Augmented reality biomechanical simulations for pelvic conditions diagnoses

Hadrien Courtecuisse, Jérémie Dequidt

Augmented reality biomechanical simulations for pelvic conditions diagnoses

Hadrien Courtecuisse, Jérémie Dequidt

May 2022

1 Introduction

Digital tools are increasingly used for intraoperative assistance. Initially designed for training and learning, Finite Element (FE) simulations are now used and considered essential in the operating room. FE models can be used to display through Augmented Reality (AR) the internal structures (vessels, tumors...) on top of the intraoperative images. An essential advantage of biomechanical models is to provide physics-based extrapolation, not just geometric, of organs' deformation, even in areas where few or no intraoperative data are available. Nevertheless, a significant difficulty of such simulations concerns the need for real-time computing without sacrificing accuracy. For this purpose, this chapter shows a generic constraint-based biomechanical framework allowing for both the simulations of complex interactions (such as friction, contacts, etc.) and impose displacements coming from medical image to correct modeling errors and maintain low errors with respect to the displacement of real-structures.

In the context of diagnosing women's pelvic pathologies, medical imaging is routinely used by clinicians, especially since modern MRI allows for dynamic scanning without any artifacts due to movement. For instance, the MRI MAGNETOM® Aera SIEMENS 1.5 T allows getting 2.5 acquisitions per second of a given slice. This frequency is sufficient to capture the dynamic behavior of tissues during breathing cycles, but it is for now restricted to only one plan of acquisition, and given the absence of any volumetric information, some anatomical structures might be excluded. To overcome this limitation we propose to describe in this chapter a method that will simulate a bio-mechanical model of the pelvic system which will be registered on 2D dynamic MR images allowing to extrapolate the 2D image to estimate off-plane motions and as a result to obtain a 3D motion of the pelvic system in augmented reality. This application will provide better knowledge and understanding of pathologies such as prolapsus or abnormal mobility of tissues by limiting the subjectivity induced by human perception when mentally inferring 3D motion from 2D data in the diagnoses.

To achieve this goal, the obvious requirement is to simulate the mechanical behavior of organs with high accuracy. However, due to patient inter- and intra-variability of the mechanical properties of anatomical structures, having a predictive simulation is not achievable. We propose to mechanically constrain the simulation with MR images by adding boundary conditions that will increase the accuracy of the whole method. With any medical application that involves mechanical simulation, a strong requirement exists for the computation time. Without meeting this requirement, the simulator cannot be used in an interactive context, regardless of the quality and accuracy of models. Therefore, the underlying contributions of this chapter are a set of advanced numerical techniques, including modeling, constraints definition, and high parallelization strategies to meet these requirements.

Outline: The remainder of this chapter is organized as follows. Section 2 is dedicated to the description of numerical methods for fast simulation of deformable structures with contacts. Section 3 concerns the non-rigid registration of soft tissues with intraoperative data. Finally, in section 4 we show the benefits of our approach for the registration of the pelvis organs using dynamic MRI images.

2 Simulation of deformable structures

Our methodology is based on the following general background of deformable simulations with contact using an implicit integration time.

2.1 FE models and constitutive law

Most of the contributions that will be introduced in this chapter are generic and do not depend on the constitutive model. However, we mainly rely on the linear co-rotational approach to model deformations. This model is extremely fast and not restricted to small deformations thanks to precomputations.

Real-time implementations of nonlinear models exist. However, although more accurate, hyperelastic models usually raise stability conditions which are difficult to enforce in a context where the model is subject to user interactions or driven by data extracted from medical images. Since stiffness matrices are precomputed using the rest configuration, the stability of corotational models does not depend on the current positions (at the price of the accuracy in highly deformed states). Since stability is also essential for medical simulation, the corotational model provides a balanced solution between accuracy and computation time.

2.1.1 Linear elastic model

Volume organs are meshed with linear tetrahedral elements composed of 4 nodes and 3 Degrees of Freedom (DOF) per node. The local 12×12 stiffness matrix $\mathbf{K}_e$ for a volume element e can be written with the synthetic formulation:

$$\mathbf{K}_e = \int_{V_e} (\mathbf{C}_e \mathbf{D}_e \mathbf{C}_e \, dV_e) \tag{1}$$

where $\mathbf{D}_e$ corresponds to the stress-strain matrix parametrized by the Young's modulus E and the Poisson's ratio ν:

$$\mathbf{D}_e = \frac{E}{(1+\nu)(1-2\nu)} \begin{pmatrix} (1-\nu) & \nu & \nu & 0 & 0 & 0 \\ \nu & (1-\nu) & \nu & 0 & 0 & 0 \\ \nu & \nu & (1-\nu) & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} \end{pmatrix} \tag{2}$$

