



**HAL**  
open science

## Data-driven simulation for augmented surgery

Andrea Mendizabal, Eleonora Tagliabue, Tristan Hoellinger, Jean-Nicolas Brunet, Sergei Nikolaev, Stéphane Cotin

► **To cite this version:**

Andrea Mendizabal, Eleonora Tagliabue, Tristan Hoellinger, Jean-Nicolas Brunet, Sergei Nikolaev, et al.. Data-driven simulation for augmented surgery. Bilen Emek Abali; Ivan Giorgio. Developments and Novel Approaches in Biomechanics and Metamaterials, 132, Springer, pp.71-96, 2020, Advanced Structured Materials, 978-3-030-50464-9. 10.1007/978-3-030-50464-9\_5 . hal-02538101

**HAL Id: hal-02538101**

**<https://inria.hal.science/hal-02538101>**

Submitted on 9 Apr 2020

**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.

# Chapter 1

## Data-driven simulation for augmented surgery

Andrea Mendizabal, Eleonora Tagliabue, Tristan Hoellinger, Jean-Nicolas Brunet,  
Sergei Nikolaev, Stéphane Cotin

To build an augmented view of an organ during surgery, it is essential to have a biomechanical model with appropriate material parameters and boundary conditions, able to match patient specific properties. Adaptation to the patient's anatomy is obtained by exploiting the image-rich context specific to our application domain. While information about the organ shape, for instance, can be obtained preoperatively, other patient-specific parameters can only be determined intraoperatively. To this end, we are developing data-driven simulations, which exploit information extracted from a stream of medical images. Such simulations need to run in real-time. To this end we have developed dedicated numerical methods, which allow for real-time computation of finite element simulations.

The general principle consists in combining finite element approaches with Bayesian methods or deep learning techniques, that allow to keep control over the underlying computational model while allowing for inputs from the real world. Based on *a priori* knowledge of the mechanical behavior of the considered organ, we select a constitutive law to model its deformations. The predictive power of such constitutive law highly depends on the knowledge of the material parameters and

---

A. Mendizabal  
Inria and University of Strasbourg, Strasbourg, France  
e-mail: andrea.mendizabal@inria.fr

E. Tagliabue  
Università degli Studi di Verona

T. Hoellinger  
Inria, Strasbourg, France

J-N. Brunet  
Inria, Strasbourg, France

S. Nikolaev  
Inria, Strasbourg, France

S. Cotin (*corresponding author*)  
Inria, Strasbourg, France  
e-mail: stephane.cotin@inria.fr

the boundary conditions. In our first approach, material properties are modeled as stochastic parameters whose probability distributions will be modified in real-time using Kalman filters, given observations extracted from intraoperative data. The second option we propose is to directly learn material parameters and boundary conditions from patient-specific data using deep neural networks. This has been applied to the modeling of liver biomechanics, its real-time simulation, and parametrization to achieve patient-specific augmented reality during surgery.

**Keywords:** Data-driven simulation · Bayesian filtering · Deep neural networks · Finite element method · Augmented reality

## 1.1 Introduction

In computer-aided intervention, the correct alignment of the preoperative images to the intraoperative ones remains a real challenge especially when large deformations are involved. In the context of hepatic surgery for instance, the objective is to accurately locate the internal structures such as tumors and blood vessels (that need to be preserved for the post-operative regeneration of the liver tissue). While the initial position of these structures is known from the preoperative images, their actual position during surgery is often hidden or uncertain. To guide the surgeon, augmented reality techniques are used to enrich visual information through fusion of intraoperative images and a preoperative 3D model of the patient's anatomy. This is usually done by overlaying a virtual representation of the liver built from preoperative images over intraoperative images or through augmented reality glasses. However, surgical manipulations and interactions with the surrounding anatomy can induce significant deformations to the patient's liver. As a consequence, the virtual model of the liver has to account for non-rigid transformations and produce its deformed state in real-time, which is difficult given the complexity of the physical systems needed for accurate biomechanical modeling.

Existing works in this area rely on patient-specific biomechanical models that can provide in-depth motion given surface deformation (14; 26; 75; 3). The Finite Element (FE) method is the preferred one due to its ability to numerically solve the complex partial differential equations that come into play. However, the demanding accuracy of medical applications (e.g. registration of internal structures below 5 *mm* (68)) raise several challenges that are far from being solved.

In augmented surgery, the computational efficiency of the FE method becomes crucial. In the case of augmented hepatic surgery, intraoperative images are acquired at about 20Hz leading to update times of less than 50ms. During this small amount of time, acquisition and processing of the images as well as model update need to take place. As a result, FE computation times should require less than 30 *ms*. If only small deformations take place, achieving such computation times is feasible (40). However, if large non-linear deformations happen, computation times become incompatible with such time constraints. A solution might be the use of the co-rotational FE

method, where geometrical non-linearities can be handled in real-time (26; 62). Nevertheless, when more complex biomechanical models need to be used these optimizations no longer hold. Alternative solutions have been proposed with different trade-offs regarding the ratio between computation time and model accuracy (58; 75; 47; 52; 28; 2). Marchesseau *et al.* (38) proposed the Multiplicative Jacobian Energy Decomposition (MJED) that allows for fast and realistic liver deformations including hyperelasticity, porosity and viscosity. Also, Miller *et al.* (45) introduced Total Lagrangian explicit dynamics (TLED) which can achieve real-time performances when coupled with explicit time integration and GPU-based solvers (29).

More recently, another class of methods made use of machine learning (ML) algorithms to solve the deformed state of a model ((35; 66; 76; 57)). Such ML models are often trained with synthetic data generated by the FE method. While the offline training phase can be computationally expensive, the online predictions satisfy real-time compliance and can provide very accurate estimations of the displacement.

Furthermore, to guarantee the high level of precision required, accurate modeling adapted to the patient anatomy needs to be pursued. The first step towards patient-specific modeling is the patient-specific geometry of the organ. Generally, the 3D anatomical model of the organ is constructed from preoperative volumetric medical images such as CT scan or MRI without too much difficulty. Moreover, boundary conditions (BCs) are essential for the FE method to produce accurate results (9). The location and the elastic properties of the BCs are also patient-specific but are not visible on preoperative images. The partial intraoperative images give inaccurate information about the BCs that are often out of the field of view of the laparoscopic camera. In addition, the correct identification of the patient-specific elasticity parameters is essential for an accurate estimation of the deformation of the considered tissue. Note that the values of these parameters are intrinsic to the choice of the constitutive law.

To ensure the aforementioned requirements in terms of model parameterization and computational efficiency, we combine FE approaches with either Bayesian methods or deep learning techniques, in order to keep control over the underlying computational model while allowing for intraoperative inputs. We first propose an image-driven stochastic assimilation method to identify the BCs on the one hand, and the elasticity parameters on the other hand. Second, we present a data-driven deep neural network that learns the desired biomechanical model including its BCs and material parameters, to predict complex non-linear deformations in real-time. This chapter is divided in three main segments. First of all, the biomechanical formulation of the general problem we want to solve is presented, with an emphasis on the role of each parameter in the predictive power of such models. Next, we look at the importance of the correct estimation of patient specific boundary conditions and material parameters, which are identified based on real observations using Kalman filtering. In the third section, we go a step further by directly learning from data the mechanical behavior of a liver through deep neural networks. Lastly, the learned model is adapted to patient specific properties through transfer learning.

## 1.2 Numerical simulation of hyperelasticity problems

### 1.2.1 Hyperelasticity of soft tissues

The deformable behavior of soft tissues can be described following the laws of continuum mechanics. Hyperelastic formulations are usually exploited to characterize biological materials undergoing large deformations, which cannot be accurately handled by linear models (17). Our reference problem corresponds to the boundary value problem of computing the deformation of a hyperelastic material under both Dirichlet and Neumann boundary conditions. The solid occupies a volume  $\Omega$  whose boundary is  $\Gamma$ . We assume the Dirichlet conditions on  $\Gamma_D$ , a subset of  $\Gamma$ , known *a priori*, while Neumann boundary conditions on  $\Gamma_N$  can vary at any time step. Relying on the Lagrangian formulation, the relationship between the deformed  $\mathbf{x}$  and undeformed state  $\mathbf{X}$  of each point of the solid reads as

$$\mathbf{x} = \mathbf{X} + \mathbf{u} \quad (1.1)$$

where  $\mathbf{u}$  is the displacement field. Throughout this chapter, we describe material behavior with the Saint-Venant-Kirchhoff constitutive model, which is the simplest generalization of the linear model for large displacements. The Green-Lagrange strain tensor  $\mathbf{E} \in \mathbf{R}^{3 \times 3}$  is computed as a non-linear (quadratic) function of the deformation gradient  $\mathbf{F} = \mathbf{I} + \nabla_{\mathbf{x}}\mathbf{u}$ :

$$\mathbf{E} = \frac{1}{2}(\mathbf{F}^T\mathbf{F} - \mathbf{I}) \quad (1.2)$$

where  $\mathbf{I} \in \mathbf{R}^{3 \times 3}$  is the identity matrix. The strain-energy density function  $\mathbf{W}$  for a Saint-Venant-Kirchhoff material is obtained according to the following equation:

$$\mathbf{W}(\mathbf{E}) = \frac{\lambda}{2}[\text{tr}(\mathbf{E})]^2 + \mu \text{tr}(\mathbf{E}^2) \quad (1.3)$$

where  $\lambda$  and  $\mu$  are the material parameters called Lamé's constants, derived from the Young's modulus  $Y$  and the Poisson's ratio  $\nu$  such that:

$$\begin{aligned} \lambda &= \frac{Y\nu}{(1+\nu)(1-2\nu)}, \\ \mu &= \frac{Y}{2(1+\nu)}. \end{aligned} \quad (1.4)$$

The constitutive law is then obtained by differentiating  $\mathbf{W}$  with respect to  $\mathbf{E}$ :

$$\mathbf{S} = \frac{\partial \mathbf{W}(\mathbf{E})}{\partial \mathbf{E}} = [\lambda \text{tr}(\mathbf{E})\mathbf{I} + 2\mu\mathbf{E}] : \mathbf{E} \quad (1.5)$$

where  $\mathbf{S}$  is the second Piola-Kirchhoff stress.  $\mathbf{S}$  is related to the first Piola-Kirchhoff stress tensor  $\mathbf{P}$  by  $\mathbf{P} = \mathbf{F}\mathbf{S}$ .

