laquelle un champ de déplacement dense doit être récupéré. Dans ce contexte, une solution basée sur la physique est très efficace pour estimer le déplacement en profondeur à partir du mouvement de la surface, ou plus généralement à partir de données éparses.

Tout problème mécanique comporte plusieurs composantes principales: (1) une approche numérique, telle que la méthode des éléments finis, qui relie la géométrie de la déformation avec la déformation du matériau; (2) une méthode d'optimisation, telle que celle de Newton-Raphson, qui traite les équations d'équilibre entre la contrainte interne et les contraintes appliquées ou les forces externes; (3) une loi constitutive qui décrit les propriétés du matériau en interconnectant la contrainte et la déformation; et (4) un ensemble de conditions aux limites (CAL) qui caractérisent la façon dont la région d'intérêt est connectée au milieu environnant. Actuellement, un certain nombre d'articles liés à ce sujet (y compris ceux énumérés ci-dessus) sont consacrés à la construction d'un modèle numérique pour les simulations de foie en temps réel, nécessaire pour créer le système de réalité augmentée. D'autres décrivent des propriétés matérielles hyperélastiques du foie. Cependant, beaucoup moins de travaux ont porté sur le rôle des conditions aux limites dans la déformation des tissus mous.

Malheureusement, dans le cas du foie, les CAL sont principalement donnés par les ligaments, les vaisseaux sanguins et les organes environnants, dont les propriétés ne peuvent être mesurées de manière fiable. Dans tous les cas, le comportement du foie peut être recalé à l'aide d'une caméra, d'une échographie ou d'une autre modalité, mais en raison des erreurs d'observation, il existe une grande incertitude dans le système. Par conséquent, nous avons besoin de trouver une approche qui peut permettre d'estimer les caractéristiques des attaches de l'organe en utilisant la déformation observée du tissu hépatique.

Le système d'estimation des conditions aux limites

Nous proposons une solution pour estimer les conditions aux limites du foie afin d'avoir une simulation plus précise de sa déformation. Elle est basée sur la combinaison d'un modèle hyperélastique du foie et d'une simulation non linéaire du ligament, initialisée à partir d'un atlas statistique et corrigée par un processus d'assimilation de données stochastiques. Dans notre approche, nous gardons un fort accent sur la modélisation spécifique au patient et la possibilité de déployer notre méthode dans un cadre clinique sans avoir besoin d'équipement supplémentaire.

Notre idée comporte deux étapes principales. Dans la première étape, nous créons une approximation initiale généralisée des CAL. Dans la deuxième étape, nous corrigeons l'approximation sur la base de données spécifiques au patient obtenues à partir d'une modalité utilisée dans la procédure chirurgicale.

Modèle Saint Venant-Kirchhoff pour simuler le comportement du foie

L'un des principaux défis auxquels nous avons été confrontés était lié au choix du meilleur modèle pour simuler le comportement du foie. Il doit être rapide pour effectuer une simulation en temps réel, mais aussi être capable de simuler un comportement non linéaire puisque le tissu hépatique a des propriétés hyperélastiques. Les principales possibilités disponibles sont la méthode des éléments finis corotatifs et les approches par éléments finis avec l'un des modèles de matériaux hyperélastiques de base. Pour notre expérience, nous avons choisi les modèles Saint Venant-Kirchhoff, Néo-Hookéen et Mooney-Rivlin comme étant les plus simples et nous les avons comparés au modèle corotatif. Nous avons décidé d'utiliser la déformation par flexion car les chirurgiens plient généralement les parties antérieures du foie pour accéder aux parties postérieures. Nous avons plié un objet cylindrique à l'aide de quatre modèles différents et comparé les formes obtenues.

Les résultats montrent que le modèle corotatif fonctionne aussi bien que les modèles hyperélastiques de Saint Venant-Kirchhoff, de Néo-Hookéen et de Mooney-Rivlin avec des différences de moins que 5%, tandis que le modèle linéaire a un comportement complètement différent. En matière de temps de calcul, le modèle corotatif est légèrement plus rapide que le modèle Saint Venant-Kirchhoff et plus rapide que les modèles Néo-Hookéen et Mooney-Rivlin. Cependant, malgré sa simplicité, le modèle corotationnel est assez instable, principalement parce que toute déformation peut être décrite par une combinaison de déformation linéaire et de rotation rigide. Le problème le plus grave est que cette instabilité pourrait influencer le processus d'estimation, entraînant une mauvaise estimation des paramètres. Comme les différences des résultats de simulation et de temps de calcul sont faibles entre les matériaux corotationnels et Saint Venant-Kirchhoff, nous avons finalement décidé de choisir un modèle hyperélastique avec un matériau Saint Venant-Kirchhoff pour simuler les déformations du foie.

Ressorts cubiques pour simuler les déformations des ligaments

Une autre question importante est la simulation des attaches du foie. Le point principal ici est que nous prenons en compte les CAL pour améliorer la précision prédictive du modèle. Par conséquent, nous n'avons pas besoin de simuler la déformation exacte des attaches. D'un autre côté, comme elles sont faites d'un matériau mou, nous avons besoin d'un modèle qui prenne en compte leurs déformations.

Les principales conditions aux limites du foie sont constituées par le péritoine, qui forme la plupart des ligaments (falciforme, coronaire, triangulaire, hépatogastrique et hépatoduodénal). Ces ligaments couvrent de grandes régions de la surface du foie. Ils jouent donc un rôle essentiel dans la formation des contraintes lors des déformations du foie. Le ligament rond, quant à lui, est un cordon fibreux dérivé de la veine ombilicale, et possède donc des propriétés différentes.

Il existe plusieurs approches pour modéliser les ligaments: des modèles linéaires basés sur les éléments finis (éléments finis à volume mince, triangles à déformation constante, éléments finis plans), des éléments de poutre, et aussi un modèle masse-ressort. Ce dernier est préférable pour nous car il est rapide et permet de traiter les changements topologiques d'une manière simple. À titre de comparaison, nous avons réalisé une expérience dans laquelle nous avons

étiré un rectangle échantillon, modélisé avec les éléments finis à volume mince et le modèle masse-ressort.

Comme les tissus humains ont des propriétés non linéaires, nous avons décidé d'utiliser des ressorts avec une relation contrainte-déformation non linéaire, pour une modélisation stable des attaches du foie. Malheureusement, il n'existe aucun article décrivant les propriétés biomécaniques du péritoine abdominal. Nous avons donc choisi des propriétés générales pour tous les ligaments. Les articles décrivant les ligaments de la cheville et du genou chez l'homme, ainsi que le péritoine de porc, montrent que la relation contrainte-déformation d'un ligament ressemble généralement à une courbe polynomiale. Nous avons donc décidé de prendre des ressorts cubiques avec des coefficients inconnus, qui seront identifiés au cours du processus d'estimation.

Pour valider le comportement des ressorts, nous utilisons l'approche par éléments finis hyperélastiques comme modèle de référence. Sur la base des données obtenues dans des articles, nous avons modélisé une forme rectangulaire mince en utilisant un matériau Néo-Hookéen et avons choisi une paramétrisation qui produirait des courbes de contrainte-déformation similaires. Cela a conduit à un ensemble de lois constitutives avec un module d'Young dans une certaine plage. Ensuite, nous avons généré une discrétisation du même domaine à l'aide de ressorts et assimilé les paramètres des ressorts. Puisque seule la région du ligament attaché au foie a une influence sur son mouvement, nous ne comparons que les noeuds de cette région.

Les résultats montrent une erreur moyenne entre les deux modèles d'environ plusieurs millimètres, pour des déformations allant jusqu'à 45% (c'est-à-dire couvrant la gamme de déformation du ligament en chirurgie). Cela montre que le comportement de notre système masse-ressort est similaire à celui d'un matériau Néo-Hookéen. Les expériences de performance réalisées pour les mêmes modèles montrent que le système masse-ressort est 35 fois plus rapide à calculer et peut-être généré automatiquement à partir de l'atlas.

Construction d'atlas statistique

Pour approximer les positions de ligaments au stade initial, l'idée est d'utiliser un atlas statistique. Bien que l'atlas ne puisse pas nous fournir les propriétés exactes des CAL dans le cas spécifique du patient, il nous permet de réduire la zone potentielle d'estimation des CAL. Cela permet d'éviter un grand nombre de calculs redondants, ce qui est crucial dans un contexte de modélisation en temps réel.

La construction de l'atlas est basée sur une technique appelée cartographie métrique difféomorphe à grande déformation. Dans cette méthode, un recalage déformable de modèle est considéré comme une construction d'une transformation difféomorphe entre des formes. Pour construire une forme moyenne, la forme sélectionnée est recalée sur toutes les formes d'une base de données. Et ensuite, la moyenne est calculée en tant que minimisation de la transformation différentielle globale.

Notre atlas contient une forme moyenne générée à base de données d'images obtenues à partir d'un ensemble de foies réels. D'autres formes avec des positions de ligament segmentées sont recalées sur la forme moyenne en utilisant la cartographie métrique difféomorphe. Il permet d'effectuer un recalage avec une minimisation des différences de forme. On essaye

de garder donc seulement la différence entre les positions de ligaments. Les attaches ligamentaires extraites ressemblent à un ensemble de courbes. Cependant, nous n'avons pas de description paramétrique définie pour ces dernières. Donc, pour calculer la statistique, l'ensemble des courbes est intersecté par un certain nombre de plans et les points moyens avec l'écart-type sont calculés pour chaque intersection. Enfin, la forme moyenne avec la statistique est recalée sur le patient cible du foie en utilisant la même approche de recalage déformable.

Initialisation des paramètres des ligaments

Les paramètres du ressort doivent également être initialisés d'une manière ou d'une autre. Pour définir l'approximation initiale, des recherches ont été menées sur les propriétés du péritoine de porc. Nous avons décidé de nous appuyer sur cette loi constitutive comme base pour le modèle de CAL. Les coefficients du matériau hyperélastique pour le modèle d'éléments finis non linéaires ont été calculés à l'aide de l'algorithme de Levenberg-Marquardt, qui est utilisé pour résoudre les problèmes d'ajustement de courbe par les moindres carrés non linéaires. Nous utilisons le modèle Néo-Hookéen et estimons son module d'Young.

Les paramètres du ressort ont ensuite été estimés en utilisant une approche d'assimilation de données. Nous utilisons le comportement contrainte-déformation du modèle hyperélastique comme modèle de référence. Sur la base de la description des plis du péritoine tiré de la littérature, nous avons modélisé une forme rectangulaire et un matériau Néo-Hookéen. Ensuite, nous avons généré une discrétisation du même domaine à l'aide de ressorts et assimilé les paramètres des ressorts. Puisque seule la région du ligament attaché au foie a une influence sur son mouvement, nous ne comparons que les noeuds de cette région. Les résultats de l'assimilation sont ensuite utilisés comme valeurs initiales pour l'étape de correction.

Correction stochastique

La deuxième étape, c'est la correction des conditions aux limites. Le processus de correction est soumis à plusieurs exigences. Il doit être réalisé en temps réel puisque la correction est effectuée pendant une procédure chirurgicale, mais aussi gérer les changements topologiques puisque les chirurgiens coupent souvent des ligaments pour accéder aux parties arrière du foie. Enfin, il doit gérer des observations bruitées, car les caméras médicales suivent le foie avec certaines erreurs de mesure.

Pour faire face à cette incertitude, nous proposons d'utiliser l'inférence bayésienne. Elle permet d'estimer l'emplacement et la rigidité des ligaments. En bref, cette approche utilise le théorème de Bayes pour mettre à jour la probabilité d'une hypothèse lorsque de nouvelles preuves ou informations sont disponibles. L'inférence bayésienne tient compte du bruit statistique dans les données et fournit une régularisation probabiliste, ce qui permet de résoudre les problèmes inverses avec des observations limitées.

Une implémentation efficace d'une méthode d'inférence bayésienne, capable de traiter des systèmes non linéaires comme nos modèles, est le filtre de Kalman. Dans cette méthode, l'algorithme général consiste en une boucle qui contient deux étapes principales. Dans l'étape de prédiction, nous formons la nouvelle hypothèse sur l'état estimé, tandis que dans l'étape

de correction, nous la corrigeons en comparant les mesures prédites aux observations réelles (bruitées et partielles).

Pour traiter les systèmes non linéaires, il existe deux extensions principales de l'approche de filtrage. Le filtre de Kalman étendu suppose la linéarisation de l'opérateur du processus, principalement à l'aide du Jacobien. Une autre option est le filtre de Kalman dit "unscented". Par rapport à un filtre de Kalman étendu, il ne nécessite pas le calcul du Jacobien du système, ce qui est difficile étant donné le contexte de notre problème. Les inconnues pour l'estimation (l'état stochastique du système) sont décrites comme une distribution gaussienne, dont la transformation à travers le système non linéaire est effectuée en utilisant une transformation unscented. L'idée principale est de paramétrer la distribution gaussienne en utilisant une série de points sigma, qui contiennent les informations de moyenne et de covariance, mais qui sont plus faciles à transférer à travers une transformation non linéaire.