$\mathbf{C}_e$ is the strain-displacement matrix which is employed to compute strains at any point inside the element, based on its nodal displacements:

$$\mathbf{C}_e = \frac{1}{6V} \begin{pmatrix} b_1 & 0 & 0 & b_2 & 0 & 0 & b_3 & 0 & 0 & b_4 & 0 & 0 \\ 0 & c_1 & 0 & 0 & c_2 & 0 & 0 & c_3 & 0 & 0 & c_4 & 0 \\ 0 & 0 & d_1 & 0 & 0 & d_2 & 0 & 0 & d_3 & 0 & 0 & d_4 \\ c_1 & b_1 & 0 & c_2 & b_2 & 0 & c_3 & b_3 & 0 & c_4 & b_4 & 0 \\ d_1 & 0 & b_1 & d_2 & 0 & b_2 & d_3 & 0 & b_3 & d_4 & 0 & b_4 \\ 0 & d_1 & c_1 & 0 & d_2 & c_2 & 0 & d_3 & c_4 & 0 & d_4 & c_4 \end{pmatrix} \tag{3}$$

where V is the volume of the tetrahedral element. The method of *cofactors* can be used to invert the following 4 x 4 matrix and obtain the above coefficients:

$$\begin{pmatrix} 1 & x_1 & y_1 & z_1 \\ 1 & x_2 & y_2 & z_2 \\ 1 & x_3 & y_3 & z_3 \\ 1 & x_4 & y_4 & z_4 \end{pmatrix}^{-1} = \begin{pmatrix} a_1 & a_2 & a_3 & a_4 \\ b_1 & b_2 & b_3 & b_4 \\ c_1 & c_2 & c_3 & c_4 \\ d_1 & d_2 & d_3 & d_4 \end{pmatrix} \tag{4}$$

where x_i, y_i, z_i are the nodal coordinates of the tetrahedra in the global coordinate system.

All the above matrices are constant and can be pre-computed since they depend only on the position of the mesh at the initial state. However, when large rotations occur, the linear approximation of the strain function leads to well-known inflation artifacts. Therefore a co-rotational formulation is mandatory to limit these artifacts.

2.1.2 Co-rotational formulation

The corotational formulation is widely used in computer graphics (Felippa, 2000) for its stability and the fast computation time possible thanks to precomputations, while the model is not restricted to small displacements. For this purpose, the popular corotational method computes displacements in a rotated local coordinate system.

In each element, the local frame is represented by a rigid rotation matrix $\mathbf{R}_e$, and its transpose is used to align the deformed tetrahedron with its reference, undeformed shape. The forces $\mathbf{f}_e$ are computed in the rotated coordinate system, then transformed back to the world coordinates:

$$\mathbf{f}_e = \mathbf{R}_e \mathbf{K}_e \mathbf{R}_e^T \, \mathbf{u}_e \tag{5}$$

where $\mathbf{u}_e = \mathbf{p} - \bar{\mathbf{p}}$ is the displacement vector between $\mathbf{p}$ and $\bar{\mathbf{p}}$, being respectively the current and initial positions.

The rotation matrix $\mathbf{R}_e$ associated to the element e is a 12×12 block-diagonal matrix computed as follows:

$$\mathbf{R}_e = \begin{bmatrix} \bar{\mathbf{R}}_e & & & \\ & \bar{\mathbf{R}}_e & & \\ & & \bar{\mathbf{R}}_e & \\ & & & \bar{\mathbf{R}}_e \end{bmatrix} \tag{6}$$

where $\bar{\mathbf{R}}_e$ is a 3×3 rotation matrix. Several solutions have been proposed to compute the rotation matrix, for instance by computing a polar decomposition of the nodal positions Nesme et al. (2005).

2.2 Implicit time integration

Equations used to model the dynamic behavior of bodies can be written within a synthetic formulation, given by Newton's second law:

$$\mathbf{M}\ \mathbf{a} = \mathcal{G}(t) - \mathcal{F}\ (\mathbf{p}, \mathbf{v}) \tag{7}$$

where $\mathbf{p} \in \mathbb{R}^n$ is the vector of generalized degrees of freedom (here, the mesh node positions), $\mathbf{M}$ is the mass matrix assumed constant. The function $\mathcal{G}$ gathers external forces at time t. $\mathcal{F}$ is a nonlinear function providing internal forces of the simulated object depending on its current position $\mathbf{p}$. The internal forces are derived from the physics-based deformable model described above (i.e., corotational formulation).

Let's consider the time interval $[t_i, t_f]$ whose length is $h = t_f - t_i$. We have:

$$\begin{array}{rcl} \mathbf{M}(\mathbf{v}_f - \mathbf{v}_i) & = & \displaystyle\int_{t_i}^{t_f} (\mathcal{G}(t) - \mathcal{F}(\mathbf{p}, \mathbf{v}))\, dt \\ \mathbf{p}_f & = & \mathbf{p}_i + \displaystyle\int_{t_i}^{t_f} \mathbf{v}\ dt \end{array} \tag{8}$$

To evaluate integrals $\int_{t_i}^{t_f} (\mathcal{G}(t) - \mathcal{F}(\mathbf{p}, \mathbf{v}, t))\, dt$ and $\int_{t_i}^{t_f} \mathbf{v} dt$ we chose the following implicit Euler integration scheme:

$$\begin{array}{rcl} \mathbf{M}(\mathbf{v}_f - \mathbf{v}_i) & = & h\,(\mathbf{g}_f - \mathcal{F}(\mathbf{p}_f, \mathbf{v}_f)) \\ \mathbf{p}_f & = & \mathbf{p}_i + h\mathbf{v}_f \end{array} \tag{9}$$

where $\mathbf{g}_f$ is the value of function $\mathcal{G}$ at time t_f.