Ignoring time-dependent terms, the boundary value problem formulated in material coordinates (i.e., considering kinematic quantities with respect to the undeformed geometry) is then given by:

$$\begin{cases} \nabla(\mathbf{FS}) = \mathbf{b} \text{ in } \Omega \\ \mathbf{u}(\mathbf{X}) = 0 \text{ on } \Gamma_D \\ (\mathbf{FS})\mathbf{n} = \mathbf{t} \text{ on } \Gamma_N \end{cases} \quad (1.6)$$

where  $\mathbf{b}$  gathers the external body forces,  $\mathbf{n}$  is the unit normal to  $\Gamma_N$  and  $\mathbf{t}$  is the traction applied to the boundary  $\Gamma_N$ . The weak form of (1.6), obtained from the principle of virtual work, brings forward the boundary term and reads as:

$$\int_{\Omega} \mathbf{S} : \delta \mathbf{E} \, d\Omega = \int_{\Omega} \mathbf{b} \eta \, d\Omega + \int_{\Gamma_N} \mathbf{t} \eta \, d\Gamma \quad (1.7)$$

where  $\delta \mathbf{E} = \frac{1}{2}(\mathbf{F}^T \nabla \eta + \nabla^T \eta \mathbf{F})$  is the variation of the strain, and  $\eta = \{\eta \in H^1(\Omega) \mid \eta = 0 \text{ on } \Gamma_D\}$  is any vector-valued test function ( $H^1(\Omega)$  being a Hilbert space). The left side of equation (1.7) denotes the internal virtual work, and the right side, the virtual work from the applied external load.

### 1.2.2 Finite element method

A typical approach to find a numerical solution to equation (1.7) is the finite element method (FEM). FEM relies on a discretization of the domain into a finite number of elements, usually hexahedral (H8) or tetrahedral (T4). The displacement of each point in the volume is represented as a function of the displacement values at the element nodes. The methods we propose in the following rely on either H8 or T4 elements. H8 elements are known to have better convergence and stability, but it is difficult to use them to describe irregular shapes (72). On the contrary, T4 elements can fit complex geometries, but can be highly inaccurate in the computation of stresses and strains (8). Therefore, tetrahedral meshes are the main approach used for solid organs.

Due to the non-linearity of equation (1.2), the unknown displacements are obtained as the solution of a non-linear system of equations. Using an iterative Newton-Raphson method, from an initial displacement  $\mathbf{u}^0$ , we try to find a correction  $\delta_u^n$  after  $n$  iterations that balances the linearized set of equations:

$$\dot{\mathbf{K}}^{n-1} \delta_u^n = \mathbf{r}(\mathbf{u}^0 + \delta_u^{n-1}) + \mathbf{b} \quad (1.8)$$

where  $\dot{\mathbf{K}}$  is the tangent stiffness matrix and  $\mathbf{r}$  is the internal elastic force vector. In order to solve the linearized system, both the matrix  $\dot{\mathbf{K}}$  and the vector  $\mathbf{r}$  need to be re-computed at each iteration. A more detailed description of the solution process is described in (10).

### 1.3 Stochastic identification of patient-specific properties

Our finite element simulations depend on parameters of the constitutive model such as Young's modulus, Poisson's ratio and the boundary conditions (BCs). Since the liver may be represented with an incompressible material, Poisson's ratio can be safely set to a value close to 0.5. However, the value of Young's modulus  $E$  is more difficult to estimate as it varies with the age of the patient or even with pathology. For instance, a cirrhotic liver is significantly stiffer than the average. Therefore, values from the literature do not directly match each patient. Besides that, the location and the elastic properties of the attachments of the organ play a major role in the accurate approximation of the displacement field. Such BCs are not visible in the preoperative images and it is difficult to estimate them intraoperatively as they are often out of the field of view of the surgery.

The elastic properties of materials can be identified by solving inverse problems (81; 21; 36; 74) or using elastography techniques (71; 50; 79) initially developed for diagnosis purposes. Some works have focused on the estimation of BCs intraoperatively such as (59), (64) and (28) but these methods are difficult to use in practice as either additional intraoperative scanning is required or they are sensitive to anatomical variations. Moreover, when acquiring information intraoperatively, observational errors may occur, thus introducing uncertainty to the system. Alternative solutions accounting for such uncertainty rise from the use of Bayesian methods. For instance, authors in (44), employed the reduced-order unscented Kalman filter (ROUKF) to estimate Young's modulus of a porcine sclera based on observations extracted from optical coherence tomography images. Also works in (51; 60), employed the ROUKF to model the BCs of a liver as stochastic parameters, leading to more accurate simulations of the deformations of the organ.

Similarly to works in (60; 44), we propose to use the ROUKF to estimate the value of Young's modulus and the BCs of a liver using observations of the target model. To this end, each sought parameter  $p$  is described as a stochastic parameter associated to a Gaussian probability density function (PDF). Initially  $p \sim \mathcal{N}(\mu_0, \sigma_0)$  with  $\mu_0$  the mean value of  $p$  reported in the literature and  $\sigma_0$  its standard deviation. The aim of the assimilation process is to iteratively reduce the standard deviation  $\sigma$  of  $p$  in order to find the most likely value for  $\mu$ . To this end, the PDF of the parameter  $p$  is transformed based on observations. The transformation of the PDF is modelled using a ROUKF which can handle non-linear processes, and is computationally efficient (48).

In this section we provide a brief description of the ROUKF algorithm, that is first used to estimate Young's modulus of a synthetic liver, and in a second time used to estimate the boundary conditions of an *in vivo* liver.

### 1.3.1 ROUKF: Overview of the algorithm

Once the FE model of the organ is built (from a preoperative CT scan for instance), the constraints imposed on the surface  $\Gamma_D$  need to be identified in order to generate a deformation. In other words, the organ attachments representing Dirichlet boundary conditions and the traction or displacement imposed on the free part of the boundary need to be identified. In this work, the former can be either fixed (Sec. 1.3.2) or set as a set stochastic of parameters (Sec. 1.3.3) and the latter can be determined intraoperatively. During the intervention, points in the surface of the organ can be tracked in each video frame. Such points are called *features* and are separated into *control features* and *observation features*. The control features govern the deformation of the liver model (imposed displacement on  $\Gamma_D$  or traction if a force sensor is available) and the observation features correspond to ground truth data (used in the filter *correction* phase to compute the Kalman gain). The control features can be selected close to the surgical tool and be used to prescribe displacements in the mechanical model.

An efficient implementation of a Bayesian inference method able to process nonlinear systems like our models is the Unscented Kalman Filter (UKF) (30). Compared to an Extended Kalman Filter, it does not require to compute the Jacobian of the system, which would be prohibitive given the size of our problem. The unknown data to be estimated (the *stochastic state* of the system) is described as a Gaussian distribution, which transformation through the nonlinear system is performed using an *unscented transformation* (see (30) for details). The main idea is to parameterize the Gaussian distribution using a set of *sigma points*, which hold the mean and covariance information, but are easier to transfer through a nonlinear function. The general algorithm is described in Alg. 1. It consists of a loop that contains two main steps. During the *prediction* step we form the new hypothesis about the estimated state, while during the *correction* step we correct it by comparing the predicted measurements with (noisy and partial) observations.



**Algorithm 1:** Main steps of unscented Kalman filter

---

```

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

```

---

The prediction step can be very costly when using a model with many degrees of freedom, as it is the case when using a FEM method. Using the simplex method to generate the sigma points would require  $m + 1$  simulations if  $m$  is the number of elements in the stochastic state vector (line 9 of the algorithm). With a mesh of  $n$  nodes and  $k$  stiffness parameters, this would mean  $3n + k + 1$  simulations. A simple FEM mesh of only a few hundred nodes would be too time-consuming for a clinical application, as it would require more than 300 simulations for each step of the assimilation process. To solve this issue, we use a Reduced Order Kalman Filter (ROUKF) instead of the UKF. This method significantly reduces the computation cost since only  $k + 1$  simulations (in the best case) are required. This approach was proposed in (60).

Let us assume there are  $k$  unknown parameters in our model, so  $k$  different parameters to estimate that can be either the elasticity of the material or the elasticity of the organ attachments. Since we are using the simplex version of the ROUKF, there are  $k + 1$  sigma points meaning that  $k + 1$  evaluations of the model are performed in each prediction step of the assimilation process. At each step of the assimilation, the control features are extracted from the actual video frame and mapped onto the FE model through barycentric coordinates, in order to prescribe displacements. At the first step,  $\mu$  and  $\sigma$  are initialized to  $\mu_0$  and  $\sigma_0$  for each parameter. Then,  $k + 1$  vectors of parameters are sampled and  $k + 1$  simulations are performed. Each simulation corresponds to one of the sampled  $k + 1$  values of the parameter and they can be

done in parallel as they are independent. After the simulations for all sigma points are performed, the *a priori* expected value and covariance matrix are updated. This is called the prediction phase. Later, in a correction phase, the extracted observation features are compared to the model predicted positions to compute the innovation that is used to compute the Kalman gain. The *a posteriori* expected values and the covariance matrix are computed based on the Kalman gain.