Chaque point sigma dans l'étape de prédiction correspond à des valeurs de paramètres et à des positions de modèle perturbées. Le système d'éléments finis est résolu en prenant les positions modifiées au lieu des positions originales. Ensuite, l'état prédit est calculé comme une moyenne des résultats de la simulation d'éléments finis pour tous les points sigma. Par conséquent, l'étape de prédiction peut être très coûteuse lorsqu'on utilise un modèle comportant de nombreux degrés de liberté, comme c'est le cas lorsqu'on utilise une méthode d'éléments finis. En utilisant la méthode du simplexe pour générer les points sigma, et avec un maillage de N noeuds et K paramètres de rigidité, cela impliquerait $3N + K + 1$ simulations. Un maillage simple de seulement quelques centaines de noeuds prendrait trop de temps pour une application clinique, car il faudrait plus de trois cents simulations pour chaque étape du processus d'assimilation.

Une idée pour résoudre ce problème est d'utiliser un filtre de Kalman unscented d'ordre réduit. L'idée principale est de manipuler la matrice d'état sous la forme factorisée. Donc le coût de calcul réduit considérablement jusqu'à $K + 1$ simulations (dans le meilleur des cas).

Une autre option consiste à utiliser le filtre de Kalman d'ensemble. Ici l'idée est de présenter la distribution de probabilité comme une série de particules appelées membres de l'ensemble. Au lieu de propager la covariance d'état à travers le système non linéaire en utilisant la linéarisation ou un ensemble de points sigma, elle est approximée par l'estimateur de Monte Carlo. Par conséquent, l'état peut être prédit par un plus petit nombre de membres transformés.

Les deux options permettent de calculer les étapes de prédiction avec une quantité essentiellement plus faible de simulations réalisées.

Logiciel Optimus

Pour réaliser des expériences liées à une simulation stochastique, nous avons besoin d'un logiciel qui met en oeuvre des approches de filtrage et permet d'estimer diverses caractéristiques du modèle biomécanique. Par conséquent, nous avons implémenté notre propre plugiciel pour le logiciel SOFA, appelé Optimus. Maintenant, il est disponible gratuitement sur Internet. Nous avons créé une architecture qui permet d'ajouter une approche générique en suivant la structure prédiction-correction.

L'idée principale de ce plugiciel est la séparation entre l'état stochastique, utilisé dans

l'algorithme de Kalman, et de l'état mécanique, nécessaire pour la simulation physique. De cette façon, nous minimisons le nombre de dépendances entre les approches d'assimilation de données et les composants logiciels liés à la simulation physique. Ainsi, d'une part, nous pouvons facilement ajouter de nouvelles méthodes qui sont basées sur le schéma prédictif correcteur comme, par exemple, un filtre particulaire ou une approche séquentielle des moindres carrés. D'autre part, nous pouvons utiliser différents types de modèles basés sur la physique et d'autres composants dans SOFA, y compris ceux disponibles dans l'autres logiciels et extensions développés sur la base de SOFA. Il est intéressant de noter que le logiciel Optimus permet techniquement d'estimer ou d'identifier toute quantité spécifiée dans le système construit.

Préconditionnement pour accélérer le processus d'estimation

Comme nous l'avons noté ci-dessus, la partie la plus coûteuse en calcul du processus de filtrage est la prédiction. Pour prédire l'étape suivante de la simulation du foie, un certain nombre de simulations doivent être effectuées. La version d'ordre réduit ou le filtre Kalman d'ensemble nous permet de réduire le nombre de points sigma, mais, dans tous les cas, si nous estimons plus de quelques paramètres, l'ensemble du processus prendra encore du temps. Pour améliorer encore la simulation, nous proposons d'utiliser un préconditionneur.

En général, le préconditionneur est un opérateur qui effectue la transformation d'un problème biomécanique vers un espace beaucoup plus régulier. Cela permet de faire converger une étape du processus de simulation en quelques itérations seulement. Dans le cas idéal, le préconditionneur est l'inverse de la matrice du système, mais en pratique, il suffit d'avoir une approximation. Par conséquent, l'idée est de calculer le préconditionneur pour le premier point sigma et de l'utiliser pour tous les autres points dans des étapes de simulation données. En effet, chaque point sigma est obtenu à partir des valeurs originales en effectuant la perturbation de quelques inconnues seulement. Donc, la différence entre les matrices de simulation pour tous ces points sigma est faible et, par conséquent, la transformation de préconditionnement donne une matrice presque unique. Dans ce cas, la simulation sera effectuée rapidement pour tous les points sigma sauf le premier. Ainsi, le préconditionneur peut également améliorer les performances du processus d'estimation.

Expériences d'estimation des conditions aux limites

Pour répondre aux exigences du contexte chirurgical, nous avons choisi un modèle d'éléments finis hyperélastiques avec un matériau de Saint Venant-Kirchhoff pour simuler les déformations du foie et un système masse-ressort pour les attaches du foie. Ces deux modèles sont rapides, ce qui nous permet de les utiliser dans un contexte de temps réel. De plus, dans le modèle masse-ressort, il est facile d'effectuer des changements topologiques sans restructuration complexe du maillage. Le filtre a également besoin d'effectuer des calculs presque en temps réel.

Comparaison des versions du filtre

Tout d'abord, nous avons décidé de comparer les versions différentes de filtres. Pour les expériences, nous avons pris un objet de forme simple et appliqué des forces périodiques pour le déformer. Nous avons considéré deux scénarios possibles de conditions aux limites: des conditions fixes et le système masse-ressort. Nous estimons les coefficients des matériaux et les paramètres du ressort à la frontière ainsi que le temps de calcul.

Les résultats montrent que les deux versions réduites sont beaucoup plus rapides que le filtre de Kalman unscented. Et en général, la version réduite est plus rapide que le filtre de Kalman d'ensemble, tout en ayant des résultats d'estimation similaires. En réduisant davantage le nombre de membres de l'ensemble, la version d'ensemble pourrait être plus rapide que la version réduite, mais cela pourrait également entraîner des instabilités de simulation et des divergences dans le processus d'assimilation des données.

Nous avons également réalisé une expérience pour observer comment les filtres se comporteront en cas de changement topologique. Pour effectuer ce changement, nous avons simulé la coupure des ligaments en fixant simplement leur rigidité à des valeurs nulles. Les résultats montrent que les filtres sont capables de s'adapter rapidement à ces changements, avec des résultats de performance similaires aux autres expériences. C'est pourquoi, finalement, nous avons décidé de nous appuyer sur la version d'ordre réduit du filtrage de Kalman unscented.

Pour vérifier l'amélioration des performances en cas de préconditionneur, nous avons réalisé une expérience avec le même modèle. Le modèle était fixé par des ressorts, dont la rigidité était estimée. Pendant le processus d'estimation, nous avons enregistré le temps de calcul de milliers d'itérations pour les cas avec et sans préconditionneur. Les résultats montrent qu'avec le préconditionneur, le temps de calcul est presque 25% moins élevé que sans lui. D'autre part, la différence entre les valeurs estimées est faible et peut-être négligée. Globalement, le préconditionneur n'améliore pas de manière significative les performances pour l'ensemble du processus d'estimation, mais nous pouvons tout de même obtenir un certain avantage.

Expériences avec modèle du foie et observations synthétiques

Nous avons d'abord validé notre modèle de foie optimisé par rapport à une solution numérique. Le foie a été généré à partir d'un scanner segmenté d'un patient, et l'emplacement des ligaments a été défini par un expert. Pour des raisons de simplicité, nous avons décidé de ne pas considérer d'atlas et de se concentrer sur le ligament falciforme.

La simulation de référence est calculée à l'aide du logiciel FEBio. Le foie a été maillé avec environ soixante-dix mille éléments tétraédriques linéaires. Les mailles du ligament (éléments coques) ont été générées par extrusion dans la direction de la normale à la surface du foie. Des contraintes de Dirichlet ont été appliquées aux noeuds les plus éloignés du foie, et des contraintes bilatérales ont été utilisées pour coupler les modèles de foie et de ligament. Un modèle d'Ogden a été utilisé pour le foie et un matériau Néo-Hookéen pour les ligaments. Pour générer une déformation représentative de la manipulation chirurgicale, nous avons appliqué des charges périodiques aux deux lobes du foie. Nous avons également défini plusieurs

marqueurs virtuels pour représenter les observations qui pourraient être obtenues lors d'une chirurgie réelle. Ils ont été utilisés dans le processus d'assimilation des données.

D'autre part, le modèle de foie et de ligament en temps réel a été implémenté en utilisant le logiciel SOFA. Le maillage du foie a été généré avec environ plusieurs milliers d'éléments. Nous avons utilisé un modèle Saint Venant-Kirchhoff, pour lequel un coefficient de Poisson de matériau presque incompressible. Les ligaments ont été également extrudés le long de la normale à la surface du foie comme deux couches de ressorts non linéaires. Les paramètres des ressorts ont été déduits de la courbe contrainte-déformation du péritoine de porc en utilisant la méthode d'initialisation décrite précédemment. Pour optimiser davantage le processus d'assimilation, nous avons divisé l'ensemble des ressorts en groupes qui partagent les mêmes paramètres de rigidité. De cette façon, nous avons estimé les paramètres pour chaque région, mais pas pour chaque ressort.

Pour évaluer nos résultats, nous avons utilisé des marqueurs virtuels uniformément répartis sur le modèle de foie. Pour montrer comment notre estimation des CAL peut améliorer la précision globale de la déformation du foie, nous avons comparé les positions des marqueurs obtenues avec notre méthode avec une simulation impliquant le même modèle de foie mais utilisant des CAL fixes. Les résultats montrent que pour les CAL estimés avec notre méthode, l'erreur moyenne de déformation n'est que de 2 millimètres, alors qu'elle est de 7 millimètres lorsqu'on utilise des CAL constants prédéfinis. Pour la plus grande déformation, l'erreur est réduite d'environ quatre fois.

Résultats de construction de l'atlas statistique

Pour la construction de l'atlas, nous avons pris 15 modèles de foie où les ligaments ont été segmentés par un expert. Nous avons utilisé 14 maillages pour générer l'atlas statistique et 1 pour la validation. Nous avons également réalisé des statistiques pour les ligaments falciforme, coronaire et triangulaire.

La construction de la forme moyenne et le recalage déformable étaient basés sur la cartographie métrique difféomorphe à grande déformation. Pour l'effectuer, on a utilisé le logiciel Deformetrica. Ensuite, pour calculer les statistiques des courbes, nous avons intersecté la région d'intérêt avec une série de plans. Pour générer des statistiques en cas de ligament falciforme, nous avons utilisé des plans radiaux, tandis que pour les ligaments coronaux et triangulaires, nous avons sélectionné une série de plans parallèles. Enfin, en utilisant la cartographie barycentrique et recalage déformable, nous avons fait correspondre les résultats statistiques à la forme moyenne, recalé la forme sur un modèle de validation, et extrait les statistiques avec les coordonnées barycentriques.

Les résultats montrent que la différence entre la moyenne statistique et la segmentation de la vérité fondamentale dépend de la position du ligament et atteint 3,5 centimètres. Quoi qu'il en soit, d'après notre expérience, il s'agit de la meilleure approximation que nous connaissions pour initialiser les positions du ligament. Et nous pensons que nous pouvons encore compenser cette différence en estimant des paramètres de rigidité légèrement différents.

Expériences avec foie humain

Finalement, nous avons réalisé une expérience basée sur des données réelles. Nous avons pris un cadavre humain et simulé la véritable procédure chirurgicale. Nous avons attaché plusieurs marqueurs à la surface du foie qui servent comme observations et des points profonds pour valider la différence entre les déformations. Plusieurs images de tomographie par ordinateur ont été acquises au cours de l'expérience. Pour effectuer la manipulation, on a utilisé des outils chirurgicaux laparoscopiques. Pendant la manipulation, nous avons enregistré les tackers à l'aide d'une caméra RGB-D.

À partir de l'image tomographie par ordinateur, nous avons reconstruit le modèle volumique du foie ainsi que les positions des marqueurs. Nous avons construit ensuite un atlas statistique à partir de 15 modèles comme décrit ci-dessus et l'enregistrons sur le maillage volumique. Sur la base de l'atlas statistique, nous avons généré les attaches de ressort. Les ligaments ont été extrudés le long de la normale à la surface du foie. Comme les tackers sont également visibles sur la séquence de données RGB-D, l'enregistrement du modèle reconstruit à partir des données RGB-D a été effectué en minimisant la différence entre les tackers à l'aide d'une approche itérative du point le plus proche. L'étape finale de l'expérience consiste à effectuer un processus d'assimilation de données et à montrer la différence entre les repères profonds. Pour l'assimilation nous avons utilisé un modèle Saint Venant-Kirchhoff, pour lequel un coefficient de Poisson de matériau presque incompressible.