We apply a Taylor series expansion and make a first-order approximation of the nonlinear function $\mathcal{F}$:

$$\mathcal{F}\ (\mathbf{p}_i + \Delta\mathbf{p}, \mathbf{v}_i + \Delta\mathbf{v}) \simeq \mathbf{f}_i + \frac{\boldsymbol{\delta}\mathcal{F}}{\boldsymbol{\delta}\mathbf{p}}\Delta\mathbf{p} + \frac{\boldsymbol{\delta}\mathcal{F}}{\boldsymbol{\delta}\mathbf{v}}\Delta\mathbf{v} \tag{10}$$

This linearization corresponds to the first iteration of the Newton-Raphson algorithm. Therefore, the problem is solved with a single iteration, assuming a temporal coherency of the mechanical behavior. It may lead to minor numerical errors in the dynamic behavior, but these errors tend to decrease at equilibrium or with null velocity.

Replacing (10) in (9) and using $\Delta\mathbf{p} = \mathbf{p}_f - \mathbf{p}_i = h\mathbf{v}_f$ and $\Delta\mathbf{v} = \mathbf{v}_f - \mathbf{v}_i$, we obtain:

$$(\mathbf{M} + h\mathbf{B} + h^2\mathbf{K})\Delta\mathbf{v} = h(\mathbf{g}_f - \mathbf{f}_i) - h^2\mathbf{K}\mathbf{v}_i \tag{11}$$

where $\mathbf{M}$ is the mass matrix, $\mathbf{f}_i$ and $\mathbf{v}_i$ are respectively the internal forces and velocities at the beginning of the time step. $\mathbf{K} = \frac{\delta\mathcal{F}}{\delta\mathbf{p}}$ is the global stiffness matrix and $\mathbf{B} = \frac{\delta\mathcal{F}}{\delta\mathbf{v}}$ is the damping matrix, used to dissipate energy.

Following Baraff and Witkin (1998), the damping matrix can be approximated with *Rayleigh damping* $\mathbf{B} \simeq \alpha\mathbf{M} + \beta\mathbf{K}$, where α is the *rayleigh stiffness* and β the *rayleigh damping*. Finally, equation (11) can be rewritten:

$$\underbrace{((1 + h\alpha)\mathbf{M} + h(h + \beta)\mathbf{K})}_{\mathbf{A}}\underbrace{\Delta\mathbf{v}}_{\mathbf{x}} = \underbrace{h(\mathbf{g}_f - \mathbf{f}_i) - h^2\mathbf{K}\mathbf{v}_i}_{\mathbf{b}} \tag{12}$$

Equation (12) provides a linear problem $\mathbf{A}\mathbf{x} = \mathbf{b}$ that must be solved at each time step. Indeed, the left-hand side is a global system matrix $\mathbf{A}$ mainly composed of the mass and stiffness matrices. Using the co-rotational formulation the stiffness matrix $\mathbf{A}$ is obtained with:

$$\mathbf{K} = \sum(\mathbf{G}_e\mathbf{R}_e\mathbf{K}_e\mathbf{R}_e^T{\mathbf{G}_e}^T) \tag{13}$$

where $\mathbf{G}_e$ is the *globalization matrix* transferring the local stiffness matrix $\mathbf{K}_e$ of an element e to the global stiffness matrix $\mathbf{K}$. The rotation matrices are not constant and must be recomputed at each simulation step. As a result, the linear system must be solved at each step. In an interactive context, the equation (12) is usually solved with the iterative Conjugate Gradient. A possible hardware-dependant optimization is to implement the full implicit integration solver on the GPU (including CG iterations) where the system matrix is not assembled and the matrix computations are parallelized directly on the original object mesh. It considerably reduces the number of operations required, and more importantly, the consumed bandwidth, enabling the method to be fast enough for highly complex stiff body simulations.

2.2.1 Preconditioner

The condition number κ of the matrix $\mathbf{A}$ measures how much the output value of the function can change for a slight change in the input argument. For heterogeneous objects or ill-structured meshes, the condition number κ is often high, which raises convergence issues for the CG algorithm. A common technique is to use a preconditioner to reduce the condition number, ensuring a faster convergence of the algorithm. By definition, a preconditioner approximates the

system matrix $\mathbf{A}$, which is less costly to invert. Solving equation (12) with a preconditioner $\mathbf{P}$ can be written:

$$\mathbf{P}^{-1}\mathbf{A}\mathbf{x} = \mathbf{P}^{-1}\mathbf{b}, \quad \text{such that} \quad \kappa(\mathbf{P}^{-1}\mathbf{A}) < \kappa(\mathbf{A}) \tag{14}$$

Several preconditioners can be used, ranging from simple diagonal matrices to accurate but costly Cholesky factorization Baraff and Witkin (1998). Generally, the performance improvement remains limited since the preconditioner is either too simple to significantly reduces the number of iterations or expensive to compute.