### 1.3.2 Estimation of the Young's modulus using Kalman filters

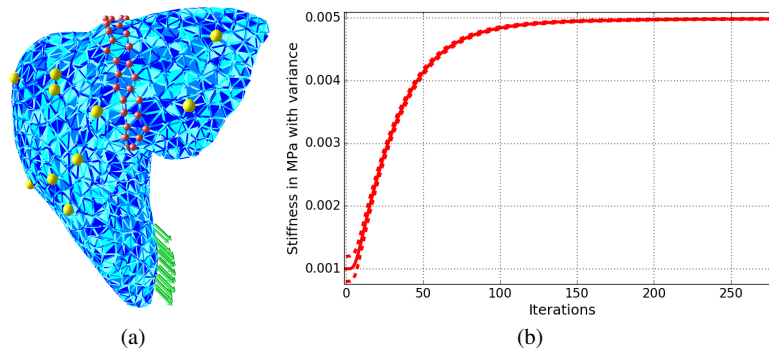


Fig. 1.1: (a) Liver simulation mesh made of 11,000 tetrahedral elements. The red points highlight the fixed points (Dirichlet boundary conditions), the yellow points correspond to the observed features, and the green arrows illustrate the direction of the applied forces. (b) Variation of  $\mu$  and  $\sigma$  for Young's modulus estimation using the ROUKF. The value of the parameter converges to  $4992 \pm 15 Pa$  in 500 seconds.

In this paragraph, we aim at estimating the value of the Young's modulus of the liver using the ROUKF. This estimation is done using synthetic data, but a similar process can be followed for real data. We build a biomechanical model of the liver, with fixed boundary conditions (red points in Figure 1(a)) to mimic the effect of the falciform ligament and of the vascular tree. A force of fixed magnitude and varying amplitude is continuously applied to one of the liver lobes to generate observations (yellow points in Figure 1(a)). The amplitude of such force follows the sinusoidal function  $\frac{1}{2} \times (1 - \cos(2 \times \pi \times \tau))$  where  $\tau$  is a period. In this case, the control features defined in section 1.3.1 correspond to the force applied (that is known). The Young's modulus is set to  $5,000 Pa$  in the reference simulation.

For the initialization of the ROUKF, we set  $\mu_0$  to  $1,000 Pa$  and  $\sigma_0$  to  $200 Pa$ . The state vector contains all the degrees of freedom of the mesh and the parameters to estimate (one parameter in our case). Hence, there are only 2 sigma points which allows a very fast assimilation process to take place as only two evaluations of the

model need to be performed at each prediction phase. As depicted in Figure 1(b), the value of the Young's modulus reaches rapidly a value close to the ground truth (at iteration 150,  $\mu = 4948$  and  $\sigma = 97$ ). The value of the parameter converges to  $4992 \pm 15 Pa$  in 631 iterations (that is 500 seconds). Such assimilation process could take place before the surgery starts. Note that if the assimilation needs to be done in real-time, the simulations could be parallelized and simplified (we chose here a relatively high mesh resolution).

### 1.3.3 Estimation of boundary conditions using Kalman filters

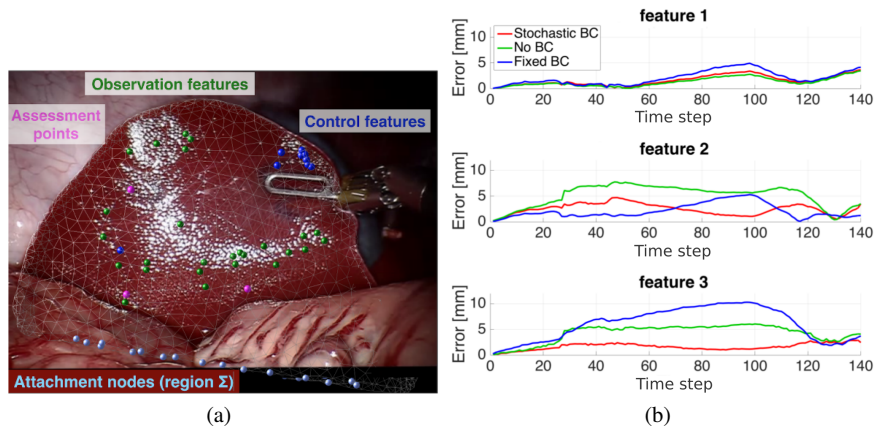


Fig. 1.2: (a) The first frame of the video sequence with features. (b) Temporal evolution of the assessment error computed for each assessment point.

Apart from material properties, the same approach can be employed to estimate the unknown attachments of the liver. We consider a scenario where an *in vivo* porcine liver is deformed with laparoscopic pincers. A video sequence of 7 s was acquired with a laparoscopic monocular camera inserted in the porcine abdomen inflated with gas. We assume there is a region  $\Sigma$  on which hidden boundary conditions are applied.

Before the intervention, a CT scan was collected from the organ's geometry and a FE model was built following the pipeline described in section 1.2.2. A tetrahedral mesh having 315 nodes was generated. The obtained model is fixed with elastic springs in a region  $\Sigma$  that is hidden to the laparoscopic camera view. There are 35 nodes in  $\Sigma$  meaning that 35 nodes were attached with springs to mimic the boundary conditions at these specific locations. The elasticity parameters of such springs are modelled as stochastic parameters (see section 1.3.1). The elasticity values can

range from 0 (no attachment) to high values (stiff attachment). A different elasticity is associated with each spring.

Known surface displacements are prescribed on a small area of the visible surface to mimic the effect of the surgical tool based on the control features. The considered scenario falls within the category of displacement-zero traction problems, where the relation between surface and volume displacements is independent of the Young's modulus, for homogeneous materials (46). As a consequence, without lack of generality we set  $Y$  to the fixed value of  $5,000 Pa$ , which is the average stiffness value for a liver.

For the initialization step of the ROUKF,  $\mu = 0$  and  $\sigma = 0.01$  for each parameter. Three assessment points are placed on the surface of the liver mesh in order to compute the prediction error between the observed data and the model. We ensure that the assessment points are different from the control and the observation features (see Fig. 2(a)). In Fig. 2(b) is shown the temporal evolution of the prediction error computed over the three assessment points. The error achieved with the stochastic simulation is compared to the error obtained with either fixed BCs (e.g. stiff attachment) or without BCs (e.g. spring stiffness set to zero). Results show that the stochastic simulation leads to smaller errors than the deterministic simulation.

## 1.4 Deep neural networks for data-driven simulations of a liver

Methods described in Section 1.3 allow to estimate the boundary conditions and the material parameters of the anatomy, assuming that the constitutive model is known. In practice, the choice of the constitutive relation is usually based on previous works and/or phantom tests, and it is not possible to guarantee that it is the most representative of the real clinical scenario. Within this context, it seems natural to try to learn the complete biomechanical behavior of the organ directly from real data acquired intraoperatively (including its constitutive equation).

In recent years, machine learning (ML) started to revolutionize several fields (vision, language processing, image recognition, genomics). With sufficient ground truth data, machine learning algorithms can map the input of a function to its output without any mathematical formulation of the problem, thus acting like a black box. The high inferring speed of these methods makes it useful for many applications where the prediction speed is of critical importance. Due to this characteristic and the fact that they are driven directly by data, these methods seem promising for the learning of the entire mechanical behavior of the anatomy without relying on prior models.

Some first attempts that exploit learning methods to estimate the deformation of biological tissues have recently been made. By implicitly encoding soft tissue mechanical behavior in the trained ML models, they proved successful to predict the entire 3D organ deformation starting either by applied surface forces (49; 76; 65; 42) or by surface displacements (63; 11; 35; 43). However, the accuracy of a ML model highly depends on the quality and on the amount of data used to train it. In an ideal

scenario, such a model would be trained with an infinite amount of real patient-specific noise-free data, which is in practice not possible. As a matter of fact, acquiring large number of volumetric deformations of an organ is a challenging problem. Moreover, the constraints applied should be precisely controlled which is in practice very complicated to guarantee. Within this framework, FE simulations can be exploited to generate synthetic data that is highly representative of the reality, to be used as training samples.

Among the various ML techniques, the use of neural networks (NN) has considerably increased. This is due to the fact that they are the building blocks of deep learning, a class of methods which is able to learn data representations and has demonstrated strong abilities at extracting high-level representations of complex processes. For example, neural networks are used by Tonutti and Rechowicz to predict the displacement of brain tumors and of the rib cage surface respectively, starting from the acting forces (76; 65). However, both these works do not predict whole volume deformation but only surface displacements. Neural network based methods have been also used to predict liver deformation in augmented surgery. Moorooka et al. trained a NN to predict liver deformations for a given input force. They use their model together with PCA to compress the size of the output deformation modes, and thus reduce the training time. Although the model proved able to learn the deformation modes, it was only applied to simulated data and not to real cases (49). An additional example is proposed by Pfeiffer et al., who used a deep NN to estimate liver deformations from the known displacement of partial liver surface. The innovative aspect of this work is that the proposed network is able to provide an accurate prediction on a liver mesh even though the synthetic data used for the training were generated from a set of random meshes. Similarly, Pellicer-Valero et al. trained a NN on various liver geometries by registering them to an average liver geometry. However, the authors rely on the assumption that both the boundary conditions and the elastic properties of the object are known, and they did not test their network performance on real data acquired during surgery (63; 57).