Pour la validation, nous avons extrait des marqueurs d'une autre tomographie par ordinateur, où le foie est déformé. Nous avons comparé les cas où seuls les vaisseaux sanguins ont été fixés, avec estimation initiale des ligaments et avec estimation des paramètres des ligaments. Les résultats montrent que la précision s'améliore lorsque nous ajoutons un modèle de ligament et que nous l'estimons ensuite, mais le résultat dépend également de la façon dont nous effectuons le processus de validation, qui peut ne pas être très précis. Cela pourrait aussi dépendre de questions supplémentaires comme les contraintes unilatérales que nous ne prenons pas en compte.

Conclusion

Dans ce travail, nous avons proposé une nouvelle méthode pour estimer des CAL, basée sur une approche de filtrage de Kalman unscented. Les CAL sont présentées comme des ressorts avec des paramètres inconnus. Le filtre les estime en utilisant des informations spécifiques au patient obtenues pendant la chirurgie. L'approximation initiale des CAL est dérivée d'un atlas statistique, construit à partir d'une série de modèles de foie avec des ligaments segmentés, et d'une loi constitutive expérimentale.

Nous pensons que la solution proposée a un grand potentiel. Le recalage de l'atlas permet de se concentrer uniquement sur les zones spécifiques d'intérêt et ne prend pas en compte les autres régions. Le processus d'assimilation des données nous permet de corriger l'approximation initiale en fonction des données obtenues sur un patient spécifique, ce qui constitue un grand avantage. En général, l'anatomie de chaque personne présente des variations spécifiques, ce qui la rend différente d'une moyenne statistique. Le processus de filtrage nous permet aussi

d'effectuer la correction des paramètres dans un contexte proche du temps réel. Ceci est essentiel dans les cas où les données recalées doivent être traitées pendant la procédure, ou lorsqu'il n'y a pas de temps supplémentaire pour les calculs.

Notre principal domaine d'intérêt est la simulation des tissus mous dans le contexte clinique. Cependant, l'approche proposée pourrait être utilisée dans des domaines divers. La méthode des éléments finis est l'une des méthodes les plus populaires pour résoudre les systèmes différentiels partiels. Le préconditionnement est une approche courante pour améliorer la convergence des systèmes linéaires. Le filtrage de Kalman est également une approche générale pour l'estimation de certaines inconnues dans un processus dynamique. Ainsi, les principales restrictions de l'approche proposée sont la présence d'une dynamique dans un événement considéré et l'existence d'une observation qui le suit. Nous essayons également d'estimer les CAL du foie, mais dans le cas général, nous pourrions estimer divers paramètres du modèle construit créé pour n'importe quel organe ou tissu. Les paramètres peuvent être dépendants ou indépendants du temps. De plus, dans notre travail, nous nous concentrons principalement sur les caméras, mais en général, d'autres modalités d'imagerie médicales pourraient être utilisées, telles que les sondes à ultrasons, les trackers magnétiques ou les capteurs infrarouges.

Par conséquent, il existe de nombreuses possibilités d'utiliser l'approche proposée et ses parties pour résoudre les problèmes qui apparaissent dans des situations diverses.



Bibliography



- [1] S. R. Z. Abdel-Misih and M. Bloomston, “Liver anatomy,” *Surgical Clinics of North America*, vol. 90, no. 4, pp. 643–653, 2010, liver Surgery: From Basics to Robotics.
- [2] C. C. Aggarwal, *Neural Networks and Deep Learning: A textbook*. Springer, Cham, 2018.
- [3] M. R. Aguirre, M. G. Linguraru, K. Marias, N. Ayache, L.-p. Nolte, and M. A. Gonzalez Ballester, “Statistical shape analysis via principal factor analysis,” in *2007 4th IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, 2007, pp. 1216–1219.
- [4] B. Ahn and J. Kim, “Measurement and characterization of soft tissue behavior with surface deformation and force response under large deformations,” *Medical Image Analysis*, vol. 14, no. 2, pp. 138–148, 2010.
- [5] J. Allard, S. Cotin, F. Faure, P.-J. Bensoussan, F. Poyer, C. Duriez, H. Delingette, and L. Grisoni, “Sofa - an open source framework for medical simulation,” Feb 2007, vol. 125, pp. 13–18.
- [6] H. O. Altamar, R. E. Ong, C. L. Glisson, D. P. Viprakasit, M. I. Miga, S. D. Herrell, and R. L. Galloway, “Kidney deformation and intraprocedural registration: A study of elements of image-guided kidney surgery,” *Journal of Endourology*, vol. 25, no. 3, pp. 511–517, 2011.
- [7] P. Alvarez, S. Narasimhan, S. Rouzé, J.-L. Dillenseger, Y. Payan, M. I. Miga, and M. Chabanas, “Biphasic model of lung deformations for video-assisted thoracoscopic surgery (vats),” pp. 1367–1371, 2019.
- [8] V. Andersen, J. Sonne, S. Sletting, and A. Prip, “The volume of the liver in patients correlates to body weight and alcohol consumption,” *Alcohol and Alcoholism*, vol. 35, no. 5, pp. 531–532, 09 2000.
- [9] I. M. Arias, H. J. Alter, J. L. Boyer, D. E. Cohen, N. Fausto, D. A. Shafritz, and A. W. Wolkoff, *The Liver: Biology and Pathobiology: Fifth Edition*. John Wiley & Sons, Ltd., 10 2009.

- [10] M. Asch, M. Bocquet, and M. Nodet, *Data Assimilation: Methods, Algorithms, and Applications*. Society for Industrial and Applied Mathematics, 2016.
- [11] V. M. Banz, M. Baechtold, S. Weber, M. Peterhans, D. Inderbitzin, and D. Candinas, “Computer planned, image-guided combined resection and ablation for bilobar colorectal liver metastases.” *World journal of gastroenterology*, vol. 20, no. 40, pp. 14 992–14 996, 2014.
- [12] V. M. Banz, P. C. Müller, P. Tinguely, D. Inderbitzin, D. Ribes, M. Peterhans, D. Candinas, and S. Weber, “Intraoperative image-guided navigation system: development and applicability in 65 patients undergoing liver surgery,” *Langenbeck’s Archives of Surgery*, vol. 401, no. 4, pp. 495–502, 2016.
- [13] P. Bao, T. K. Sinha, C.-C. R. Chen, J. R. Warmath, R. L. Galloway, and A. J. Herline, “A prototype ultrasound-guided laparoscopic radiofrequency ablation system,” *Surgical Endoscopy*, vol. 21, no. 1, pp. 74–79, 2007.
- [14] P. Q. Bao, J. Warmath, R. L. Galloway, and A. J. Herline, “Ultrasound-to-computer-tomography registration for image-guided laparoscopic liver surgery,” *Surgical Endoscopy And Other Interventional Techniques*, vol. 19, no. 3, pp. 424–429, 2005.
- [15] M. Baumhauer, T. Simpfendorfer, B. P. Müller-Stich, D. Teber, C. N. Gutt, J. Rassweiler, H.-P. Meinzer, and I. Wolf, “Soft tissue navigation for laparoscopic partial nephrectomy,” *International Journal of Computer Assisted Radiology and Surgery*, vol. 3, no. 3, pp. 307–314, 2008.
- [16] M. Becker, M. Kirschner, S. Fuhrmann, and S. Wesarg, “Automatic construction of statistical shape models for vertebrae,” in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2011*, G. Fichtinger, A. Martel, and T. Peters, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 500–507.
- [17] J. G. Betts, K. A. Young, J. A. Wise, E. Johnson, B. Poe, D. H. Kruse, O. Korol, J. E. Johnson, M. Womble, and P. DeSaix, *Anatomy and Physiology*. OpenStax, 2013.
- [18] J. M. Blackall, G. P. Penney, A. P. King, and D. J. Hawkes, “Alignment of sparse freehand 3-d ultrasound with preoperative images of the liver using models of respiratory motion and deformation,” *IEEE Transactions on Medical Imaging*, vol. 24, no. 11, pp. 1405–1416, 2005.
- [19] A. Bône, M. Louis, B. Martin, and S. Durrleman, “Deformetrica 4: An open-source software for statistical shape analysis,” in *Shape in Medical Imaging*, M. Reuter, C. Wachinger, H. Lombaert, B. Paniagua, M. Lüthi, and B. Egger, Eds. Cham: Springer International Publishing, 2018, pp. 3–13.

- [20] 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 *Medical Image Computing and Computer Assisted Intervention – MICCAI 2019*, D. Shen, T. Liu, T. M. Peters, L. H. Staib, C. Essert, S. Zhou, P.-T. Yap, and A. Khan, Eds. Cham: Springer International Publishing, 2019, pp. 137–145.
- [21] R. P. Burke, Z. Xu, C. P. Lee, R. B. Baucom, B. K. Poulouse, R. G. Abramson, and B. A. Landman, “Multi-atlas segmentation for abdominal organs with gaussian mixture models,” in *Medical Imaging 2015: Biomedical Applications in Molecular, Structural, and Functional Imaging*, B. Gimi and R. C. Molthen, Eds., vol. 9417, International Society for Optics and Photonics. SPIE, 2015, pp. 49–54.
- [22] F. J. Carter, T. G. Frank, P. J. Davies, D. A. McLean, and A. Cuschieri, “Measurements and modelling of the compliance of human and porcine organs,” *Medical Image Analysis*, vol. 5, no. 4, pp. 231–236, 2001.
- [23] D. M. Cash, M. I. Miga, S. C. Glasgow, B. M. Dawant, L. W. Clements, Z. Cao, R. L. Galloway, and W. C. Chapman, “Concepts and preliminary data toward the realization of image-guided liver surgery,” *Journal of Gastrointestinal Surgery*, vol. 11, no. 7, pp. 844–859, 2007.
- [24] N. Charon and A. Trouvé, “The varifold representation of nonoriented shapes for diffeomorphic registration,” *SIAM Journal on Imaging Sciences*, vol. 6, no. 4, pp. 2547–2580, 2013.
- [25] L. Chen, W. Tang, and N. W. John, “Real-time geometry-aware augmented reality in minimally invasive surgery,” *Healthcare Technology Letters*, vol. 4, no. 5, pp. 163–167, October 2017.
- [26] N. Chopin and O. Papaspiliopoulos, *An Introduction to Sequential Monte Carlo*. Springer, Cham, 2020.
- [27] C.-K. Chui, E. Kobayashi, X. Chen, T. Hisada, and I. Sakuma, “Combined compression and elongation experiments and non-linear modelling of liver tissue for surgical simulation,” *Medical and Biological Engineering and Computing*, vol. 42, no. 6, pp. 787–798, 2004.
- [28] C.-K. Chui, E. Kobayashi, X. Chen, T. Hisada, and I. Sakuma, “Transversely isotropic properties of porcine liver tissue: experiments and constitutive modelling,” *Medical & Biological Engineering & Computing*, vol. 45, no. 1, pp. 99–106, 2007.
- [29] H. Chui and A. Rangarajan, “A new point matching algorithm for non-rigid registration,” *Computer Vision and Image Understanding*, vol. 89, no. 2, pp. 114–141, 2003.
- [30] P. G. Ciarlet, *Theory of Shells, Volume 3*. Amsterdam: Elsevier, 2000.