A faster method is to use an asynchronous preconditioning technique. The method relies on the assumption that $\mathbf{A}$ in equation (12) undergoes small perturbations between two consecutive time steps. Indeed, if $\mathbf{P}_t = \mathbf{A}_t^{-1}$ is available at a specific time t, it remains a "good" approximation for the following time steps. The preconditioner can then be updated at a low frequency on a dedicated CPU thread, and the last preconditioner available can be used to advance the simulation. Therefore, the overhead in computing the preconditioner is removed from the simulation loop, which allows using more advanced and computationally costly preconditioners such as a complete factorization of the system[1]:

$$\mathbf{P} = \mathbf{A} = \mathbf{L}\mathbf{D}\mathbf{L}^{\mathrm{T}} \tag{15}$$

where $\mathbf{D}$ is a diagonal matrix and $\mathbf{L}$ is a sparse lower-triangular matrix. An essential advantage of this factorization is that the resulting $\mathbf{L}$ matrix remains sparse, making the preconditioner application faster within the CG. Finally, the method allows simulating the deformation of homogeneous as well as heterogeneous tissues in real-time.

2.3 Time-stepping and collision detection

Collision response on mechanical objects leads to discontinuities in the velocities of the colliding points. For such discontinuous events, the acceleration is not defined: the problem belongs to the field of non-smooth mechanics.

To integrate the mechanics and the non-smooth events due to contact over time, we use a *time-stepping* method: The time step is fixed, and there is no limitation on the number of discontinuities that could take place during a time step but low-order integration schemes should be used. Although it could lead to excessive dissipation if the time step is too significant, it provides stable simulations.

2.3.1 Collision detection

Before enforcing the contact between anatomical structures or with surgical instruments (rigid or deformable), one needs to detect them (see Teschner et al. (2005) for a survey): a common strategy is to build a collision pipeline where

[1] Note that even if we compute an exact factorization of $\mathbf{A}_{t-h}$, the preconditioner remains an approximation since its computation is based on a previous configuration of the objects.

non-colliding objects will be pruned out of the pipeline and only parts of meshes that are close enough will remain.

At the end of the pipeline, collision detection returns either penetration depth or a minimal distance between meshes or pairs of primitives (between which distance constraints can be formulated to solve the contacts).

2.3.2 Contact mapping

Despite these optimizations, collision detection is a time-consuming operation in the simulation loop. To save computation time, the collision surface of the models is usually simplified compared to the FE mesh used to compute the deformation. In this case, the constraint forces can be transferred from the collision models to the FE model using a *mapping* function $\mathcal{J}$. This relation is used to map the positions $\mathbf{p}_{\text{fe}}$ of the deformable model to the positions $\mathbf{p}_{\text{col}}$ of the collision models:

$$\mathbf{p}_{\text{col}} = \mathcal{J}(\mathbf{p}_{\text{fe}}) \tag{16}$$

The most widely used mapping is to couple the collision surface (edges, triangles, or quads) with the tetrahedral mesh supporting the FE computations. In this case, a barycentric mapping is used to link the various representations. At the beginning of the simulation, each vertex of the collision models is associated with the "closest" tetrahedron in the FE mesh. During the simulation, both the collision mesh and the FE model are deformed simultaneously. Therefore, this relation remains constant over the simulation:

$$\mathbf{p}_{\text{col}} = \mathbf{J}_{\text{col}}\mathbf{p}_{\text{fe}} \tag{17}$$

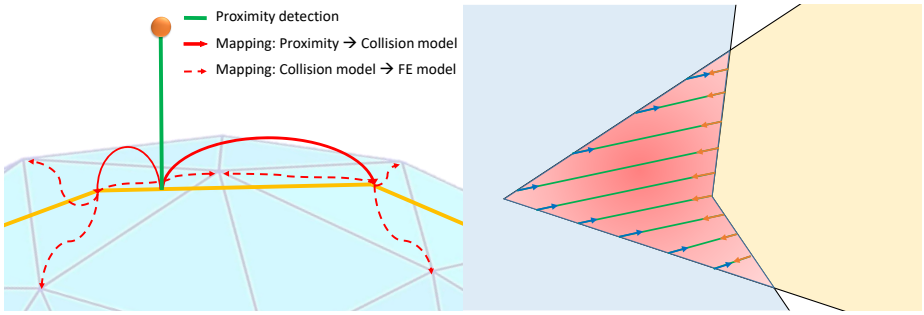
where $\mathbf{J}_{\text{col}}$ is the matrix of the *mapping* computed from the barycentric coordinates of collision points with respect to the associated tetrahedron's positions. In addition, the mapping also provides a way to transfer any given force $\mathbf{f}_{\text{col}}$ applied on the collision models to an equivalent force applied on the FE nodes. The transferred forces $\mathbf{f}_{\text{fe}}$ are then weighted with the same barycentric coefficients:

$$\mathbf{f}_{\text{fe}} = \mathbf{J}_{\text{col}}^T \, \mathbf{f}_{\text{col}} \tag{18}$$