From all these works it emerges that the main advantage of using neural networks to predict anatomical deformations is that the prediction speed is in the order of few milliseconds and is not affected by the complexity of the model used to generate the training dataset. In this section, we propose a method that, similarly to the approaches described above, allows for extremely fast and accurate simulations by using an artificial neural network that learns the stress-strain relationship directly from data, without any *a priori* mathematical formulation of the problem. Such a network can not only learn the desired biomechanical model, but also the desired boundary conditions and material properties; and predict deformations at haptic feedback rates with very good accuracy. This section is divided in three main segments. First of all, we present the selected network based on a U-Net architecture and the strategy adopted for the generation of training data. Afterwards, we report some representative results, obtained when using the proposed U-Net both in simulated and real scenarios. In a further section, we explain how we employed transfer learning methods to make the U-Net able to generalize to new unseen boundary conditions and elastic parameters.

### 1.4.1 Method

In this section we propose to use a deep neural network that learns the relation function between surface constraints and volumetric deformation accounting for the specificity of each patient. The general procedure, named the *U-Mesh framework*, consists in training a 3D U-Net architecture (67) with synthetic deformed meshes generated with the FEM described in section 1.2.2.

#### 1.4.1.1 The U-Mesh framework

Formally, our network  $h$  is a parameterized function that accepts a  $3 \times n_x \times n_y \times n_z$  tensor of input constraints  $\mathbf{C}$  and produces a tensor of volume displacements  $\mathbf{U}_v$  of the same size as output. The computational domain  $\Omega$  is sampled by a 3-dimensional grid of resolution  $n_x \times n_y \times n_z$  (see Fig.1.5(a)). The tensor  $\mathbf{C}$  represents the constraints applied over the surface boundary  $\Gamma$  of the domain.

Our problem consists of finding the function  $h$  that produces the best estimations of the displacement field given prescribed constraints  $\mathbf{C}$ . This is performed by minimizing the expected error over the training set  $\{(\mathbf{C}_n, \mathbf{U}_{v_n})\}_{n=1}^N$  of  $N$  samples:

$$\min_{\theta} \frac{1}{N} \sum_{n=1}^N \|h(\mathbf{C}_n) - \mathbf{U}_{v_n}\|_2^2. \quad (1.9)$$

To characterize our network  $h$ , we choose a 3D U-Net (67) architecture for its similarities with model order reduction techniques from the mechanics community. It is a modified fully convolutional network with an encoding path that transforms the input into a low-dimensional space and a decoding path that expands it back to the original size (see Fig. 1.3). Additional skip connections transfer detailed information along matching levels from the encoding path to the decoding path. For a more detailed explanation of the U-Net, the reader may refer to our previous publication on the subject (42).

#### 1.4.1.2 Synthetic data generation for U-Net training

Training data for our network are made of pairs of  $(\mathbf{C}, \mathbf{U}_v)$  which are obtained from the previously explained FEM. We perform multiple simulations by imposing random constraints on the boundary  $\Gamma$ . At the end of each simulation, the pair of imposed constraints and obtained volumetric deformation is stored as an element of the data set.

For a correct spatial understanding, the U-Net requires regular grids as input, meaning that the displacements of the considered object need to be encoded in a regular grid. To this end, we propose to mesh the domain  $\Omega$  using tetrahedral elements to compute the FE deformations and then, map a 3-dimensional regular grid onto the tetrahedral mesh to follow its deformation (see Fig.1.5(a)). This mapping introduces

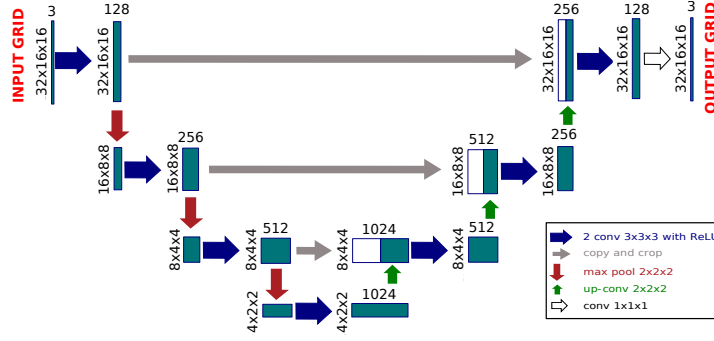


Fig. 1.3: Network architecture for an initial grid resolution of size  $20 \times 16 \times 15$ , padded to  $32 \times 16 \times 16$ , 128 channels in the first layer and 3 steps

an approximation error that is reduced when the grid resolution is increased. Random forces are directly applied to the surface of the object and then mapped to the nodes of the grid and to the T4 elements. At this stage, we can store the imposed constraints  $\mathbf{C}$  (applied traction on grid nodes or grid surface displacement) and the resulting volumetric deformation  $\mathbf{U}_v$  as a point of the data set. In total  $N$  training samples and  $M$  testing samples are generated.

The generated  $N$  training samples are used to train the network by minimizing Eq. 1.9 with the Adam optimizer (31). All our experiments are performed in a GeForce 1080 Ti using a batch size of 4 and 100,000 iterations for training. We use a PyTorch implementation of the U-Net. We recall that the batch size is the number of samples that are given to the network at each iteration of the minimization process.

### 1.4.1.3 Validation metrics

To assess the efficiency of our method, we perform a statistical analysis of the error over the testing data set  $\{(\mathbf{C}_m, \mathbf{U}_{v_m})\}_{m=1}^M$ . Let  $\mathbf{U}_{v_m}$  be the ground truth displacement tensor for sample  $m$  generated using the FEM described in section 1.2.2 and  $h(\mathbf{C}_m)$  the U-Mesh prediction. We define the *mean Euclidean error*  $e$  between  $\mathbf{U}_{v_m}$  and  $h(\mathbf{C}_m)$  for sample  $m$  as:

$$e(\mathbf{U}_{v_m}, h(\mathbf{C}_m)) = \frac{1}{n} \sum_{i=1}^n \|\mathbf{U}_{v_m}^i - h(\mathbf{C}_m)^i\|_2 \quad (1.10)$$

where  $n$  is the number of nodes of the mesh. We compute the average  $\bar{e}$  and standard deviation  $\sigma(e)$  of such norm over the testing data set. The *mean Euclidean error* represents the intuitive nodal distance, averaged over all the nodes of the mesh.

### 1.4.2 Predicting the deformation of the liver

In this section we will show the performance of the U-Mesh in predicting the deformations of a liver. To start with, the U-Net learns to predict the displacement field of a virtual liver given an input contact force. In a follow-up, the U-Net is used in an augmented reality context, where the full volumetric displacement field needs to be estimated from a partial surface deformation. In both cases the network is trained with FEM-generated data since for the moment, we do not know how to collect a sufficient amount of real volumetric information of a liver using current imaging techniques.

#### 1.4.2.1 U-Mesh on a synthetic liver

A surface mesh is obtained from a pre-operative CT scan of a human liver. The liver volume is meshed with 4859 tetrahedral elements and Dirichlet boundary conditions are used (67 nodes between the two lobes were fixed) to mimic the effect of the vascular tree and of the falciform ligament (1). The length of the liver is  $0.2\text{ m}$ . The Young's modulus  $Y$  is set to  $5,000\text{ Pa}$  and the Poisson's ratio to  $0.48$ . A regular grid of resolution  $21 \times 23 \times 21$  is mapped onto the tetrahedral mesh to encode the displacement fields and the forces in a "U-Net interpretative manner".

Normal forces of random magnitudes are computed on the liver surface. Only one force is applied at each time step on a small surface area. To fit the time requirements of a clinical routine (e.g. a few hours between the pre-operative CT scans and the surgery), we decided to limit the size of the data set to  $2,000$  samples (generated in  $180\text{ min}$ ). The data set consists of  $2,000$  pairs of input forces and corresponding volume deformation.  $N = 1,600$  samples are used to train the network in  $327\text{ min}$  and  $M = 400$  samples are left for validation.

Metrics obtained on the validation set are reported in Table 1.1. The maximal *mean Euclidian error* over the testing data set is of only  $6.8e-04\text{ m}$  (see Fig. 1.5(b)) and the maximal deformation is  $0.08\text{ m}$ . The most impacting result is the small prediction time: outputs are predicted in only  $3.47 \pm 0.60\text{ ms}$ . In Fig. 1.4 are shown some samples of U-Mesh-deformed livers with the corresponding reference solutions.

These results show the potential of the U-Mesh in applications requiring both high accuracy and speed. In the following, we will show its performance on an augmented reality scenario where the available information can be noisy and sparse.

$\bar{e}$ in m	$\sigma(e)$ in m	prediction time in ms	training time in min
$6.94e-05$	$7.81e-05$	$3.47 \pm 0.60$	327

Table 1.1: Error measures on a liver of length  $0.2\text{ m}$  immersed in a  $21 \times 23 \times 21$  grid. The maximal  $e$  is  $6.8e-04\text{ m}$  and the maximal deformation is  $0.08\text{ m}$ .



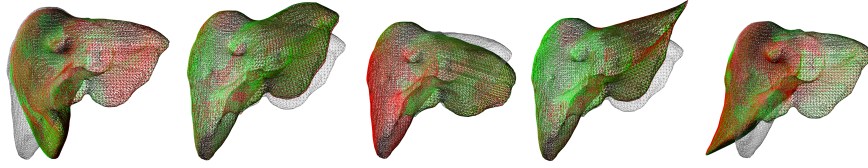


Fig. 1.4: Various liver samples from the testing data set. The reference solution appears in red and the U-Mesh prediction is in green. The rest shape of the liver is shown in gray.