- [31] R. Ciria, D. Cherqui, D. A. Geller, J. Briceno, and G. Wakabayashi, “Comparative short-term benefits of laparoscopic liver resection: 9000 cases and climbing,” *Annals of Surgery*, vol. 263, no. 4, pp. 761–777, 2016.
- [32] O. Comas, S. Cotin, and C. Duriez, “A shell model for real-time simulation of intra-ocular implant deployment,” in *Biomedical Simulation*, F. Bello and S. Cotin, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 160–170.
- [33] S. Cotin, H. Delingette, and N. Ayache, “A hybrid elastic model for real-time cutting, deformations and force feedback for surgery training and simulation,” *The Visual Computer*, vol. 16, no. 8, pp. 437–452, 2000.
- [34] H. Courtecuisse, J. Allard, C. Duriez, and S. Cotin, “Asynchronous preconditioners for efficient solving of non-linear deformations,” in *Workshop in Virtual Reality Interactions and Physical Simulation VRIPHYS (2010)*, K. Erleben, J. Bender, and M. Teschner, Eds. The Eurographics Association, 2010.
- [35] 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.
- [36] H. Courtecuisse, H. Jung, J. Allard, C. Duriez, D. Y. Lee, and S. Cotin, “Gpu-based real-time soft tissue deformation with cutting and haptic feedback,” *Progress in Biophysics and Molecular Biology*, vol. 103, no. 2, pp. 159–168, 2010.
- [37] W. J. Daniel, “Three-dimensional orthotropic viscoelastic finite element model of a human ligament,” *Computer Methods in Biomechanics and Biomedical Engineering*, vol. 4, no. 3, pp. 265–279, 2001.
- [38] F. Dorema, “Dynamic data driven applications systems: A new paradigm for application simulations and measurements,” in *Computational Science - ICCS 2004*, M. Bubak, G. D. van Albada, P. M. A. Sloot, and J. Dongarra, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 662–669.
- [39] L. E. DeFrate and G. Li, “The prediction of stress-relaxation of ligaments and tendons using the quasi-linear viscoelastic model,” *Biomechanics and Modeling in Mechanobiology*, vol. 6, no. 4, pp. 245–251, 2007.
- [40] H. Delingette and N. Ayache, “Soft tissue modeling for surgery simulation,” in *Computational Models for the Human Body*, ser. Handbook of Numerical Analysis. Elsevier, 2004, vol. 12, pp. 453–550.
- [41] F. J. Detmer, J. Hettig, D. Schindele, M. Schostak, and C. Hansen, “Virtual and augmented reality systems for renal interventions: A systematic review,” *IEEE Reviews in Biomedical Engineering*, vol. 10, pp. 78–94, 2017.

- [42] J. O. Dow, "Lesson 10 - six-node linear strain element," in *A Unified Approach to the Finite Element Method and Error Analysis Procedures*. San Diego: Academic Press, 1999, pp. 233–248.
- [43] J. O. Dow, "Lesson 8 - the development of strain-gradient-based finite elements," in *A Unified Approach to the Finite Element Method and Error Analysis Procedures*. San Diego: Academic Press, 1999, pp. 169–210.
- [44] E. Dura, J. Domingo, A. Rojas-Arboleda, and L. Marti-Bonmati, "Mean sets for building 3d probabilistic liver atlas from perfusion mr images," in *2012 3rd International Conference on Image Processing Theory, Tools and Applications (IPTA)*, 2012, pp. 186–191.
- [45] E. Dura, J. Domingo, G. Ayala, L. Marti-Bonmati, and E. Goceri, "Probabilistic liver atlas construction," *BioMedical Engineering OnLine*, vol. 16, no. 15, 2017.
- [46] C. Duriez, S. Cotin, L. Lenoir, and P. Neumann, "New approaches to catheter navigation for interventional radiology simulation," *Computer Aided Surgery*, vol. 11, no. 6, pp. 300–308, 2006.
- [47] S. Durrleman, M. Prastawa, N. Charon, J. R. Korenberg, S. Joshi, G. Gerig, and A. Trouvé, "Morphometry of anatomical shape complexes with dense deformations and sparse parameters," *NeuroImage*, vol. 101, pp. 35–49, 2014.
- [48] S. Durrleman, M. Prastawa, J. R. Korenberg, S. Joshi, A. Trouvé, and G. Gerig, "Topology preserving atlas construction from shape data without correspondence using sparse parameters," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2012*, N. Ayache, H. Delingette, P. Golland, and K. Mori, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 223–230.
- [49] B. J. Ellis, T. J. Lujan, M. S. Dalton, and J. A. Weiss, "Medial collateral ligament insertion site and contact forces in the acl-deficient knee," *Journal of Orthopaedic Research*, vol. 24, no. 4, pp. 800–810, 2006.
- [50] K. M. Elsayes, P. T. Staveteig, V. R. Narra, J. R. Leyendecker, J. S. Lewis, and J. J. Brown, "Mri of the peritoneum: Spectrum of abnormalities," *American Journal of Roentgenology*, vol. 186, no. 5, pp. 1368–1379, May 2006.
- [51] A. C. Evans, A. L. Janke, D. L. Collins, and S. Baillet, "Brain templates and atlases," *NeuroImage*, vol. 62, no. 2, pp. 911–922, 2012.
- [52] G. Evensen, "Sequential data assimilation with a nonlinear quasi-geostrophic model using monte carlo methods to forecast error statistics," *Journal of Geophysical Research: Oceans*, vol. 99, no. C5, pp. 10 143–10 162, 1994.
- [53] J. Ferlay, D. Parkin, and E. Steliarova-Foucher, "Estimates of cancer incidence and mortality in europe in 2008," *European Journal of Cancer*, vol. 46, no. 4, pp. 765–781, 2010.

- [54] M. Feuerstein, T. Mussack, S. M. Heining, and N. Navab, "Registration-free laparoscope augmentation for intra-operative liver resection planning," in *Medical Imaging 2007: Visualization and Image-Guided Procedures*, K. R. Cleary and M. I. Miga, Eds., vol. 6509, International Society for Optics and Photonics. SPIE, 2007, pp. 404–411.
- [55] M. Feuerstein, T. Mussack, S. M. Heining, and N. Navab, "Intraoperative laparoscope augmentation for port placement and resection planning in minimally invasive liver resection," *IEEE Transactions on Medical Imaging*, vol. 27, no. 3, pp. 355–369, 2008.
- [56] I. Figueroa-Garcia, J.-M. Peyrat, G. Hamarneh, and R. Abugharbieh, "Biomechanical kidney model for predicting tumor displacement in the presence of external pressure load," in *2014 IEEE 11th International Symposium on Biomedical Imaging (ISBI)*, 2014, pp. 810–813.
- [57] E. A. for the Study of the Liver, "Clinical practice guidelines: Management of hepatocellular carcinoma," *Journal of Hepatology*, vol. 69, no. 1, pp. 182–236, 2018.
- [58] A. Forestiero, E. L. Carniel, and A. N. Natali, "Biomechanical behaviour of ankle ligaments: constitutive formulation and numerical modelling," *Computer Methods in Biomechanics and Biomedical Engineering*, vol. 17, no. 4, pp. 395–404, 2014.
- [59] A. H. Foruzan, Y.-W. Chen, M. Hori, Y. Sato, and N. Tomiyama, "Capturing large shape variations of liver using population-based statistical shape models," *International Journal of Computer Assisted Radiology and Surgery*, vol. 9, no. 6, pp. 967–977, 2014.
- [60] L. Frajhof, J. Borges, E. Hoffmann, J. Lopes, and R. Haddad, "Virtual reality, mixed reality and augmented reality in surgical planning for video or robotically assisted thoracoscopic anatomic resections for treatment of lung cancer," *Journal of Visualized Surgery*, vol. 4, no. 7, 2018.
- [61] Å. A. Fretland, V. J. Dagenborg, G. M. W. Bjørnelv, A. M. Kazaryan, R. Kristiansen, M. W. Fagerland, J. Hausken, T. I. Tønnessen, A. Abildgaard, L. Barkhatov, S. Yaqub, B. I. Røsok, B. A. Bjørnbeth, M. H. Andersen, K. Flatmark, E. Aas, and B. Edwin, "Laparoscopic versus open resection for colorectal liver metastases: The oslo-comet randomized controlled trial," *Annals of Surgery*, vol. 267, no. 2, pp. 199–207, 2018.
- [62] K. O. Friedrichs, "On the boundary-value problems of the theory of elasticity and korn's inequality," *Annals of Mathematics*, vol. 48, no. 2, pp. 441–471, 1947.
- [63] Y. B. Fu and C.-K. Chui, "Modelling and simulation of porcine liver tissue indentation using finite element method and uniaxial stress–strain data," *Journal of Biomechanics*, vol. 47, no. 10, pp. 2430–2435, 2014.
- [64] Y. B. Fu, C.-K. Chui, and C. Teo, "Liver tissue characterization from uniaxial stress–strain data using probabilistic and inverse finite element methods," *Journal of the Mechanical Behavior of Biomedical Materials*, vol. 20, pp. 105–112, 2013.

- [65] Y. Fung, *Biomechanics, Mechanical Properties of Living Tissues*, 2nd ed. Springer-Verlag New York, 1993.
- [66] J. R. Funk, G. W. Hall, J. R. Crandall, and W. D. Pilkey, “Linear and quasi-linear viscoelastic characterization of ankle ligaments,” *Journal of Biomechanical Engineering*, vol. 122, no. 1, pp. 15–22, 09 2000.
- [67] R. Furukawa, G. Nagamatsu, S. Oka, T. Kotachi, Y. Okamoto, S. Tanaka, and H. Kawasaki, “Simultaneous shape and camera-projector parameter estimation for 3d endoscopic system using cnn-based grid-oneshot scan,” *Healthcare Technology Letters*, vol. 6, no. 6, pp. 249–254, 2019.
- [68] M. Fusaglia, H. Hess, M. Schwalbe, M. Peterhans, P. Tinguely, and S. Weber, “A clinically applicable laser-based image-guided system for laparoscopic liver procedures,” *International Journal of Computer Assisted Radiology and Surgery*, vol. 11, no. 8, pp. 1499–1513, 2016.
- [69] Z. Gao and J. P. Desai, “Estimating zero-strain states of very soft tissue under gravity loading using digital image correlation,” *Medical Image Analysis*, vol. 14, no. 2, pp. 126–137, 2010.
- [70] J. C. Gardiner and J. A. Weiss, “Subject-specific finite element analysis of the human medial collateral ligament during valgus knee loading,” *Journal of Orthopaedic Research*, vol. 21, no. 6, pp. 1098–1106, 2003.
- [71] T. C. Gasser, R. W. Ogden, and G. A. Holzapfel, “Hyperelastic modelling of arterial layers with distributed collagen fibre orientations,” *Journal of The Royal Society Interface*, vol. 3, no. 6, pp. 15–35, 2006.
- [72] H. Gray, *Anatomy of the Human Body*. Philadelphia: Lea & Febiger, 1918.
- [73] E. S. Grood and M. S. Hefzy, “An analytical technique for modeling knee joint stiffness—part i: Ligamentous forces,” *Journal of Biomechanical Engineering*, vol. 104, no. 4, pp. 330–337, 11 1982.
- [74] 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 *2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, Oct 2013, pp. 199–208.
- [75] S. Haykin, *Neural Networks and Learning Machines, 3rd Edition*. Pearson, 2008.
- [76] M. S. Hefny, T. Okada, M. Hori, Y. Sato, and R. E. Ellis, “A liver atlas using the special euclidean group,” in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, N. Navab, J. Hornegger, W. M. Wells, and A. Frangi, Eds. Cham: Springer International Publishing, 2015, pp. 238–245.

- [77] M. R. Hestenes and E. Stiefel, "Methods of conjugate gradients for solving linear systems," *J Res NIST*, vol. 49, no. 6, pp. 409–436, 1952.
- [78] S. Hirokawa and R. Tsuruno, "Three-dimensional deformation and stress distribution in an analytical/computational model of the anterior cruciate ligament," *Journal of Biomechanics*, vol. 33, no. 9, pp. 1069–1077, 2000.
- [79] K. D. Hjelmstad, "Boundary value problems in elasticity," in *Fundamentals of Structural Mechanics*. Boston, MA: Springer US, 2005, pp. 159–192.
- [80] G. A. Holzapfel, *Nonlinear Solid Mechanics: A Continuum Approach for Engineering*. New York: Wiley, 2000.
- [81] T. Hu and J. P. Desai, "Characterization of soft-tissue material properties: Large deformation analysis," in *Medical Simulation*, S. Cotin and D. Metaxas, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 28–37.
- [82] Y. Hu, H. U. Ahmed, Z. Taylor, C. Allen, M. Emberton, D. Hawkes, and D. Barratt, "Mr to ultrasound registration for image-guided prostate interventions," *Medical Image Analysis*, vol. 16, no. 3, pp. 687–703, 2012, computer Assisted Interventions.
- [83] B. R. Hunt, E. J. Kostelich, and I. Szunyogh, "Efficient data assimilation for spatiotemporal chaos: A local ensemble transform kalman filter," *Physica D: Nonlinear Phenomena*, vol. 230, no. 1, pp. 112–126, 2007.
- [84] A. Hutt, "Divergence of the ensemble transform kalman filter (letkf) by nonlocal observations," *Frontiers in Applied Mathematics and Statistics*, vol. 6, pp. 1–42, 2020.
- [85] A. Isaza-Restrepo, J. S. Martin-Saavedra, J. L. Velez-Leal, F. Vargas-Barato, and R. Riveros-Dueñas, "The peritoneum: Beyond the tissue - a review," *Frontiers in physiology*, vol. 9, no. 738, Jun 2018.
- [86] R. K. Jain, J. D. Martin, and T. Stylianopoulos, "The role of mechanical forces in tumor growth and therapy," *Annual Review of Biomedical Engineering*, vol. 16, no. 1, pp. 321–346, 2014.
- [87] S. F. Johnsen, S. A. Thompson, M. J. Clarkson, M. Modat, Y. Song, J. Totz, K. S. G. Gurusamy, B. R. Davidson, Z. A. Taylor, D. J. Hawkes, and S. Ourselin, "Database-based estimation of liver deformation under pneumoperitoneum for surgical image-guidance and simulation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, N. Navab, J. Hornegger, W. M. Wells, and A. Frangi, Eds. Cham: Springer International Publishing, 2015, pp. 450–458.
- [88] P. Jordan, S. Socrate, T. Zickler, and R. Howe, "Constitutive modeling of porcine liver in indentation using 3d ultrasound imaging," *Journal of the Mechanical Behavior of Biomedical Materials*, vol. 2, no. 2, pp. 192–201, 2009.