Similarly, collision detection may provide contact information at an arbitrary location on the collision surface (for instance, in the middle of a collision triangle). Another mapping relation is, therefore, necessary to transfer contact forces to equivalent nodal forces on the collision mesh (Figure 1a). To simplify the solving process this relation is considered constant during the resolution of each time step. As previously, a mapping matrix $\mathbf{J}_{\text{CD}}$ can be defined to transfer the contact forces $\mathbf{f}_{\text{prox}}$ on the collision mesh:

$$\mathbf{f}_{\text{col}} = \mathbf{J}_{\text{CD}}^T \, \mathbf{f}_{\text{prox}} \tag{19}$$

Finally, for a given object, an arbitrary number of representation may be used and the mapping relations may be serially applied to finally compute equivalent



(a) Contact mapping (b) Constraint linearization

Figure 1: Constraint definition. (right) the constraint are linearized at the beginning of each time step providing the direction along which contact forces will be applied. (left) Contact forces are transferred from the collision model to the FE mesh using mapping relations.

forces (or displacements) on the DOFs of the FE mesh $\mathbf{f}_{\text{fe}} = \mathbf{J}_{\text{col}}^T \dots \mathbf{J}_{\text{CD}}^T \, \mathbf{f}_{\text{prox}}$. Since mapping matrices are usually sparse, this operation is inexpensive. Therefore, for simplicity, we now omit this relation considering that the constraints are directly applied to the mechanical model.

2.4 Constraint-based simulation

To avoid interpenetrations or to enforce specific behaviors, the simulation of two colliding objects 1 and 2 is now subject to a set of constraints. For this purpose, the simulation of the deformable bodies are coupled with the following equation:

$$\Lambda(\mathbf{p}_{f,1}, \mathbf{p}_{f,2}) = \mathcal{H}_1(\mathbf{p}_{f,1}) - \mathcal{H}_2(\mathbf{p}_{f,2}) \tag{20}$$

where Λ is a nonlinear function providing the violation of the constraint with respect to positions $\mathbf{p}_{f,1}$ and $\mathbf{p}_{f,2}$ of the deformable objects at the end of the time steps, and $\mathcal{H}_1$, $\mathcal{H}_2$ provide the value of the constraints. In the case of collisions, Λ corresponds to the shape of the interpenetration (red area in figure 1b) and must be canceled at the end of each time step. This function is highly nonlinear since constraint forces must be applied to cancel the violation. However, applying such forces will result in displacements, rotations, or deformations of deformable bodies that may generate new collision areas or invalidate the previous ones.

The constraint equations are linearized using a first-order Taylor expansion:

$$\mathcal{H}(\mathbf{p}_f) = \mathcal{H}(\mathbf{p}_i + h\mathbf{v}_f) \approx \mathcal{H}(\mathbf{p}_i) + h\frac{\partial \mathcal{H}(\mathbf{p}_i)}{\partial \mathbf{p}}\mathbf{v}_i + h\frac{\partial \mathcal{H}(\mathbf{p}_i)}{\partial \mathbf{p}}\Delta\mathbf{v} \tag{21}$$

where $\frac{d\mathcal{H}(\mathbf{p}_i)}{d\mathbf{p}}$ is the linearization of constraints at the begining of the time step (using positions $\mathbf{p}_i$).

In practice, the evaluation and the linearization of the constraints equations are complex. For this purpose, collision detection is performed, providing a discrete set of proximities between both objects (green lines in figure 1b). The number of proximities usually depends on the resolution of the collision mesh (that may

be adapted thanks to the *mappings*) and the collision detection method itself (for instance, by filtering proximities afterward).

Once the collision information is available, all the constraint equations are evaluated along with the directions of the proximities assumed constant for the time step. This leads to several simplifications: First of all, $\mathbf{H} \approx \frac{d\mathcal{H}(\mathbf{p}_i)}{d\mathbf{p}}$, known as the *Jacobian of the constraints*, can be defined[2], providing the constraint directions (blue and orange arraows in figure 1b). The dimension of $\mathbf{H}$ is $c \times n$ where c is the number of proximities, and n is the number of DOFs. Similariliy, the violation of the contraints $\mathcal{H}(\mathbf{p}_i) \approx \mathbf{H}\mathbf{p}_i$ is evaluated along the same constraint directions. With these simplification and replacing (21) in (20) the violation of the constraint at the end of the step can be rewritten as:

$$\boldsymbol{\delta}_f = \underbrace{(\mathbf{H}_1\mathbf{p}_{i,1} + h\mathbf{H}_1\mathbf{v}_{i,1}) + (\mathbf{H}_2\mathbf{p}_{i,2} + h\mathbf{H}_2\mathbf{v}_{i,2})}_{\boldsymbol{\delta}_i} + h\ \mathbf{H}_1\Delta\mathbf{v}_1 + h\ \mathbf{H}_2\Delta\mathbf{v}_2 \quad (22)$$

where $\boldsymbol{\delta}_f$ is the violation of the constraints at the end of the time step and $\mathbf{H}_1 = \frac{d\mathcal{H}_1(\mathbf{p}_{i,1})}{d\mathbf{p}}$ and $\mathbf{H}_2 = -\frac{d\mathcal{H}_2(\mathbf{p}_{i,2})}{d\mathbf{p}}$.

In addition, to enforce non-penetrating objects, constraint forces $\boldsymbol{\lambda}$ (also called Lagrangian Multipliers) are applied along the direction of the constraint. The governing equation (7) is modified as follows:

$$\mathbf{M}\ \mathbf{a} = \mathcal{G}(t) - \mathcal{F}\ (\mathbf{p}, \mathbf{v}) + h\mathbf{H}^T\ \boldsymbol{\lambda} \quad (23)$$

After the linearization, equation (12) is modified as follows:

$$\mathbf{A}\mathbf{x} = \mathbf{b} + h\mathbf{H}^T\ \boldsymbol{\lambda} \quad (24)$$

Gathering equations (24) and (22) for the two interacting objects results in the following Karush-Kuhn-Tucker (KKT) system:

$$\begin{cases} \mathbf{A}_1\mathbf{x}_1 + h\mathbf{H}_1^T\boldsymbol{\lambda} = \mathbf{b}_1 & (25) \\ \mathbf{A}_2\mathbf{x}_2 + h\mathbf{H}_2^T\boldsymbol{\lambda} = \mathbf{b}_2 & (26) \\ h\mathbf{H}_1\mathbf{x}_1 + h\mathbf{H}_2\mathbf{x}_2 = \Delta\boldsymbol{\delta} & (27) \end{cases}$$

where $\Delta\boldsymbol{\delta} = \boldsymbol{\delta}_f - \boldsymbol{\delta}_i$

2.4.1 Contact and friction models

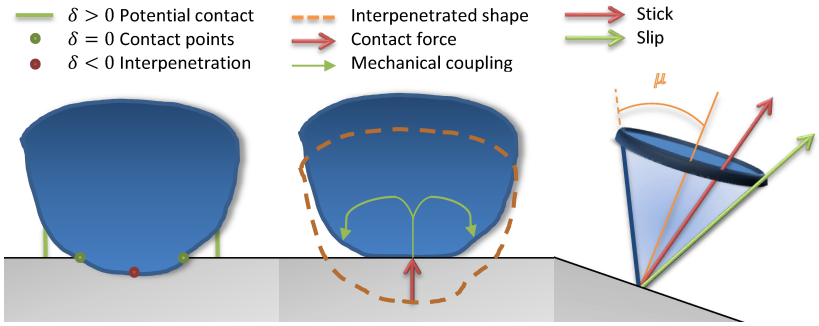
Contact equations are subject to Signorini's law (Figure 2a). It expresses that there is a complementarity[3] relation between the contact force $\boldsymbol{\lambda}_n$ and the distance $\boldsymbol{\delta}_n$, along the direction $\mathbf{n}$ of the contact:

$$\mathbf{0} \leq \boldsymbol{\delta}_\mathbf{n} \perp \boldsymbol{\lambda}_\mathbf{n} \geq \mathbf{0} \quad (28)$$

[2]Note that the matrix may be multiplied by mapping matrices, but this operation is ommited as stated above.

[3]Complementarity is noted $\perp$. It states that one of the two values $\boldsymbol{\delta}_n$ or $\boldsymbol{\lambda}_n$ must be null whereas the other is positive.

This model has several physical justifications including non-interpenetration and no sticking force. Moreover the contact force vanishes if the points are not strictly in contact. However, using Signorini's law, the contact force creates a frictionless response.



(a) Signorini's law (b) Mechanical coupling (c) Coulomb friction

Figure 2: Illustration of the physical phenomena simulated for the contact's response.