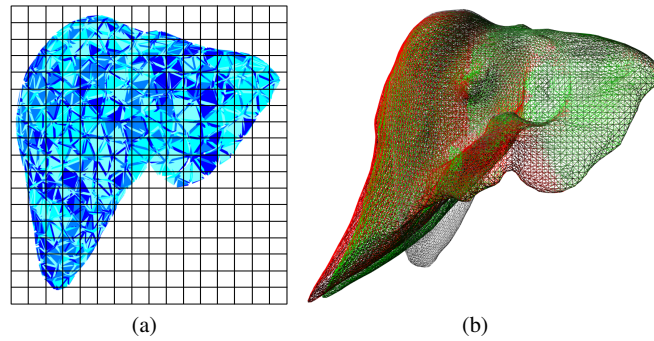


Fig. 1.5: (a) The computational domain  $\Omega$  is meshed with T4 elements and a 3-dimensional grid of resolution  $n_x \times n_y \times n_z$  is mapped onto the tetrahedral mesh. (b) Sample with maximal nodal error ( $0.00068 m$ ) The reference solution is shown in red and the U-Mesh prediction is in green. The rest shape is shown in gray.

#### 1.4.2.2 U-Mesh for augmented hepatic surgery

To build an augmented view of a liver during surgery we need to perform an elastic registration of the preoperative model to the intraoperative images acquired with a 3D imaging device (see Fig.1.6). While in minimally invasive surgeries a laparoscopic camera can be used to acquire a video of the abdominal cavity, in open surgeries and RGB-D sensor can capture the surface deformation of the tissues. From such images, a partial point cloud of the liver surface can be extracted using one of the methods listed in (62). To perform the elastic alignment between the preoperative internal structures and the surgical live images, the preferred method is the co-rotational FEM as it can provide a real-time estimation of the displacement field(62). However, the complex deformations happening during surgical manipulations may not be correctly taken into account by such a simple model. To overcome this issue we

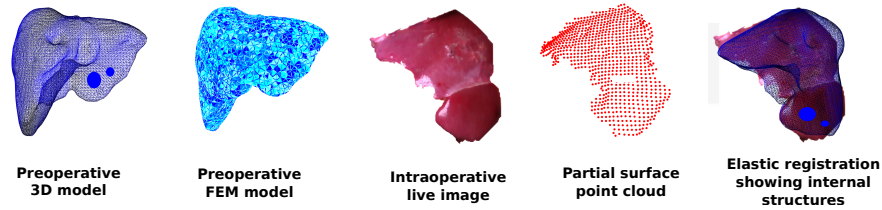


Fig. 1.6: Augmented reality pipeline: preoperative internal structures are mapped in real-time onto the live image of the organ using a FEM model.

propose to replace the FEM step with the U-Mesh trained on a more sophisticated FEM model (typically not capable of achieving real-time computations).

The U-Mesh needs to learn to predict full volumetric displacements from partial surface point clouds that give information about the position of some points of the surface of the liver. These positions can be translated as prescribed constraints. As explained in section 1.2.2, using H8 elements lead to better approximation of the stress and the strain. Therefore, to generate the data sets, we choose to use a FEM combined with an immersed boundary method (18) as it allows for the use of regular hexahedral meshes to compute accurate deformations of the liver. It is worth noting that in this scenario the FEM mesh directly matches the input to the U-Net, thus avoiding the approximation error introduced by the mapping in the previous section.

We can assume that during surgery, half of the surface of the liver is visible to the camera. As depicted in Figure 1.7, 100 points are uniformly sampled in the visible part of the surface to mimic a point cloud. Then, 100 simultaneous forces of random magnitude and direction are applied to these points in order to generate nearly random displacements. The training data set consists of pairs of  $(\mathbf{U}_s, \mathbf{U}_v)$  where the input to the network  $\mathbf{U}_s$  corresponds to the surface point cloud mapped onto the regular grid. For the same reasons stated in previous section, we limited the size of the data set to 2,000 samples ( $N = 1,600$  for training and  $M = 400$  for testing)(see Fig. 1.8 for examples of the generated deformations). It is worth mentioning that no patient-specific parameterization of the biomechanical model is required since for homogeneous materials, the relation between the surface and the volumetric displacements is independent of the stiffness of the object (46), and only depends on the Poisson's ratio (set to 0.49 as soft tissues can generally be described as incompressible).

Once the network is trained, we assess our approach on *ex vivo* human liver data, on which ten markers are embedded to compute target registration errors (TRE). During the experiments, surface data is obtained with an RGB-D sensor and ground truth data acquired at different stages of deformation using a CT scan. The RGB-D point clouds can be interpolated onto the regular grid to obtain per-node displacements on the surface and can be given as input to the network that in turn predicts the volumetric displacement fields. Each new RGB-D point cloud can be fed

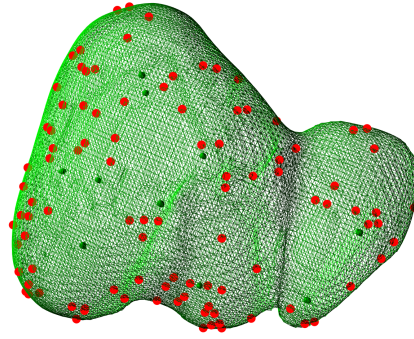


Fig. 1.7: Virtual point cloud on the visible surface of the liver to generate random displacements.

to the network, thus generating a continuous visualization of the internal structures of the organ.

The marker predicted positions are compared to our *ex vivo* ground truth by computing TREs (see Fig. 1.9). The average TRE at the 10 markers is of only  $2.92\text{ mm}$  with a maximal value of  $5.3\text{ mm}$ . The same scenario, but this time using a co-rotational FE method, leads to an average TRE of  $3.79\text{ mm}$  and is computed at about 25 ms. The solution of the Saint Venant-Kirchhoff model, used to train our model, gives nearly the same error (which was expected) but for a computation time of 1550ms.

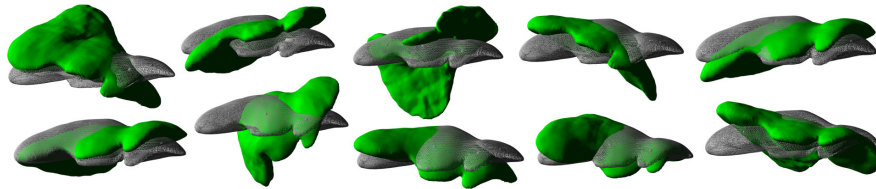


Fig. 1.8: Examples of generated deformations to train the network.

## 1.5 Updating the trained model through transfer learning

As mentioned above, there exists a correlation between our method and model reduction techniques. There is an important body of work in this area, with a well-established understanding of the process linking the fast (macro) model to the full (micro) model (82). Such theory-driven approaches define how to generate reduced

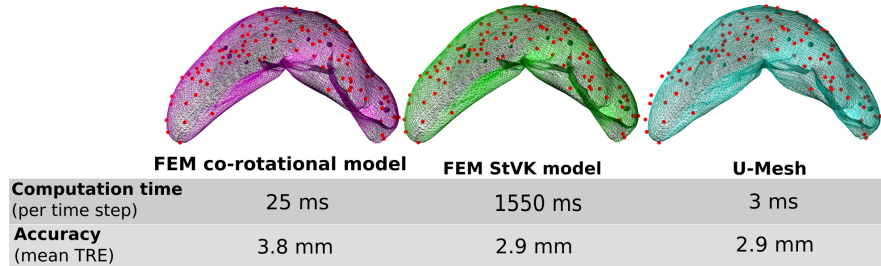


Fig. 1.9: Comparison between the reference co-rotational FEM, the Saint Venant Kirchoff model used to train the network and the U-Mesh.

models with adapted parameters that characterize the full (micro-scale) model (83). Our Deep Learning approach does something similar by learning the key characteristics (deformation and parameters) of the full-scale model, but using a data-driven approach for this.

When applied in the context of surgery, both approaches share the same limitation. The full model (micro-model) cannot always be correctly parametrized until the surgery has started, as some model parameters are not measurable in pre-operative images. In this case, the use of transfer learning methods can offer a natural, data-driven solution for adapting the neural network to a particular patient. For methods based on reduced models, Bayesian approaches are probably a good alternative, as they can estimate material properties from a probability distribution and a priori knowledge of the parameter value.

As mentioned in the section 1.3, boundary conditions have a significant impact on the accuracy of the predictions computed by biomechanical models. However, since they are hard to identify, we want to ensure the robustness of the U-Mesh to the variability of the BCs. We will show that small amount of data is required to learn patient-specific BCs, when refining a network pre-trained with variable BCs from an appropriate distribution. This could help to significantly reduce the expensive cost of the offline data generation phase. Lastly, since real data can be sparse and noisy, we explore the behaviour of the U-Net when the input tensor  $\mathbf{C}$  is highly sparse, and the effect of noise on the quality of the predictions.

### 1.5.1 Beam with hidden fixed Dirichlet BCs

In this section, we compare the accuracy of the U-Mesh either when trained from scratch with up to 16, 128 samples, or when pre-trained on various BCs and refined on the target BCs.

We consider a deformable beam (size  $4 \times 1 \times 1 \text{ m}^3$ ,  $E = 300 \text{ Pa}$ ,  $\nu = 0.4$ , 500 regular H8 elements) subject to fixed boundaries on a rectangular cuboid of its bottom part (see Figure 1.10). The beam follows the Saint-Venant-Kirchhoff behavior

described in section 1.4.1.2. We generate three different training data sets ( $N_1 = 16,128$ ;  $N_2 = 1,209$ ;  $N_3 = 100$ ) and one testing data set ( $M = 4,032$ ), all drawn from the same distribution. We performed 10 trainings to compare 7 different strategies, summarised in Table 1.2. In strategies 1, 2, and 4, the U-Net is trained from scratch whereas in strategies 3, 5, 6 and 7, the U-Net is refined starting from a network pre-trained with 16,128 data with different boundary conditions (see Table 1.2).