- [89] S. J. Julier and J. K. Uhlmann, "New extension of the kalman filter to nonlinear systems," *Proc.SPIE*, vol. 3068, p. 12 , 1997.
- [90] S. J. Julier and J. K. Uhlmann, "Reduced sigma point filters for the propagation of means and covariances through nonlinear transformations," in *Proceedings of the 2002 American Control Conference*, vol. 2, 2002, pp. 887–892.
- [91] S. J. Julier, J. K. Uhlmann, and H. F. Durrant-Whyte, "A new approach for filtering nonlinear systems," in *Proceedings of 1995 American Control Conference - ACC'95*, vol. 3, June 1995, pp. 1628–1632.
- [92] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, March 1960.
- [93] M. Kass, "An introduction to physically based modeling: An introduction to continuum dynamics for computer graphics," <http://www.cs.cmu.edu/~7Ebaraff/pbm/continuator.pdf>, 1997, online; accessed 24 September 2020.
- [94] J. E. Kelleher, T. Siegmund, M. Du, E. Naseri, and R. W. Chan, "The anisotropic hyperelastic biomechanical response of the vocal ligament and implications for frequency regulation: A case study," *The Journal of the Acoustical Society of America*, vol. 133, no. 3, pp. 1625–1636, 2013.
- [95] J. Kim and M. A. Srinivasan, "Characterization of viscoelastic soft tissue properties from in vivo animal experiments and inverse fe parameter estimation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2005*, J. S. Duncan and G. Gerig, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 599–606.
- [96] T. P. Kingham, S. Jayaraman, L. W. Clements, M. A. Scherer, J. D. Stefansic, and W. R. Jarnagin, "Evolution of image-guided liver surgery: Transition from open to laparoscopic procedures," *Journal of Gastrointestinal Surgery*, vol. 17, no. 7, pp. 1274–1282, 2013.
- [97] T. P. Kingham, M. A. Scherer, B. W. Neese, L. W. Clements, J. D. Stefansic, and W. R. Jarnagin, "Image-guided liver surgery: intraoperative projection of computed tomography images utilizing tracked ultrasound," vol. 14, no. 9, pp. 596–603, 2012.
- [98] M. Z. Kiss, T. Varghese, and T. J. Hall, "Viscoelastic characterization of in vitro canine tissue," *Physics in Medicine and Biology*, vol. 49, no. 18, pp. 4207–4218, Aug 2004.
- [99] Y. Kobayashi, H. Watanabe, T. Hoshi, K. Kawamura, and M. G. Fujie, "Viscoelastic and nonlinear liver modeling for needle insertion simulation," in *Soft Tissue Biomechanical Modeling for Computer Assisted Surgery*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 41–67.
- [100] S. Kohara, T. Tateyama, A. H. Foruzan, A. Furukawa, S. Kanasaki, M. Wakamiya, and Y.-W. Chen, "Application of statistical shape model to diagnosis of liver disease," in

- The 2nd International Conference on Software Engineering and Data Mining*, 2010, pp. 680–683.
- [101] S. Kohara, T. Tateyama, A. H. Foruzan, A. Furukawa, S. Kanasaki, M. Wakamiya, X. Wei, and Y.-W. Chen, “Preliminary study on statistical shape model applied to diagnosis of liver cirrhosis,” in *2011 18th IEEE International Conference on Image Processing*, 2011, pp. 2921–2924.
- [102] T. Köhler, S. Haase, S. Bauer, J. Wasza, T. Kilgus, L. Maier-Hein, H. Feußner, and J. Hornegger, “Tof meets rgb: Novel multi-sensor super-resolution for hybrid 3-d endoscopy,” in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2013*, K. Mori, I. Sakuma, Y. Sato, C. Barillot, and N. Navab, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 139–146.
- [103] T. Köhler, S. Haase, S. Bauer, J. Wasza, T. Kilgus, L. Maier-Hein, C. Stock, J. Hornegger, and H. Feußner, “Multi-sensor super-resolution for hybrid range imaging with application to 3-d endoscopy and open surgery,” *Medical Image Analysis*, vol. 24, no. 1, pp. 220–234, 2015.
- [104] K. Konishi, M. Hashizume, M. Nakamoto, Y. Kakeji, I. Yoshino, A. Taketomi, Y. Sato, S. Tamura, and Y. Maehara, “Augmented reality navigation system for endoscopic surgery based on three-dimensional ultrasound and computed tomography: Application to 20 clinical cases,” *International Congress Series*, vol. 1281, pp. 537–542, 2005.
- [105] K. Konishi, M. Nakamoto, Y. Kakeji, K. Tanoue, H. Kawanaka, S. Yamaguchi, S. Ieiri, Y. Sato, Y. Maehara, S. Tamura, and M. Hashizume, “A real-time navigation system for laparoscopic surgery based on three-dimensional ultrasound using magneto-optic hybrid tracking configuration,” *International Journal of Computer Assisted Radiology and Surgery*, vol. 2, no. 1, pp. 1–10, 2007.
- [106] J. Krücker, A. Viswanathan, J. Borgert, N. Glossop, Y. Yang, and B. J. Wood, “An electro-magnetically tracked laparoscopic ultrasound for multi-modality minimally invasive surgery,” *International Congress Series*, vol. 1281, pp. 746–751, 2005.
- [107] H. Lamecker, T. Lange, and M. Seebass, “A statistical shape model for the liver,” in *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2002*, T. Dohi and R. Kikinis, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 421–427.
- [108] H. Lamecker, T. Lange, and M. Seebaß, “Segmentation of the liver using a 3d statistical shape model,” Berlin, Germany, Tech. Rep., 2004.
- [109] L. Landau, E. Lifshitz, and J. Sykes, *Theory of Elasticity*, ser. Course of theoretical physics. Pergamon Press, 1989.

- [110] P. Landreau, A. Drouillard, G. Launoy, P. Ortega-Deballon, V. Jooste, C. Lepage, J. Faivre, O. Facy, and A.-M. Bouvier, "Incidence and survival in late liver metastases of colorectal cancer," *Journal of Gastroenterology and Hepatology*, vol. 30, no. 1, pp. 82–85, 2015.
- [111] Y. Lanir, "Constitutive equations for fibrous connective tissues," *Journal of Biomechanics*, vol. 16, no. 1, pp. 1–12, 1983.
- [112] F. Largillière, E. Coevoet, M. Sanz-Lopez, L. Grisoni, and C. Duriez, "Stiffness rendering on soft tangible devices controlled through inverse fem simulation," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2016, pp. 5224–5229.
- [113] C. Laurent, "13 - micromechanics of ligaments and tendons," in *Multiscale Biomechanics*, J.-F. Ganghoffer, Ed. Elsevier, 2018, pp. 489–509.
- [114] P. Lei, F. Moeslein, B. J. Wood, and R. Shekhar, "Real-time tracking of liver motion and deformation using a flexible needle," *International Journal of Computer Assisted Radiology and Surgery*, vol. 6, no. 3, pp. 435–446, 2011.
- [115] J. Leven, D. Burschka, R. Kumar, G. Zhang, S. Blumenkranz, X. D. Dai, M. Awad, G. D. Hager, M. Marohn, M. Choti, C. Hasser, and R. H. Taylor, "Davinci canvas: A telerobotic surgical system with integrated, robot-assisted, laparoscopic ultrasound capability," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2005*, J. S. Duncan and G. Gerig, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 811–818.
- [116] K. Levenberg, "A method for the solution of certain non-linear problems in least squares," *Quarterly of Applied Mathematics*, vol. 2, no. 2, pp. 164–168, 1944.
- [117] G. Li, J. Gil, A. Kanamori, and S. L.-Y. Woo, "A validated three-dimensional computational model of a human knee joint," *Journal of Biomechanical Engineering*, vol. 121, no. 6, pp. 657–662, 12 1999.
- [118] X. P. Li, D. C. Xu, H. Y. Tan, and C. L. Li, "Anatomical study on the morphology and blood supply of the falciform ligament and its clinical significance," *Surgical and Radiologic Anatomy*, vol. 26, no. 2, pp. 106–109, Apr 2004.
- [119] G. Limbert and J. Middleton, "A transversely isotropic viscohyperelastic material: Application to the modeling of biological soft connective tissues," *International Journal of Solids and Structures*, vol. 41, no. 15, pp. 4237–4260, 2004.
- [120] G. Limbert, J. Middleton, and M. Taylor, "Finite element analysis of the human acl subjected to passive anterior tibial loads," *Computer Methods in Biomechanics and Biomedical Engineering*, vol. 7, no. 1, pp. 1–8, 2004.

- [121] G. Limbert, M. Taylor, and J. Middleton, “Three-dimensional finite element modelling of the human acl: simulation of passive knee flexion with a stressed and stress-free acl,” *Journal of Biomechanics*, vol. 37, no. 11, pp. 1723–1731, 2004.
- [122] J. Lin, N. T. Clancy, and D. S. Elson, “An endoscopic structured light system using multispectral detection,” *International Journal of Computer Assisted Radiology and Surgery*, vol. 10, no. 12, pp. 1941–1950, 2015.
- [123] J. Lin, N. T. Clancy, D. Stoyanov, and D. S. Elson, “Tissue surface reconstruction aided by local normal information using a self-calibrated endoscopic structured light system,” in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, N. Navab, J. Hornegger, W. M. Wells, and A. Frangi, Eds. Cham: Springer International Publishing, 2015, pp. 405–412.
- [124] Q. Lin, R. Yang, K. Cai, P. Guan, W. Xiao, and X. Wu, “Strategy for accurate liver intervention by an optical tracking system,” *Biomed. Opt. Express*, vol. 6, no. 9, pp. 3287–3302, Sep 2015.
- [125] K. Lister, Z. Gao, and J. P. Desai, “Development of in vivo constitutive models for liver: Application to surgical simulation,” *Annals of Biomedical Engineering*, vol. 39, no. 3, pp. 1060–1073, 2011.
- [126] Y.-C. Lu, “Probabilistic analysis of the material and shape properties for human liver,” Ph.D. dissertation, 2014.
- [127] Y.-C. Lu, A. R. Kemper, S. Gayzik, C. D. Untaroiu, and P. Beillas, “Statistical modeling of human liver incorporating the variations in shape, size, and material properties,” in *57th Stapp Car Crash Conference*, vol. 57, Nov 2013, pp. 285–311.
- [128] Y.-C. Lu and C. D. Untaroiu, “Statistical shape analysis of clavicular cortical bone with applications to the development of mean and boundary shape models,” *Computer Methods and Programs in Biomedicine*, vol. 111, no. 3, pp. 613–628, 2013.
- [129] Y.-C. Lu and C. D. Untaroiu, “A statistical geometrical description of the human liver for probabilistic occupant models,” *Journal of Biomechanics*, vol. 47, no. 15, pp. 3681–3688, 2014.
- [130] A. E. Lujan, E. W. Larsen, J. M. Balter, and R. K. Ten Haken, “A method for incorporating organ motion due to breathing into 3d dose calculations,” *Medical Physics*, vol. 26, no. 5, pp. 715–720, 1999.
- [131] H. Luo, D. Yin, S. Zhang, D. Xiao, B. He, F. Meng, Y. Zhang, W. Cai, S. He, W. Zhang, Q. Hu, H. Guo, S. Liang, S. Zhou, S. Liu, L. Sun, X. Guo, C. Fang, L. Liu, and F. Jia, “Augmented reality navigation for liver resection with a stereoscopic laparoscope,” *Computer Methods and Programs in Biomedicine*, vol. 187, 2020.