Coulomb's friction law describes the macroscopic behavior of friction in the tangent contact space (Figure 2c). In this law, the reaction force is included in a cone whose height and direction are given by the normal force. If the reaction force is strictly included inside the cone, objects stick together; otherwise, the reaction force is on the cone's border, and objects are slipping along the tangential direction. In this last case, the friction force must be directed along the direction of motion:

$$\begin{aligned} &\dot{\boldsymbol{\delta}}_{\mathbf{T}} = \mathbf{0} \Rightarrow \|\boldsymbol{\lambda}_{\mathbf{T}}\| < \mu \ \|\mathbf{f}_{\mathbf{n}}\| \quad \text{(stick)} \\ &\dot{\boldsymbol{\delta}}_{\mathbf{T}} \neq \mathbf{0} \Rightarrow \boldsymbol{\lambda}_{\mathbf{T}} = -\mu \ \|\boldsymbol{\lambda}_{\mathbf{n}}\| \frac{\dot{\boldsymbol{\delta}}_{\mathbf{T}}}{\|\dot{\boldsymbol{\delta}}_{\mathbf{T}}\|} = -\mu \ \|\boldsymbol{\lambda}_{\mathbf{n}}\| \ \mathbf{T} \quad \text{(slip)} \end{aligned} \tag{29}$$

where μ is the friction parameter, and $\mathbf{T}$ is the direction of motion in the tangential plane to the contact normal $\mathbf{n}$.

During 3D slipping motion (also called *dynamic friction*), the tangential direction is unknown. Indeed, the tangential velocity depends on the tangential force that has to be found during the solving process. It creates a non-linearity equation in addition to the complementarity state stick/slip.

Signorini's law and Coulomb's law are also valid in a *multi-contact* case. This model is defined for each *potential* contact provided by the contact detection algorithm. However, to solve these laws at every contact point, we have to consider the coupling that exists between these contact points (Figure 2b). This coupling comes from the intrinsic mechanical behavior of deformable objects.

2.4.2 Constraint solving and mechanical coupling

When objects are distant, the mechanical matrices $\mathbf{A}_1$ and $\mathbf{A}_2$ are not coupled and can be solved independently. However, coupling terms must be added to take the constraints into account.

Compared to penalty-based methods, a significant advantage of relying on Lagrange multipliers is that collision events never modify the mechanical matrices. It is therefore possible to rely on the *schür complement* method to solve the equation (25), (26) and (27) with the 5 following steps:

Step 1: interacting objects are solved independently while setting $\boldsymbol{\lambda} = \mathbf{0}$. A set of independent linear systems of equations $\mathbf{Ax} = \mathbf{b}$ must then be solved for each object. The Conjugate Gradient algorithm is generally proposed to solve this problem since $\mathbf{A}$ is large, sparse, symmetric, and positive definite. It provides $\mathbf{x}_1^{\text{free}} = {\mathbf{A}_1}^{-1}\mathbf{b}_1$ and $\mathbf{x}_2^{\text{free}} = {\mathbf{A}_2}^{-1}\mathbf{b}_2$ called the *free motion* for each object, corresponding to the velocities update without considering any constraint nor collision. After integration, we obtain $\mathbf{p}_1^{\text{free}}$ and $\mathbf{p}_2^{\text{free}}$:

$$\mathbf{p}_1^{\text{free}} = \mathbf{p}_{i,1} + h\mathbf{v}_{i,1} + h\mathbf{x}_1^{\text{free}} \tag{30}$$

$$\mathbf{p}_2^{\text{free}} = \mathbf{p}_{i,2} + h\mathbf{v}_{i,2} + h\mathbf{x}_2^{\text{free}} \tag{31}$$

Step 2: the constraint laws are linearized during the time step. The linearization corresponds to the collision detection that provides the proximities of potential contacts and their associated normals between the position $\tilde{\mathbf{p}}_1$ and $\tilde{\mathbf{p}}_2$ of the collision meshes. Constraint equations are then transferred to the mechanical DOFs using the mapping relation (18), providing the Jacobian of the constraint $\mathbf{H}_1$ and $\mathbf{H}_2$ that are assumed constant during the time step. The collision detection is performed at the *beginning* of the time step, whereas the violation $\boldsymbol{\delta}^{\text{free}}$ is instead computed using the *free motion* $\mathbf{p}_1^{\text{free}}$ and $\mathbf{p}_2^{\text{free}}$ obtained at the previous step.

Step 3: The violation of the constraint is evaluated, and constraints are defined according to the collision proximities. With Euler equation one may note that:

$$\Delta\boldsymbol{\delta} - \mathbf{H}_1\mathbf{x}_1^{\text{free}} - \mathbf{H}_2\mathbf{x}_2^{\text{free}} = \boldsymbol{\delta}_f - \mathbf{H}_1\mathbf{p}_1^{\text{free}} - \mathbf{H}_2\mathbf{p}_2^{\text{free}} \tag{32}$$

Replacing (25) and (26) in (27) gives:

$$\underbrace{\left[\mathbf{H}_1\mathbf{A}_1^{-1}\mathbf{H}_1^T + \mathbf{H}_2\mathbf{A}_2^{-1}\mathbf{H}_2^T\right]}_{\mathbf{W}}\boldsymbol{\lambda} = \underbrace{\mathbf{H}_1\mathbf{p}_1^{\text{free}} - \mathbf{H}_2\mathbf{p}_2^{\text{free}}}_{\boldsymbol{\delta}_f - \boldsymbol{\delta}^{\text{free}}} \tag{33}$$

where $\mathbf{W}$ is known as the *compliance matrix*, or the *Delasus operator*.