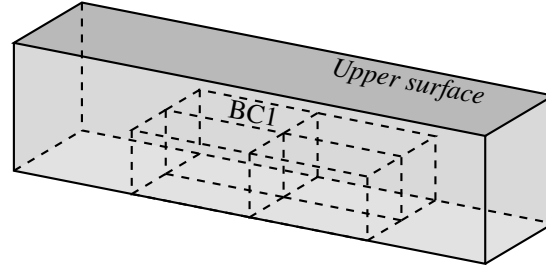


Fig. 1.10: Cuboid-like boundary conditions on which the U-Net is pre-trained in strategies 3 and 5. In strategy 6, there are four more cuboids so that the lower part of the beam is fully covered. In strategy 7, the U-Net is pre-trained on BC1.

Strategy ID	1	2	3	4	5	6	7
Training data set #	$N_1$	$N_2$	$N_3$				
Pre training data set	-	-	BC4	-	BC4	BC8	BC1

Table 1.2: Summary of the 7 strategies of interest. "BC4" stands for 4 adjacent cuboids in the middle of the hidden part of the beam. "BC8" stands for 8 adjacent cuboids fully covering the hidden part of the beam (see figure 1.10).

In Table 1.3 are reported the validation metrics computed for each strategy on the same testing dataset ( $M = 4,032$ ), as well as the index of the best iteration over 200,000 (with a saving step of 5,000). We see that strategy 3 performs better than strategy 2. More impressive yet are the strategies 5, 6 and 7 (especially 5, which, by refining, led to errors comparable to the one obtained with 12x more data without refining). Furthermore, they are substantially better than strategy 4 where no refinement was done. This is an example of a scenario where the U-Net cannot accurately learn a deformation model from scratch with very few (100) data, whereas it does learn an accurate model in a few thousands iterations using transfer learning.

The mild differences between the metrics obtained for strategies 5, 6 and 7 show that the data generated for pre-training must be reasonably distributed. Indeed, even though the network benefits from the diversity of BCs encountered in the pre-training

stage, it is more efficient when these BCs are close enough to the target boundary conditions. Hence the need for a reasonable distribution.

	$e$			# training dataset	best iteration
	in mm				
	avg	max	$\sigma(e)$		
1	<b>0.45</b>	2.48	0.29	16,128	200
2	<b>0.71</b>	2.96	0.47	1,209	180
3	<b>0.52</b>	2.66	0.32	1,209	40
4	<b>3.49</b>	32.0	3.59	100	200
5	<b>0.80</b>	4.85	0.61	100	15
6	<b>1.11</b>	5.89	0.80	100	5
7	<b>0.98</b>	8.88	0.82	100	5

Table 1.3: Error measures over all seven scenarios. Best iterations are given in thousands. Transfer learning situations are highlighted in red, first and second best results in green and blue.

So far we have seen that refining from an average model significantly reduces the quantity of data required to learn a deformation model. Results in Table 1.4 highlight the fact that it also speeds up the model convergence. Computing more metrics, we found that a good accuracy is reached approximately 20x faster when refining with 100 data than when starting from scratch with either 16,128 or 1,209 samples.

	$e$			# training dataset	iteration
	in mm				
	avg	max	$\sigma e$		
1	<b>5.74</b>	27.4	3.73	16,128	5
2	<b>7.47</b>	42.1	5.36	1,209	5
3	<b>0.55</b>	2.91	0.35	1,209	5
4	<b>5.25</b>	44.2	4.71	100	5
5	<b>0.81</b>	4.1	0.58	100	5
6	<b>1.09</b>	5.37	0.7	100	1.5
7	<b>0.95</b>	8.66	0.84	100	1.15

Table 1.4: Error measures at iteration 5000. We relaunched the training of 6 and 7 with a step of 50 iterations to ensure there was no significant overfitting before iteration 5,000, hence the values "1.5" and "1.15" (as a matter of fact, iterations 1,500 and 1,150 were actually slightly better than 5,000). Transfer learning situations are highlighted in red.

For completeness, we also investigated the case where the constitutive law changed between pre-training and refining stages. For pre-training, we modelled a beam with the linear Hooke's law, and for refining, we chose the Saint-Venant Kirchoff constitutive equation to model the deformations of the beam. In this scenario as well, we observed that transfer learning reduces the amount of data required to reach a given accuracy even when the base equations are complexified.

### 1.5.2 New boundary conditions and sparse data

As mentioned in previous paragraphs, real intra-operative data can be sparse and noisy. In this section, we show that the U-Net can still learn models when the training input tensors only contain a sparse view of the displacement  $\mathbf{U}_s$  imposed on the upper surface  $\Gamma_D$ . In return, the accuracy is reduced and we show that transfer learning is barely relevant in such an adverse scenario.

We consider the same beam as described in 1.5.1, except that the beam is supported on both ends (fixed beam). In order to train the network, we built 2 data sets of sizes  $N_1 = 10,080$  and  $N_2 = 1,008$  (see 1.5.1). Here, every tensor  $\mathbf{C}$  contains the values of an imposed surface displacement, on a randomly selected subdomain of the upper surface (in between 13 and 114 non-zero displacements in the testing data set, 67 in average) - see figure 1.11. We trained the network either directly with  $N_1$  or  $N_2$  samples (strategies 1 and 2), or with  $N_2$  data after a pre-training on a stiffer beam fixed at one end (Young's modulus of 500 Pa). The pre-training was done either with sparse data (strategy 4, same distribution as the refining data set), or dense data (strategy 3, full view of the imposed upper surface displacement  $\mathbf{U}_s$ ).



Fig. 1.11: Randomly visible sub-domains of the upper face of the beam (in yellow).

In Table 1.5 are reported the validation metrics at best iteration. The average error with  $N_1 = 10,080$  is of only  $3.03\text{ mm}$  and the maximal error is  $57.1\text{ mm}$  - meaning less than  $1.5\%$  of the length of the beam as maximal error. This shows that even though U-Net may learn much more accurate models with dense data, it still deals pretty well with sparse data when provided with a large enough training dataset. We should mention that we obtained very similar results by applying an additive white Gaussian noise of variance  $N = 10^{-3}\text{ m}$  on the testing dataset. With a variance  $N = 10^{-2}\text{ m}$ , the average mean Euclidean error  $\bar{e}$  barely exceeds one centimeter. On another note, we see that there is no meaningful difference between the validation metrics of strategy 2 (1,008 data without refining) and strategy 4 (1,008 data with refining). Eventually, except when the data set is very small, we found that refining doesn't enhance accuracy in such a scenario. What is more, these results highlight the importance of pre-training the network with sparse data whenever the refining data is sparse. We further investigated the case where only very few data ( $N_3 = 100$ ) are available, and found that it was not sufficient (with or without pre-training), although the refined model was more accurate. What remains valid is that in any scenario, the U-Mesh maintained a better accuracy with transfer learning in the first thousands of iterations. Reiterating these tests with sparse data without modifying the Young's modulus between pre-training and training stage corroborated these results.

	$e$ in mm			# training dataset	best iteration
	avg	max	$\sigma(e)$		
1	<b>3.03</b>	57.1	3.57	10,080	185
2	<b>5.69</b>	101	7.11	1,008	115
3	<b>7.73</b>	106	9.02	1,008	145
4	<b>5.79</b>	88.3	6.85	1,008	95

Table 1.5: Error measures at best iteration. Transfer learning situations are highlighted in red.

## 1.6 Conclusion

In this work we have proposed a method fulfilling the real-time and precision requirements of patient-specific augmented reality. Based on *a priori* knowledge of the biomechanics of the organ, we select a constitutive model describing the relation between stresses and strains. Such relation is heavily affected by patient-specific properties such as boundary conditions and material characteristics. While obtaining these properties preoperatively may be troublesome, having information about them intraoperatively can be straightforward. In our approach, the parameters of the preoperative finite element or deep learning models are updated based on intraoperative observations. These data-driven simulations are obtained by exploiting Bayesian filtering to update the parameters of the finite element model and by employing transfer learning to update a deep learning model. In the former, each parameter can only be modified individually. Indeed, in our pipeline using the Kalman filters, to estimate the stiffness of boundary conditions, the Young’s modulus of the material needs to be fixed (and *vice versa*). A simultaneous estimation of both sets of parameters would be more complicated (yet possible), less precise (variance of the stochastic parameters will remain high) and would require very tedious fine-tuning of the filter. In the latter, not only the elasticity parameters and the boundary conditions can be changed simultaneously, but also the constitutive model itself. An interesting point of using deep neural networks is that the parameters do not need to be explicitly identified. They are encoded in the data. Hence, the network builds its own representation and through transfer learning, the weights of the network can be modified to match the targeted function.

In the second part of this work, we have presented a method that can approximate complex elastic deformations of a real liver and generate a deformed state from an RGB-D point cloud. The obtained accuracy is comparable to the one obtained with the finite element model used to train it while being about 500 times faster. Whenever the preoperative model differs from the intraoperative one, we can use transfer learning to exploit the already learned knowledge in a fast and efficient way. We have reasons to believe that the U-Net learns local correlations in the displacement field rather than an overall model only. As a consequence, if the pre-trained model represents an average liver, transfer learning should not break the constitutive model learned previously. Note that the variability between livers can



be high but it will always vary in a bounded range. For this reason, we believe that transfer learning is the key to an accurate and fast simulation of the deformations of a liver. However, the results dedicated to transfer learning are only a proof of concept as for now, we do not know how to collect intraoperative volumetric data needed to refine our network.