- [132] S. A. Maas, B. J. Ellis, G. A. Ateshian, and J. A. Weiss, “Febio: Finite elements for biomechanics,” *Journal of Biomechanical Engineering*, vol. 134, no. 1, Feb 2012.
- [133] N. Mahmoud, Ó. G. Grasa, S. A. Nicolau, C. Doignon, L. Soler, J. Marescaux, and J. M. M. Montiel, “On-patient see-through augmented reality based on visual slam,” *International Journal of Computer Assisted Radiology and Surgery*, vol. 12, no. 1, pp. 1–11, 2017.
- [134] L. Maier-Hein, S. A. Müller, F. Pianka, A. Seitel, B. P. Müller-Stich, C. N. Gutt, U. Rietdorf, G. Richter, H.-P. Meinzer, B. M. Schmied, and I. Wolf, “In-vitro evaluation of a novel needle-based soft tissue navigation system with a respiratory liver motion simulator,” in *Medical Imaging 2007: Visualization and Image-Guided Procedures*, K. R. Cleary and M. I. Miga, Eds., vol. 6509, International Society for Optics and Photonics. SPIE, 2007, pp. 412–423.
- [135] L. Maier-Hein, S. A. Müller, F. Pianka, S. Wörz, B. P. Müller-Stich, A. Seitel, K. Rohr, H.-P. Meinzer, B. M. Schmied, and I. Wolf, “Respiratory motion compensation for ct-guided interventions in the liver,” *Computer Aided Surgery*, vol. 13, no. 3, pp. 125–138, 2008.
- [136] L. Maier-Hein, A. Tekbas, A. Seitel, F. Pianka, S. A. Müller, S. Satz, S. Schawo, B. Radeleff, R. Tetzlaff, A. M. Franz, B. P. Müller-Stich, I. Wolf, H.-U. Kauczor, B. M. Schmied, and H.-P. Meinzer, “In vivo accuracy assessment of a needle-based navigation system for ct-guided radiofrequency ablation of the liver,” *Medical Physics*, vol. 35, no. 12, pp. 5385–5396, 2008.
- [137] S. Marchesseau, H. Delingette, M. Sermesant, R. Cabrera-Lozoya, C. Tobon-Gomez, P. Moireau, R. M. Figueras i Ventura, K. Lekadir, A. Hernandez, M. Garreau, E. Donal, C. Leclercq, S. G. Duckett, K. Rhode, C. A. Rinaldi, A. F. Frangi, R. Razavi, D. Chapelle, and N. Ayache, “Personalization of a cardiac electromechanical model using reduced order unscented kalman filtering from regional volumes,” *Medical Image Analysis*, vol. 17, no. 7, pp. 816–829, 2013.
- [138] S. Marchesseau, S. Chatelin, and H. Delingette, “Chapter 11 - nonlinear biomechanical model of the liver,” in *Biomechanics of Living Organs*, ser. Translational Epigenetics, Y. Payan and J. Ohayon, Eds. Oxford: Academic Press, 2017, vol. 1, pp. 243–265.
- [139] S. Marchesseau, T. Heimann, S. Chatelin, R. Willinger, and H. Delingette, “Fast porous visco-hyperelastic soft tissue model for surgery simulation: Application to liver surgery,” *Progress in Biophysics and Molecular Biology*, vol. 103, no. 2, pp. 185–196, 2010.
- [140] Y. Matsuoka, E. Kumamoto, M. Sugimoto, T. Azuma, A. Saito, T. Shibasaki, and K. Kuroda, “Integrated mr-laparoscopy system with respiratory synchronization for minimally invasive liver surgery,” *Journal of Hepato-Biliary-Pancreatic Sciences*, vol. 17, no. 5, pp. 622–628, 2010.

- [141] S. Mersmann, M. Müller, A. Seitel, F. Arnegger, R. Tetzlaff, J. Dinkel, M. Baumhauer, B. Schmied, H.-P. Meinzer, and L. Maier-Hein, “Time-of-flight camera technique for augmented reality in computer-assisted interventions,” in *Medical Imaging 2011: Visualization, Image-Guided Procedures, and Modeling*, K. H. Wong and D. R. H. III, Eds., vol. 7964, International Society for Optics and Photonics. SPIE, 2011, pp. 700–708.
- [142] C. T. Metz, S. Klein, M. Schaap, T. van Walsum, and W. J. Niessen, “Nonrigid registration of dynamic medical imaging data using nd+t b-splines and a groupwise optimization approach,” *Medical Image Analysis*, vol. 15, no. 2, pp. 238–249, 2011.
- [143] K. Miller, “Constitutive modelling of abdominal organs,” *Journal of Biomechanics*, vol. 33, no. 3, pp. 367–373, 2000.
- [144] M. Moerland, A. van den Bergh, R. Bhagwandien, W. Janssen, C. Bakker, J. Lagendijk, and J. Battermann, “The influence of respiration induced motion of the kidneys on the accuracy of radiotherapy treatment planning, a magnetic resonance imaging study,” *Radiotherapy and Oncology*, vol. 30, no. 2, pp. 150–154, 1994.
- [145] P. Moireau, C. Bertoglio, N. Xiao, C. A. Figueroa, C. Taylor, D. Chapelle, and J.-F. Gerbeau, “Sequential identification of boundary support parameters in a fluid-structure vascular model using patient image data,” *Biomechanics and Modeling in Mechanobiology*, vol. 12, no. 3, pp. 475–496, Jul 2012.
- [146] P. Moireau and D. Chapelle, “Reduced-order unscented kalman filtering with application to parameter identification in large-dimensional systems,” *ESAIM: COCV*, vol. 17, no. 2, pp. 380–405, 2011.
- [147] T. Mommersteeg, R. Huiskes, L. Blankevoort, J. G. Kooloos, J. M. Kauer, and P. G. Maathuis, “A global verification study of a quasi-static knee model with multi-bundle ligaments,” *Journal of Biomechanics*, vol. 29, no. 12, pp. 1659–1664, 1996.
- [148] S. Morikawa, T. Inubushi, Y. Kurumi, S. Naka, K. Sato, T. Tani, H. A. Haque, J.-i. Tokuda, and N. Hata, “New assistive devices for mr-guided microwave thermocoagulation of liver tumors,” *Academic Radiology*, vol. 10, no. 2, pp. 180–188, 2003.
- [149] P. Mountney, J. Fallert, S. Nicolau, L. Soler, and P. W. Mewes, “An augmented reality framework for soft tissue surgery,” in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2014*, P. Golland, N. Hata, C. Barillot, J. Hornegger, and R. Howe, Eds. Cham: Springer International Publishing, 2014, pp. 423–431.
- [150] P. Mountney and G.-Z. Yang, “Motion compensated slam for image guided surgery,” in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2010*, T. Jiang, N. Navab, J. P. W. Pluim, and M. A. Viergever, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 496–504.

- [151] M. Nakamoto, H. Hirayama, Y. Sato, K. Konishi, Y. Kakeji, M. Hashizume, and S. Tamura, "Recovery of respiratory motion and deformation of the liver using laparoscopic freehand 3d ultrasound system," *Medical Image Analysis*, vol. 11, no. 5, pp. 429–442, 2007.
- [152] M. Nakamoto, Y. Sato, M. Miyamoto, Y. Nakamjima, K. Konishi, M. Shimada, M. Hashizume, and S. Tamura, "3d ultrasound system using a magneto-optic hybrid tracker for augmented reality visualization in laparoscopic liver surgery," in *Medical Image Computing and Computer-Assisted Intervention — MICCAI 2002*, T. Dohi and R. Kikinis, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 148–155.
- [153] A. Nava, E. Mazza, M. Furrer, P. Villiger, and W. H. Reinhart, "In vivo mechanical characterization of human liver," *Medical Image Analysis*, vol. 12, no. 2, pp. 203–216, 2008.
- [154] A. Nava, E. Mazza, O. Haefner, and M. Bajka, "Experimental observation and modelling of preconditioning in soft biological tissues," in *Medical Simulation*, S. Cotin and D. Metaxas, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 1–8.
- [155] A. Nava, E. Mazza, F. Kleinermann, N. J. Avis, J. McClure, and M. Bajka, "Evaluation of the mechanical properties of human liver and kidney through aspiration experiments," *Technol. Health Care*, vol. 12, no. 3, pp. 269–280, Aug 2004.
- [156] M. Nesme, Y. Payan, and F. Faure, "Efficient, physically plausible finite elements," in *EG Short Presentations*, J. Dingliana and F. Ganovelli, Eds. The Eurographics Association, 2005, pp. 77–80.
- [157] F. Netter, *Atlas of human anatomy: 5th edition*. Saunders/Elsevier, 2010.
- [158] H. T. Nia, H. Liu, G. Seano, M. Datta, D. Jones, N. Rahbari, J. Incio, V. P. Chauhan, K. Jung, J. D. Martin, V. Askoxylakis, T. P. Padera, D. Fukumura, Y. Boucher, F. J. Hornicek, A. J. Grodzinsky, J. W. Baish, L. L. Munn, and R. K. Jain, "Solid stress and elastic energy as measures of tumour mechanopathology," *Nature Biomedical Engineering*, vol. 1, pp. 1–11, 2016.
- [159] S. Nicolau, X. Pennec, L. Soler, X. Buy, A. Gangi, N. Ayache, and J. Marescaux, "An augmented reality system for liver thermal ablation: Design and evaluation on clinical cases," *Medical Image Analysis*, vol. 13, no. 3, pp. 494–506, 2009.
- [160] S. Nicolle, P. Vezin, and J.-F. Paliarne, "A strain-hardening bi-power law for the nonlinear behaviour of biological soft tissues," *Journal of Biomechanics*, vol. 43, no. 5, pp. 927–932, 2010.
- [161] S. Niroomandi, D. González, I. Alfaro, F. W. 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.

- [162] S. Nishiyama, Y. Kuroda, and H. Takemura, “Stiffness matrix representation of hyper-elasticity for surgical simulation and navigation,” in *37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2015, pp. 905–908.
- [163] T. Okada, R. Shimada, Y. Sato, M. Hori, K. Yokota, M. Nakamoto, Y.-W. Chen, H. Nakamura, and S. Tamura, “Automated segmentation of the liver from 3d ct images using probabilistic atlas and multi-level statistical shape model,” in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2007*, N. Ayache, S. Ourselin, and A. Maeder, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 86–93.
- [164] T. Okada, K. Yokota, M. Hori, M. Nakamoto, H. Nakamura, and Y. Sato, “Construction of hierarchical multi-organ statistical atlases and their application to multi-organ segmentation from ct images,” in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2008*, D. Metaxas, L. Axel, G. Fichtinger, and G. Székely, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 502–509.
- [165] O. Oktay, L. Zhang, T. Mansi, P. Mountney, P. Mewes, S. Nicolau, L. Soler, and C. Chefd’hotel, “Biomechanically driven registration of pre- to intra-operative 3d images for laparoscopic surgery,” in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2013*, K. Mori, I. Sakuma, Y. Sato, C. Barillot, and N. Navab, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 1–9.
- [166] R. E. Ong, S. D. Herrell, M. I. Miga, and G. R. L., “A kidney deformation model for use in non-rigid registration during image-guided surgery,” in *Medical Imaging 2008: Visualization, Image-Guided Procedures, and Modeling*, M. I. Miga and K. R. Cleary, Eds., vol. 6918, International Society for Optics and Photonics. SPIE, 2008, pp. 317–325.
- [167] S. T. Orcutt and D. A. Anaya, “Liver resection and surgical strategies for management of primary liver cancer,” *Cancer control : journal of the Moffitt Cancer Center*, vol. 25, no. 1, 2018.
- [168] M. U. Ozcan, S. Ocal, C. Basdogan, G. Dogusoy, and Y. Tokat, “Characterization of frequency-dependent material properties of human liver and its pathologies using an impact hammer,” *Medical Image Analysis*, vol. 15, no. 1, pp. 45–52, 2011.
- [169] E. Ozkan and O. Goksel, “Compliance boundary conditions for simulating deformations in a limited target region,” in *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Aug 2015, pp. 929–932.
- [170] E. Ozkan and O. Goksel, “Compliance boundary conditions for patient-specific deformation simulation using the finite element method,” *Biomedical Physics & Engineering Express*, vol. 4, no. 2, jan 2018.

- [171] H. Park, P. H. Bland, and C. R. Meyer, "Construction of an abdominal probabilistic atlas and its application in segmentation," *IEEE Transactions on Medical Imaging*, vol. 22, no. 4, pp. 483–492, 2003.
- [172] H. Park, A. Hero, P. Bland, M. Kessler, J. Seo, and C. Meyer, "Construction of abdominal probabilistic atlases and their value in segmentation of normal organs in abdominal ct scans," *IEICE Transactions on Information and Systems*, vol. E93.D, no. 8, pp. 2291–2301, 2010.
- [173] E. Peña, M. A. Martínez, B. Calvo, D. Palanca, and M. Doblaré, "A finite element simulation of the effect of graft stiffness and graft tensioning in acl reconstruction," *Clinical Biomechanics*, vol. 20, no. 6, pp. 636–644, 2005.
- [174] E. Peña, J. A. Peña, and M. Doblaré, "On modelling nonlinear viscoelastic effects in ligaments," *Journal of Biomechanics*, vol. 41, no. 12, pp. 2659–2666, 2008.
- [175] J. Penne, K. Höller, M. Stürmer, T. Schrauder, A. Schneider, R. Engelbrecht, H. Feußner, B. Schmauss, and J. Hornegger, "Time-of-flight 3-d endoscopy," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2009*, G.-Z. Yang, D. Hawkes, D. Rueckert, A. Noble, and C. Taylor, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 467–474.
- [176] J. Pérez de Frutos, E. F. Hofstad, O. V. Solberg, G. A. Tangen, F. Lindseth, T. Langø, O. J. Elle, and R. Mårvik, "Laboratory test of single landmark registration method for ultrasound-based navigation in laparoscopy using an open-source platform," *International Journal of Computer Assisted Radiology and Surgery*, vol. 13, no. 12, pp. 1927–1936, 2018.
- [177] M. Peterhans, A. vom Berg, B. Dagon, D. Inderbitzin, C. Baur, D. Candinas, and S. Weber, "A navigation system for open liver surgery: design, workflow and first clinical applications," *The International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 7, no. 1, pp. 7–16, 2011.
- [178] I. Peterlik, H. Courtecuisse, C. Duriez, and S. Cotin, "Model-based identification of anatomical boundary conditions in living tissues," in *Information Processing in Computer-Assisted Interventions*, D. Stoyanov, D. L. Collins, I. Sakuma, P. Abolmaesumi, and P. Jannin, Eds. Cham: Springer International Publishing, 2014, pp. 196–205.
- [179] I. Peterlik, C. Duriez, and S. Cotin, "Modeling and real-time simulation of a vascularized liver tissue," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2012*, N. Ayache, H. Delingette, P. Golland, and K. Mori, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 50–57.
- [180] I. Peterlik, N. Haouchine, L. Ručka, and S. Cotin, "Image-driven stochastic identification of boundary conditions for predictive simulation," in *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2017*, M. Descoteaux, L. Maier-Hein,

- A. Franz, P. Jannin, D. L. Collins, and S. Duchesne, Eds. Cham: Springer International Publishing, 2017, pp. 548–556.
- [181] I. Peterlik, S. Nikolaev, N. Schulmann, R. Trivisonne, and S. Cotin. Optimus pluign: Advanced methods of state estimation and parameter identification in SOFA. URL: <https://github.com/sofa-framework/Optimus>
- [182] D. T. Pham, “Dimension, predictability and reduced rank kalman filtering in data assimilation,” in *Proceedings of Third Bilateral French-Russian Conf.: Predictability of Atmospheric and Oceanic Circulations*, 1997, pp. 61–72.
- [183] G. Picinbono, H. Delingette, and N. Ayache, “Non-linear anisotropic elasticity for real-time surgery simulation,” *Graphical Models*, vol. 65, no. 5, pp. 305–321, 2003.
- [184] D. P. Pioletti, L. R. Rakotomanana, J.-F. Benvenuti, and P.-F. Leyvraz, “Viscoelastic constitutive law in large deformations: application to human knee ligaments and tendons,” *Journal of Biomechanics*, vol. 31, no. 8, pp. 753–757, 1998.
- [185] D. Pioletti, L. Rakotomanana, J.-F. Benvenuti, and P.-F. Leyvraz, “Viscoelastic constitutive law in large deformations: application to human knee ligaments and tendons,” *Journal of Biomechanics*, vol. 31, no. 8, pp. 753–757, 1998.
- [186] R. Plantefève, “Augmented reality and numerical simulations for hepatic tumors resection,” Ph.D. dissertation, 2016.
- [187] R. Plantefève, I. Peterlik, H. Courtecuisse, R. Trivisonne, J.-P. Radoux, and S. Cotin, “Atlas-based transfer of boundary conditions for biomechanical simulation,” in *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2014*. Springer International Publishing, 2014, pp. 33–40.
- [188] 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.
- [189] G. A. Prevost, B. Eigl, I. Paolucci, T. Rudolph, M. Peterhans, S. Weber, G. Beldi, D. Candinas, and A. Lachenmayer, “Efficiency, accuracy and clinical applicability of a new image-guided surgery system in 3d laparoscopic liver surgery.” *Journal of Gastrointestinal Surgery*, 2019.
- [190] T. C. Project. CGAL user and reference manual. URL: <https://doc.cgal.org/5.2.2/Manual/packages.html>
- [191] J. S. Przemieniecki, *Theory of Matrix Structural Analysis*. New York: Dover Publications Inc., 1968.
- [192] G. A. Puerto-Souza and G. L. Mariottini, “Toward long-term and accurate augmented-reality display for minimally-invasive surgery,” in *2013 IEEE International Conference on Robotics and Automation*, May 2013, pp. 5384–5389.

- [193] M. A. Puso and J. A. Weiss, "Finite element implementation of anisotropic quasi-linear viscoelasticity using a discrete spectrum approximation," *Journal of Biomechanical Engineering*, vol. 120, no. 1, pp. 62–70, 02 1998.
- [194] A. Qiu, T. Brown, B. Fischl, J. Ma, and M. I. Miller, "Atlas generation for subcortical and ventricular structures with its applications in shape analysis," *IEEE Transactions on Image Processing*, vol. 19, no. 6, pp. 1539–1547, 2010.
- [195] K. M. Quapp and J. A. Weiss, "Material characterization of human medial collateral ligament," *Journal of Biomechanical Engineering*, vol. 120, no. 6, pp. 757–763, 12 1998.
- [196] S. Raghunathan, D. Evans, and J. L. Sparks, "Poroviscoelastic modeling of liver biomechanical response in unconfined compression," *Annals of Biomedical Engineering*, vol. 38, no. 5, pp. 1789–1800, 2010.
- [197] C. R. Ramsey, D. Scaperoth, D. Arwood, and A. L. Oliver, "Clinical efficacy of respiratory gated conformal radiation therapy," *Medical Dosimetry*, vol. 24, no. 2, pp. 115–119, 1999.
- [198] T. P. Rauth, P. Q. Bao, R. L. Galloway, J. Bieszczad, E. M. Friets, D. A. Knaus, D. B. Kynor, and A. J. Herline, "Laparoscopic surface scanning and subsurface targeting: implications for image-guided laparoscopic liver surgery," *Surgery*, vol. 142, no. 2, pp. 207–214, 2007.
- [199] A. Raza and G. K. Sood, "Hepatocellular carcinoma review: current treatment, and evidence-based medicine," *World journal of gastroenterology*, vol. 20, no. 15, pp. 4115–4127, Apr 2014.
- [200] S. P. Reese, B. J. Ellis, and J. A. Weiss, "Multiscale modeling of ligaments and tendons," in *Multiscale Computer Modeling in Biomechanics and Biomedical Engineering*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 103–147.
- [201] M. Reyes, M. A. Gonzalez Ballester, Z. Li, N. Kozic, S. Chin, R. M. Summers, and L. M. George, "Anatomical variability of organs via principal factor analysis from the construction of an abdominal probabilistic atlas," in *2009 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, 2009, pp. 682–685.
- [202] E. Roan and K. Vemaganti, "The nonlinear material properties of liver tissue determined from no-slip uniaxial compression experiments," *Journal of Biomechanical Engineering*, vol. 129, no. 3, pp. 450–456, 11 2006.
- [203] J. M. Rodriguez Lelis, J. S. Médez Aguirre, J. A. Arellano Cabrera, J. Navarro Torres, M. Vargas Treviño, and A. Pliego, "On the mechanical design of cruciate ligaments based on polymeric ropes for knee prosthesis," *Revista Mexicana de Ingenieria biomédica*, vol. 32, no. 1, pp. 25–31, 07 2011.

- [204] J. Rosen, J. D. Brown, S. De, M. Sinanan, and B. Hannaford, “Biomechanical properties of abdominal organs in vivo and postmortem under compression loads,” *Journal of Biomechanical Engineering*, vol. 130, no. 2, 04 2008.
- [205] S. Rouzé, B. de Latour, E. Flécher, J. Guihaire, M. Castro, R. Corre, P. Haigron, and J.-P. Verhoye, “Small pulmonary nodule localization with cone beam computed tomography during video-assisted thoracic surgery: a feasibility study,” *Interactive CardioVascular and Thoracic Surgery*, vol. 22, no. 6, pp. 705–711, Feb 2016.
- [206] R. Y. Rubinstein and D. P. Kroese, *Simulation and the Monte Carlo Method, 3rd Edition*. Wiley, 2016.
- [207] T. P. Sarac, K. Carnevale, N. Smedira, E. Tanquilut, P. Augustinos, A. Patel, T. Naska, D. Clair, and K. Ouriel, “In vivo and mechanical properties of peritoneum/fascia as a novel arterial substitute,” *Journal of Vascular Surgery*, vol. 41, no. 3, pp. 490–497, Mar 2005.
- [208] F. Sato, Y. Yamamoto, D. Ito, J. Antona-Makoshi, K. Kamiji, and T. Yasuki, “Hyper-viscoelastic response of perfused liver under dynamic compression and estimation of tissue strain thresholds with a liver finite element model,” in *2013 International Research Council on the Biomechanics of Injury (IRCOBI) Conference Proceedings*, 01 2013, pp. 736–750.
- [209] K. Sato, S. Morikawa, T. Inubushi, Y. Kurumi, S. Naka, H. A. Haque, K. Demura, and T. Tani, “Alternate biplanar mr navigation for microwave ablation of liver tumors,” *Magnetic Resonance in Medical Sciences*, vol. 4, no. 2, pp. 89–94, 2005.
- [210] C. Schneider, C. Nguan, M. Longpre, R. Rohling, and S. Salcudean, “Motion of the kidney between preoperative and intraoperative positioning,” *IEEE Transactions on Biomedical Engineering*, vol. 60, no. 6, pp. 1619–1627, 2013.
- [211] N. Schulmann, S. Cotin, and I. Peterlik, “The effect of discretization on parameter identification. application to patient-specific simulations,” in *Computer Methods, Imaging and Visualization in Biomechanics and Biomedical Engineering*, G. A. Ateshian, K. M. Myers, and J. M. R. S. Tavares, Eds. Cham: Springer International Publishing, 2020, pp. 237–247.
- [212] J.-M. Schwartz, M. Denninger, D. Rancourt, C. Moisan, and D. Laurendeau, “Modelling liver tissue properties using a non-linear visco-elastic model for surgery simulation,” *Medical Image Analysis*, vol. 9, no. 2, pp. 103–112, 2005.
- [213] O. Shahin, A. Beširević, M. Kleemann, and A. Schlaefer, “Ultrasound-based tumor movement compensation during navigated laparoscopic liver interventions,” *Surgical Endoscopy*, vol. 28, no. 5, pp. 1734–1741, 2014.

- [214] M. Shahzad, A. Kamran, M. Z. Siddiqui, and M. Farhan, “Mechanical characterization and fe modelling of a hyperelastic material,” *Materials Research*, vol. 18, no. 5, pp. 918–924, Oct 2015.
- [215] R. Shekhar, O. Dandekar, V. Bhat, M. Philip, P. Lei, C. Godinez, E. Sutton, I. George, S. Kavic, R. Mezrich, and A. Park, “Live augmented reality: a new visualization method for laparoscopic surgery using continuous volumetric computed tomography,” *Surgical Endoscopy*, vol. 24, no. 8, pp. 1976–1985, 2010.
- [216] J. R. Shewchuk, “An introduction to the conjugate gradient method without the agonizing pain,” USA, Tech. Rep., 1994.
- [217] K. Shmuel, “Estimates for some computational techniques in linear algebra,” *Mathematics of Computation* 20, pp. 369–378, 1966.
- [218] D. Simon, “The discrete-time kalman filter,” in *Optimal State Estimation: Kalman, H_∞ , and Nonlinear Approaches*. John Wiley & Sons, Ltd, 2006, ch. 5, pp. 121–148.
- [219] L. Soler, A. Hostettler, V. Agnus, A. Charnoz, and J.-B. Fasquel, “3d image reconstruction for comparison of algorithm database: A patient-specific anatomical and medical image database,” 2012.
- [220] Y. Song, R. E. Debski, V. Musahl, M. Thomas, and S. L.-Y. Woo, “A three-dimensional finite element model of the human anterior cruciate ligament: a computational analysis with experimental validation,” *Journal of Biomechanics*, vol. 37, no. 3, pp. 383–390, 2004.
- [221] J. L. Sparks and R. B. Dupaix, “Constitutive modeling of rate-dependent stress–strain behavior of human liver in blunt impact loading,” *Annals of Biomedical Engineering*, vol. 36, no. 11, pp. 1883–1892, 2008.
- [222] D. Stalling, M. Westerhoff, and H.-C. Hege, “Amira: A highly interactive system for visual data analysis,” in *Visualization Handbook*, C. D. Hansen and C. R. Johnson, Eds. Burlington: Butterworth-Heinemann, 2005, pp. 749–767.
- [223] G. Strang, “A proposal for toeplitz matrix calculations,” *Studies in Applied Mathematics*, vol. 74, no. 2, pp. 171–176, 1986.
- [224] A. Tamura, K. Omori, K. Miki, J. B. Lee, K. H. Yang, and A. I. King, “Mechanical characterization of porcine abdominal organs.” *Stapp car crash journal*, vol. 46, pp. 55–69, 2002.
- [225] W. Tan, W. Ge, Y. Hang, S. Wu, S. Liu, and M. Liu, “Computer assisted system for precise lung surgery based on medical image computing and mixed reality,” *Health Information Science and Systems*, vol. 6, no. 1, pp. 1–10, 2018.