## References

1. Abdel-Misih, S. R., Bloomston, M., 2010. Liver anatomy. *Surgical Clinics*, 90(4), pp. 643-653.
2. Allard, J., Cotin, S., Faure, F., Bensoussan, P. J., Poyer, F., Duriez, C., Delingette, H., Grisoni, L., 2007. SOFA - an open source framework for medical simulation. In *MMVR 15-Medicine Meets Virtual Reality* (Vol. 125, pp. 13-18).
3. Alvarez, P., Chabanas, M., Rouzaf, S., Castro, M., Payan, Y., Dillenseger, J.L., 2018. Lung deformation between preoperative CT and intraoperative CBCT for thoracoscopic surgery: a case study. *Medical Imaging*, Vol. 10576D.
4. Amundarain, A., Borro, D., García-Alonso, A., Gil, J. J., Matey, L., Savall, J., 2004. Virtual reality for aircraft engines maintainability. *Mechanics and Industry*, 5(2), pp. 121-127.
5. Cotin, S., Delingette, H., Ayache, N., 1999. Real-time elastic deformations of soft tissues for surgery simulation. *IEEE transactions on Visualization and Computer Graphics*, 5(1), pp. 62-73.
6. Barbic, J., James, D. L., 2008. Six-dof haptic rendering of contact between geometrically complex reduced deformable models. *IEEE Transactions on Haptics*, 1(1), pp. 39-52.
7. Bhattacharjee, S., Matous, K., 2016. A nonlinear manifold-based reduced order model for multiscale analysis of heterogeneous hyperelastic materials. *Journal of Computational Physics*, 313, pp. 635-653.
8. Benzley, S.E., Perry, E., Merkley, K., Clark, B., Sjaardema, G., 1995. A comparison of all hexagonal and all tetrahedral finite element meshes for elastic and elasto-plastic analysis. *4th IMR*, (17) pp. 179-191
9. Bosman, J., et al.: The role of ligaments: Patient-specific or scenario-specific ? In: *ISBMS 2014*. (2014)
10. BroNielsen, M., and Cotin, S. 1996. Real-time volumetric deformable models for surgery simulation using finite elements and condensation. In *Computer graphics forum* (Vol. 15, No. 3, pp. 57-66). Edinburgh, UK: Blackwell Science Ltd.
11. Brunet, J.N., Mendizabal, A., Petit, A., Golse, N., Vibert, E., Cotin, S., 2019. Physics-based Deep Neural Network for Augmented Reality during Liver Surgery. In *International Conference on Medical image computing and computer-assisted intervention*.
12. Cignoni, P., Callieri, M., Corsini, M., Dellepiane, M., Ganovelli, F., Ranzuglia, G., 2008. MeshLab: an Open-Source Mesh Processing Tool. *Eurographics Italian Chapter Conference*
13. Cifuentes, A., Kalbag, A., 1992. A performance study of tetrahedral and hexahedral elements in 3-D finite element structural analysis. *Finite Elements in Analysis and Design*, 12, pp. 313-318
14. Clements, L. W., Chapman, W. C., Dawant, B. M., Galloway, R. L., and Miga, M. I., 2008. Robust surface registration using salient anatomical features for image-guided liver surgery: Algorithm and validation. *Medical physics*, 35(6Part1), pp. 2528-2540.
15. Collins, T., Pizarro, D., Bartoli, A., Canis, M. and Bourdel, N., 2013. Real-time wide-baseline registration of the uterus in monocular laparoscopic videos. *MICCAI*
16. Comas, O., Taylor, Z. A., Allard, J., Ourselin, S., Cotin, S., Passenger, J., 2008. Efficient nonlinear FEM for soft tissue modelling and its GPU implementation within the open source framework SOFA. In *International Symposium on Biomedical Simulation*, pp. 28-39. Springer, Berlin, Heidelberg.

17. Delingette, H., and Ayache, N., 2004. Soft tissue modeling for surgery simulation. *Handbook of Numerical Analysis*, 12, 453-550.
18. Düster, A., Parvizianb, J., Yanga, Z., Ranka, E., 2008. The finite cell method for three-dimensional problems of solid mechanics. *Comput Methods Appl Mech Eng.* 197(45-48), pp. 3768-3782
19. Fedorov, A., Beichel, R., Kalpathy-Cramer, J., Finet, J., Fillion-Robin, J.C., Pujol, S., Bauer, C., Jennings, D., Fennessy, F., Sonka, M., Buatti, J., Aylward, S., Miller, J.V., Pieper, S., Kikinis, R., 2012. 3D Slicer as an image computing platform for the Quantitative Imaging Network. *Magnetic resonance imaging*, 30(9), pp. 1323-1341
20. Fetene, B., N., Rajkumar S., and Uday S., D., 2018. FEM-based neural network modeling of laser-assisted bending. *Neural Computing and Applications* 29.6, pp. 69-82.
21. Gee M.W., Forster C., Wall W.A., 2010. A computational strategy for prestressing patient-specific biomechanical problems under finite deformation. *International Journal for Numerical Methods in Biomedical Engineering.* (26), pp. 52-72.
22. Goury, O., Duriez, C., 2018. Fast, Generic, and Reliable Control and Simulation of Soft Robots Using Model Order Reduction. *IEEE Transactions on Robotics*, (99), pp. 1-12.
23. Guo, R., Lu, G., Qin, B., Fei, B., 2017. Ultrasound imaging technologies for breast cancer detection and management: A review. *Ultrasound in medicine and biology*
24. Haferssas, R., Jolivet, P., Nataf, F., 2017. An Additive Schwarz Method Type Theory for Lions's Algorithm and a Symmetrized Optimized Restricted Additive Schwarz Method. *SIAM Journal on Scientific Computing*, 39(4), pp. A1345-A1365.
26. Haouchine, N., Dequidt, J., Berger, M. O., Cotin, S., 2013. Deformation-based augmented reality for hepatic surgery. *Studies in health technology and informatics*, 184.
26. Haouchine, N., Dequidt, J. et al. 2013. Image-guided simulation of heterogeneous tissue deformation for augmented reality during hepatic surgery. *ISMAR*, pp. 199-208
27. Heiselman, J.S., Clements, L.W., Collins, J.A., Weis, J.A., Simpson, A.L., Geevarghese, S.K., Kingham, T.P., Jarnagin, W.R., Miga, M.L., 2017. Characterization and correction of intraoperative soft tissue deformation in image-guided laparoscopic liver surgery. *J Med Imag*, 5(2)
28. Johnsen, S. F., Taylor, Z. A., Clarkson, M. J., Hipwell, J., Modat, M., Eiben, B., Han, L. , Hu, Y. , Mertzaniidou, T. , Hawkes, D. J. Ourselin, S., 2015. NiftySim: A GPU-based nonlinear finite element package for simulation of soft tissue biomechanics. *International journal of computer assisted radiology and surgery*, 10(7), pp. 1077-1095.
29. Joldes, G.R., Wittek, A. and Miller, K., 2010. Real-time nonlinear finite element computations on GPU—Application to neurosurgical simulation. *Computer methods in applied mechanics and engineering*, 199(49-52), p. 3305-3314.
30. Julier, S. J., Uhlmann, J. K. and Durrant-Whyte, H. F., 1995. A new approach for filtering nonlinear systems, *Proceedings of ACC'95* (3), pp. 1628-1632.
31. Kingma, D. P., Ba, J., 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
32. Kourounis, D., Fuchs, A., Schenk, O., 2018. Toward the next generation of multiperiod optimal power flow solvers. *IEEE Transactions on Power Systems*, 33(4), pp. 4005-4014.
33. Krücker, J., Xu, S., Venkatesan, A., Locklin, J.K., Amalou, H., Glossop, N., Wood, B., 2011. Clinical utility of real-time fusion guidance for biopsy and ablation. *J Vasc Interv Radiol* 22(4), pp. 515-524
34. Lasso, A., Heffter, T., Rankin, A., Pinter, C., Ungi, T., Fichtinger, G., 2014. PLUS: open-source toolkit for ultrasound-guided intervention systems. *IEEE Transactions on Biomedical Engineering*, 61(10), pp. 2527-2537
35. Lorente, D., Martínez-Martínez, F., Rupérez, M. J., Lago, M. A., Martínez-Sober, M., Escandell-Montero, P., Martínez-Martínez, J.M. , Martínez Sanchis, S., Serrano-López, A. , Monserrat, C, Martín-Guerrero, J. D., 2017. A framework for modelling the biomechanical behaviour of the human liver during breathing in real time using machine learning. *Expert Systems with Applications*, 71, pp. 342-357.