- [226] Z. A. Taylor, O. Comas, M. Cheng, J. Passenger, D. J. Hawkes, D. Atkinson, and S. Ourselin, "On modelling of anisotropic viscoelasticity for soft tissue simulation: Numerical solution and gpu execution," *Medical Image Analysis*, vol. 13, no. 2, pp. 234–244, 2009.
- [227] J. Tokuda, S. Morikawa, H. A. Haque, T. Tsukamoto, K. Matsumiya, H. Liao, K. Masamune, and T. Dohi, "New 4-d imaging for real-time intraoperative mri: Adaptive 4-d scan," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2006*, R. Larsen, M. Nielsen, and J. Sporring, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 454–461.
- [228] S. Tomoshige, E. Oost, A. Shimizu, H. Watanabe, and S. Nawano, "A conditional statistical shape model with integrated error estimation of the conditions; application to liver segmentation in non-contrast ct images," *Medical Image Analysis*, vol. 18, no. 1, pp. 130–143, 2014.
- [229] K. Tong, Y. Wu, and F. Zhou, "Recent advances in small object detection based on deep learning: A review," *Image and Vision Computing*, vol. 97, p. 103910, 2020.
- [230] N. Tsutsumi, M. Tomikawa, M. Uemura, T. Akahoshi, Y. Nagao, K. Konishi, S. Ieiri, J. Hong, Y. Maehara, and M. Hashizume, "Image-guided laparoscopic surgery in an open mri operating theater," *Surgical Endoscopy*, vol. 27, no. 6, pp. 2178–2184, 2013.
- [231] J. C. Tung, J. M. Barnes, S. R. Desai, C. Sistrunk, M. W. Conklin, P. Schedin, K. W. Eliceiri, P. J. Keely, V. L. Seewaldt, and W. V. M., "Tumor mechanics and metabolic dysfunction," *Free Radical Biology and Medicine*, vol. 79, pp. 269–280, 2015.
- [232] M. Uetani, T. Tateyama, S. Kohara, H. Tanaka, X.-H. Han, S. Kanasaki, A. Furukawa, and Y.-W. Chen, "Statistical shape model of the liver and its application to computer-aided diagnosis of liver cirrhosis," *Electrical Engineering in Japan*, vol. 190, no. 4, pp. 37–45, 2015.
- [233] S. Umale, C. Deck, N. Bourdet, P. Dhumane, L. Soler, J. Marescaux, and R. Willinger, "Experimental mechanical characterization of abdominal organs: liver, kidney & spleen," *Journal of the Mechanical Behavior of Biomedical Materials*, vol. 17, pp. 22–33, 2013.
- [234] C. D. Untaroiu and Y.-C. Lu, "Material characterization of liver parenchyma using specimen-specific finite element models," *Journal of the Mechanical Behavior of Biomedical Materials*, vol. 26, pp. 11–22, 2013.
- [235] C. D. Untaroiu, Y.-C. Lu, S. K. Siripurapu, and A. R. Kemper, "Modeling the biomechanical and injury response of human liver parenchyma under tensile loading," *Journal of the Mechanical Behavior of Biomedical Materials*, vol. 41, pp. 280–291, 2015.
- [236] M. Verlaan and A. W. Heemink, "Tidal flow forecasting using reduced rank square root filters," *Stochastic Hydrology and Hydraulics*, vol. 11, no. 5, pp. 349–368, Oct 1997.

- [237] S. Vijayan, S. Klein, E. Hofstad, F. Lindseth, B. Ystgaard, and T. Langø, “Validation of a non-rigid registration method for motion compensation in 4d ultrasound of the liver,” Apr 2013, pp. 792–795.
- [238] V. Walimbe and R. Shekhar, “Automatic elastic image registration by interpolation of 3d rotations and translations from discrete rigid-body transformations,” *Medical Image Analysis*, vol. 10, no. 6, pp. 899–914, 2006.
- [239] C. Wan, Z. Hao, and S. Wen, “The effect of the variation in acl constitutive model on joint kinematics and biomechanics under different loads: A finite element study,” *Journal of Biomechanical Engineering*, vol. 135, no. 4, 04 2013.
- [240] A. J. Wathen, “Preconditioning,” *Acta Numerica*, vol. 24, pp. 329–376, 2015.
- [241] D. Weed, L. G. Maqueda, M. A. Brown, and A. A. Shabana, “A multibody/finite element non-linear formulation of a two-ligament knee joint,” in *Proceedings of ASME 2008 International Mechanical Engineering Congress and Exposition*, ser. ASME International Mechanical Engineering Congress and Exposition, vol. Volume 2: Biomedical and Biotechnology Engineering, 10 2008, pp. 407–416.
- [242] J. A. Weiss, J. C. Gardiner, and C. Bonifasi-Lista, “Ligament material behavior is non-linear, viscoelastic and rate-independent under shear loading,” *Journal of Biomechanics*, vol. 35, no. 7, pp. 943–950, 2002.
- [243] J. A. Weiss and B. J. Maakestad, “Permeability of human medial collateral ligament in compression transverse to the collagen fiber direction,” *Journal of Biomechanics*, vol. 39, no. 2, pp. 276–283, 2006.
- [244] J. A. Weiss, B. N. Maker, and S. Govindjee, “Finite element implementation of incompressible, transversely isotropic hyperelasticity,” *Computer Methods in Applied Mechanics and Engineering*, vol. 135, no. 1, pp. 107–128, 1996.
- [245] G. Welch and G. Bishop, “An introduction to the kalman filter,” Chapel Hill, NC, USA, Tech. Rep., 1995.
- [246] E. J. White, E. M. Cunnane, M. McMahon, M. T. Walsh, J. C. Coffey, and L. O’Sullivan, “Mechanical characterisation of porcine non-intestinal colorectal tissues for innovation in surgical instrument design,” *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine*, vol. 232, no. 8, pp. 796–806, 2018.
- [247] D. Wirtz, K. Konstantopoulos, and P. C. Searson, “The physics of cancer: the role of physical interactions and mechanical forces in metastasis,” *Nature Reviews Cancer*, vol. 11, no. 7, pp. 512–522, 2011.
- [248] J. Wismans, F. Veldpaus, J. Janssen, A. Huson, and P. Struben, “A three-dimensional mathematical model of the knee-joint,” *Journal of Biomechanics*, vol. 13, no. 8, pp. 677–685, 1980.

- [249] C. A. Witz, I. A. Montoya-Rodriguez, S. Cho, V. E. Centonze, L. F. Bonewald, and R. S. Schenken, "Composition of the extracellular matrix of the peritoneum," *Journal of the Society for Gynecologic Investigation*, vol. 8, no. 5, pp. 299–304, 2001.
- [250] P. Wriggers, *Nonlinear Finite Element Methods*. Springer-Verlag Berlin Heidelberg, 2008.
- [251] X. Wu, M. S. Downes, T. Goktekin, and F. Tendick, "Adaptive nonlinear finite elements for deformable body simulation using dynamic progressive meshes," *Computer Graphics Forum*, vol. 20, no. 3, pp. 349–358, 2001.
- [252] W. Xiong, S. H. Ong, Q. Tian, G. Xu, J. Zhou, J. Liu, and S. K. Venkatas, "Construction of a linear unbiased diffeomorphic probabilistic liver atlas from ct images," in *2009 16th IEEE International Conference on Image Processing (ICIP)*, 2009, pp. 1773–1776.
- [253] J. Yang, H. Li, D. Campbell, and Y. Jia, "Go-icp: A globally optimal solution to 3d icp point-set registration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 11, pp. 2241–2254, 2016.
- [254] J. Yao and R. M. Summers, "Statistical location model for abdominal organ localization," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2009*, G.-Z. Yang, D. Hawkes, D. Rueckert, A. Noble, and C. Taylor, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 9–17.
- [255] H. M. Yin, L. Z. Sun, G. Wang, and M. W. Vannier, "Modeling of elastic modulus evolution of cirrhotic human liver," *IEEE Transactions on Biomedical Engineering*, vol. 51, no. 10, pp. 1854–1857, 2004.
- [256] A. B. Zavatsky and J. J. O'Connor, "A model of human knee ligaments in the sagittal plane: Part 2: Fibre recruitment under load," *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine*, vol. 206, no. 3, pp. 135–145, 1992.
- [257] X. Zhang, G. Jiang, C. Wu, and S. L.-Y. Woo, "A subject-specific finite element model of the anterior cruciate ligament," in *2008 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2008, pp. 891–894.
- [258] H. Zhong and T. Peters, "A real time hyperelastic tissue model," *Computer Methods in Biomechanics and Biomedical Engineering*, vol. 10, no. 3, pp. 185–193, 2007.



List of published works



- [1] Nikolaev S., Peterlik I., Cotin S., "Stochastic Correction of Boundary Conditions during Liver Surgery", 2018 Colour and Visual Computing Symposium (CVCS), pp. 1-4, 2018, IEEE
- [2] Nikolaev S., Cotin S., "Estimation of boundary conditions for patient-specific liver simulation during augmented surgery", International Journal of Computer Assisted Radiology and Surgery, vol. 15, no. 7, pp. 1107-1115, 2020
- [3] Mendizabal A., Tagliabue E., Hoellinger T., Brunet J.-N., Nikolaev S., Cotin S., "Data-driven simulation for augmented surgery", In: Development and Novel Approaches in Biomechanics and Metamaterials, chap. 5, pp. 71-96, 2020
- [4] Teaniti A., Brunet J.N., Nikolaev S., Wang C., Edwin B., Cotin S., Elle O.J., "Use of stereo-laparoscopic liver surface reconstruction to compensate for pneumoperitoneum deformation through biomechanical modeling", 2020 Virtual Physiological Human Conference (VPH), pp. 1-2, 2020.

Identification et caractérisation des conditions aux limites pour des simulations biomécaniques patient-spécifiques

Résumé

L'objectif de ce travail est trouvé un moyen d'estimer les conditions aux limites du foie. Elles jouent un rôle essentiel dans la capacité de prédiction du modèle biomécanique, mais sont principalement présentées par les ligaments, les vaisseaux et les organes environnants, dont les propriétés sont "spécifiques au patient" et ne peuvent être mesurées fidèlement. Nous proposons de présenter ces conditions comme des ressorts non linéaires et d'estimer ses paramètres. D'abord, nous créons une approximation initiale en utilisant la loi constitutive disponible dans la littérature et un atlas statistique obtenu à partir des modèles avec des ligaments segmentés. Après, nous la corrigeons basée sur le filtrage de Kalman non linéaire, qui assimile les données acquises d'une modalité pendant la chirurgie. Pour évaluation, nous avons réalisé des expériences avec des données synthétiques et réelles. Les résultats montrent une amélioration de la précision pour les cas avec des limites estimées.

Mots-clés: Modélisation spécifique au patient, Simulation numérique, Assimilation de données, Réalité augmentée, Biomécanique, Chirurgie assistée par ordinateur, Atlas statistique

Résumé en anglais

The purpose of the work is to find a way to estimate the boundary conditions of the liver. They play an essential role in forming the predictive capacity of the biomechanical model, but are presented mainly by ligaments, vessels, and surrounding organs, the properties of which are "patient specific" and cannot be measured reliably. We propose to present the boundary conditions as nonlinear springs and estimate their parameters. Firstly, we create a generalized initial approximation using the constitutive law available in the literature and a statistical atlas, obtained from a set of models with segmented ligaments. Then, we correct the approximation based on the nonlinear Kalman filtering approach, which assimilates data obtained from a modality during surgical intervention. To assess the approach, we performed experiments for both synthetic and real data. The results show a certain improvement in simulation accuracy for the cases with estimated boundaries.

Keywords: Patient-specific modeling, Numerical simulation, Data assimilation, Augmented reality, Biomechanics, Computer-aided surgery, Statistical atlas