36. Lu J., Zhao X.F., 2009. Pointwise Identification of Elastic Properties in Nonlinear Hyperelastic Membranes-Part I: Theoretical and Computational Developments. *J Appl Mech-T Asme*. 76:061013/061011-061013/061010
37. Luo, R., Shao, T., Wang, H., Xu, W., Zhou, K., Yang, Y., 2018. DeepWarp: DNN-based Nonlinear Deformation. arXiv preprint arXiv:1803.09109.
38. Marchesseau, S., Heimann, T., Chatelin, S., Willinger, R., Delingette, H., 2010. Multiplicative jacobian energy decomposition method for fast porous visco-hyperelastic soft tissue model. *MICCAI*, pp. 235-242
39. Martínez-Martínez, F., Rupérez-Moreno, M.J., Martínez-Sober, M., Solves-Llorens, J.A., Lorente, D., Serrano-López, A.J., Martínez-Sanchis, S., Monserrat, C., Martin-Guerrero, J.D., 2017. A finite element-based machine learning approach for modeling the mechanical behavior of the breast tissues under compression in real-time. *Computers in biology and medicine*, 90, pp. 116-124
40. Meier, U., López, O., Monserrat, C. U., Alcañiz J.M., 2005. Real-time deformable models for surgery simulation: a survey. *Comput Methods Programs Biomed.*, 77(3), 183-197
41. Meister, F., Passerini, T., Mihalef, V., Tuysuzoglu, A., Maier, A., Mansi, T., 2018. Towards Fast Biomechanical Modeling of Soft Tissue Using Neural Networks. arXiv preprint arXiv:1812.06186 .
42. Mendizabal, A., Márquez-Neila, P., and Cotin, S. 2019. Simulation of hyperelastic materials in real-time using deep learning. *Medical Image Analyses*. Volume 59, January 2020, 101569 10.1016/j.media.2019.101569
43. Mendizabal, A., Tagliabue, E., Brunet, J-N., Dall'Alba, D., Fiorini, P., Cotin, S., 2019. Physics-based Deep Neural Network for Real-Time Lesion Tracking in Ultrasound-guided Breast Biopsy. *Computational Biomechanics for Medicine Workshop*.
44. Mendizabal, A., Sznitman, R., and Cotin, S., 2019. Force classification during robotic interventions through simulation-trained neural networks. *International journal of computer assisted radiology and surgery*, 14(9), 1601-1610.
45. Miller, K., Joldes, G., Lance, D., Wittek, A., 2007. Total Lagrangian explicit dynamics finite element algorithm for computing soft tissue deformation. *Communications in numerical methods in engineering*, 23(2), pp. 121-134.
46. Miller, K., Lu, J., 2013. On the prospect of patient-specific biomechanics without patient-specific properties of tissues. *Journal of the mechanical behavior of biomedical materials*, 27, pp. 154-166.
47. Modrzejewski, R., Collins, T., Bartoli, A., Hostettler, A., Marescaux, J., 2018. Soft-body registration of pre-operative 3d models to intra-operative RGBD partial body scans. *MICCAI*, pp. 39-46
48. Moireau, P., Chapelle, D., 2011. Reduced-order Unscented Kalman Filtering with application to parameter identification in large-dimensional systems. *ESAIM: Control, Optimisation and Calculus of Variations*, 17(2), pp. 380-405.
49. Morooka, K. I., Chen, X., Kurazume, R., Uchida, S., Hara, K., Iwashita, Y., and Hashizume, M., 2008. Real-time nonlinear FEM with neural network for simulating soft organ model deformation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention* (pp. 742-749). Springer, Berlin, Heidelberg.
50. Muthupillai R., Lomas D.J., Rossman P.J., Greenleaf J.F., Manduca A., Ehman R.L., 1995. Magnetic-resonance elastography by direct visualization of propagating acoustic strain waves. *Science*. 1995; 269:1854-1857. [PubMed: 7569924]
51. Nikolaev, S., Peterlík, I., and Cotin, S., 2018. Stochastic Correction of Boundary Conditions during Liver Surgery. In *2018 Colour and Visual Computing Symposium (CVCS)* (pp. 1-4). IEEE.
52. Niroomandi, S., Alfaro, I., Cueto, E., Chinesta, F., 2008. Real-time deformable models of nonlinear tissues by model reduction techniques. *Computer methods and programs in biomedicine*, 91(3), pp. 223-231.

53. Niroomandi, S., Gonzalez, D., Alfaro, I., Bordeu, F., Leygue, A., Cueto, E., Chinesta, F., 2013. Real-time simulation of biological soft tissues: a PGD approach. *International journal for numerical methods in biomedical engineering*, 29(5), pp. 586-600.
54. Niroomandi, S., Alfaro, I., Cueto, E., Chinesta, F., 2010. Model order reduction for hyperelastic materials. *International Journal for Numerical Methods in Engineering*, 81(9), pp. 1180-1206.
55. O'Flynn, E.A.M., Wilson, A.R.M., Michell, M.J., 2010. Image-guided breast biopsy: state-of-the-art. *Clinical radiology*, 65(4), pp. 259-270.
56. Paulus, C. J., Maier, R., Peterseim, D., Cotin, S., 2017. An Immersed Boundary Method for Detail-Preserving Soft Tissue Simulation from Medical Images. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 55-67.
57. Pellicer-Valero, O. J., Rup rez, M. J., Mart nez-Sanchis, S., and Mart n-Guerrero, J. D., 2019. Real-time biomechanical modeling of the liver using Machine Learning models trained on Finite Element Method simulations. *Expert Systems with Applications*, 113083.
58. Peterl k, I., Duriez, C., Cotin, S., 2012. Modeling and real-time simulation of a vascularized liver tissue. *MICCAI*, pp. 50-57
59. Peterl k, I., Courtecuisse, H., Duriez, C., Cotin, S., 2014. Model-based identification of anatomical boundary conditions in living tissues. *IPCAI*. pp. 196-205
60. Peterl k, I., Haouchine, N., Ru nka, L., and Cotin, S., 2017. Image-driven stochastic identification of boundary conditions for predictive simulation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention* (pp. 548-556)
61. Petit, A., Lippiello, V., Siciliano, B., 2015. Real-time tracking of 3D elastic objects with an RGB-D sensor. *IROS*, pp. 3914-3921
62. Petit, A., Cotin, S., 2018. Environment-aware non-rigid registration in surgery using physics-based simulation. *ACCV - 14th Asian Conference on Computer Vision*
63. Pfeiffer, M., Iediger, C., Weitz, J., Speidel, S., 2019. Learning soft tissue behavior of organs for surgical navigation with convolutional neural networks. *IJCARS* pp. 1-9
64. Plantefeve, R., Peterl k, I., Haouchine, N., Cotin, S., 2016. Patient-specific biomechanical modeling for guidance during minimally-invasive hepatic surgery. *Ann Biomed Eng.* 44 (1), pp. 139-153
65. Rechowicz, K. J., and McKenzie, F. D., 2013. Development and validation methodology of the Nuss procedure surgical planner. *Simulation*, 89(12), 1474-1488.
66. Roewer-Despres, F., Khan, N., Stavness, I., 2018. Towards finite element simulation using deep learning, 15th International Symposium on Computer Methods in Biomechanics and Biomedical Engineering.
67. Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pp. 234-241.
68. Ruiter, N., Stotzka, R., Muller, T.-O., Gemmeke, H., Reichenbach, J., Kaiser, W., 2006. Model-based registration of X-ray mammograms and MR images of the female breast. *IEEE Transactions on Nuclear Science* (53), pp. 204 - 211, 10.1109/TNS.2005.862983
69. Runge, G., Wiese, M., Raatz, A., 2017. FEM-based training of artificial neural networks for modular soft robots. *IEEE International Conference on Robotics and Biomimetics (ROBIO)*.
70. Ryckelynck, D., 2005. A priori hyperreduction method: an adaptive approach. *Journal of computational physics*, 202(1), pp. 346-366.
71. Sarvazyan A.P., Rudenko O.V., Swanson S.D., Fowlkes J.B., Emelianov S.Y., 1998. Shear wave elasticity imaging: A new ultrasonic technology of medical diagnostics. *Ultrasound in Medicine and Biology*. 24:1419-1435. [PubMed: 10385964]
72. Shepherd, J. F., and Johnson, C. R., 2008. Hexahedral mesh generation constraints. *Engineering with Computers*, 24(3), 195-213.
73. Shewchuk, J. R., 1994. An introduction to the conjugate gradient method without the agonizing pain.
74. Sinkus, R., Daire, J.L., Van Beers, B.E., Vilgrain, V., 2010. Elasticity reconstruction: Beyond the assumption of focal homogeneity. *C. R. Mec.* 2010; 338:474-479

75. Suwelack, S., Röhl, S., Bodenstedt, S., Reichard, D., Dillmann, R., dos Santos, T., Maier-Hein, L., Wagner, M., Wünscher, J., Kennigott, H., Müller, B.P., Speidel, S., 2014. Physics-based shape matching for intraoperative image guidance. *Med Phys.* 41(11):111901
76. Tonutti, M., Gauthier G., and Guang-Zhong Y., 2017. A machine learning approach for real-time modelling of tissue deformation in image-guided neurosurgery. *Artificial intelligence in medicine*, 80, 39-47.
77. Visentin, F., Groenhuis, V., Maris, B., Dall'Alba, D., Siepel, F., Stramigioli, S., Fiorini, P., 2018. Iterative simulations to estimate the elastic properties from a series of mri images followed by mri-us validation. *Medical and biological engineering and computing*, 194(21-24), pp. 1-12
78. Wang, E., Nelson, T., Rauch, R., 2004. Back to elements-tetrahedra vs. hexahedra. *Proceedings of the 2004 international ANSYS conference*
79. Xu, L., Lin, Y., Han, J.C., Xi, Z.N., Shen, H., Gao, P.Y., 2007. Magnetic resonance elastography of brain tumors: Preliminary results. *Acta Radiologica.* 48:327-330. [PubMed: 17453505]
80. Yushkevich, P.A., Piven, J., Hazlett, H.C., Smith, R.G., Ho, S., Gee, J.C., Gerig, G., 2006. User-Guided 3D Active Contour Segmentation of Anatomical Structures: Significantly Improved Efficiency and Reliability. *Neuroimage*, 31(3), pp. 1116-1128
81. Zhao, X.F., Chen, X.L., Lu, J., 2009. Pointwise Identification of Elastic Properties in Nonlinear Hyperelastic Membranes-Part II: Experimental Validation. *J Appl Mech-T Asme.* 76:061014/061011-061014/061018
82. De Angelo, M., Barchiesi, E., Giorgio, I. and Abali B. E. (2019) Numerical identification of constitutive parameters in reduced order bi-dimensional models for pantographic structures: Application to out-of-plane buckling, *Archive of Applied Mechanics*, 89(7): 1333–1358.
83. Boutin, C., dell'Isola, F., Giorgio, I. and Placidi, L. (2017) Linear pantographic sheets: Asymptotic micro-macro models identification. *Mathematics and Mechanics of Complex Systems*, 5(2): 127–162.