In the above equation, both $\boldsymbol{\lambda}$ and $\boldsymbol{\delta}$ are unknown. For Signorini's law (equation (28)), this equation describes an LCP (Linear Complementarity Problem). If it is combined with Coulomb's law (equation (29)), we obtain an NLCP (Nonlinear Complementarity Problem). An important difficulty lies in the fat that this equation involves evaluating the inverse of large matrices $\mathbf{A}_1$ and $\mathbf{A}_2$ (same dimension as the number of DOFs) to compute the *compliance matrix* $\mathbf{W}$.

Step 4: We obtain the value of $\boldsymbol{\lambda}$ using a Gauss-Seidel algorithm dedicated to the NLCP created by contact and friction equations. Considering a contact

α, among m instantaneous contacts, one can write the behavior of the model in contact space:

$$\underbrace{\boldsymbol{\delta}_\alpha - \mathbf{W}_{\alpha\alpha}\ \boldsymbol{\lambda}_\alpha}_{\text{unknown}} = \underbrace{\sum_{\beta=1}^{\alpha-1} \mathbf{W}_{\alpha\beta}\ \boldsymbol{\lambda}_\beta + \sum_{\beta=\alpha+1}^{m} \mathbf{W}_{\alpha\beta}\ \boldsymbol{\lambda}_\beta}_{\text{frozen}} + \ \boldsymbol{\delta}_\alpha^{\text{free}} \tag{34}$$

where $\mathbf{W}_{\alpha,\beta}$ is a compliance matrix modeling the mechanical coupling between contact points α and β. For each contact α, this method solves the contact equations by considering the other contact points ($\alpha \neq \beta$) as "frozen". The new value of $\boldsymbol{\lambda}_\alpha$ is given by solving Signorini's law and Coulomb's law on this contact (see Duriez et al. (2006) for details of implementation and performance).

Step 5: When the value of $\boldsymbol{\lambda}$ is available for all the contacts, the corrective motion is computed:

$$\begin{aligned} \mathbf{p}_{1,t+h} &= \mathbf{p}_1^{\text{free}} + h\Delta\mathbf{v}_1^{\text{cor}} \quad \text{with} \quad \Delta\mathbf{v}_1^{\text{cor}} = \mathbf{A}_1^{-1}\mathbf{H}_1^T\boldsymbol{\lambda} \\ \mathbf{p}_{2,t+h} &= \mathbf{p}_2^{\text{free}} + h\Delta\mathbf{v}_2^{\text{cor}} \quad \text{with} \quad \Delta\mathbf{v}_2^{\text{cor}} = \mathbf{A}_2^{-1}\mathbf{H}_2^T\boldsymbol{\lambda} \end{aligned} \tag{35}$$

We finally obtain $\mathbf{p}_{1,t+h}$ and $\mathbf{p}_{2,t+h}$, the positions of objects 1 and 2 that fulfills the contact and friction laws.

2.4.3 Compliance and mechanical coupling

In the following, the term "*mechanical coupling*" describes the coupling between contact constraints applied on two subsets of the boundary of a deformable body. This coupling occurs through the deformation of the body itself. Indeed, even if the contact points are only defined on the boundary of the deformable bodies, they are all influenced by each other through the stiffness of the material, which is represented by $\mathbf{W}$ in equation (33).

Formally speaking, the *compliance matrix* $\mathbf{W}$ requires the computation of $\mathbf{A}^{-1}$, which is a large matrix (same dimension as the number of DOF) and changes at each time step. Although computing this inverse in real-time is possible for coarse models, it raise significant difficulties for large meshes in real-time.

A possible approximation of the *compliance matrix* $\mathbf{W}$ is to use the asynchronous preconditioner of the previous section:

$$\mathbf{W} = \mathbf{H}\mathbf{A}^{-1}\mathbf{H}^{\mathrm{T}} \approx \mathbf{H}\mathbf{P}^{-1}\mathbf{H}^{\mathrm{T}} = \mathbf{H}\big(\mathbf{R}\,\mathbf{L}\mathbf{D}\mathbf{L}^{\mathrm{T}}\mathbf{R}^{\mathrm{T}}\big)^{-1}\mathbf{H}^{\mathrm{T}} \tag{36}$$

Indeed, since $\mathbf{P}$ is an efficient preconditioner (ie.e a close approximation of the factorization of $\mathbf{A}$), we propose to use it to compute $\mathbf{W}$ in equation (33). The above equation requires computing the product of the inverse of the preconditioner with the Jacobian of contacts $\mathbf{H}$, which can be achieved by computing columns independently of the following matrix:

$$\mathbf{L}\mathbf{D}\mathbf{L}^{\mathrm{T}}\ \mathbf{X}_i = \mathbf{H}_i \quad \Leftrightarrow \quad \mathbf{X}_i = \big(\mathbf{L}\mathbf{D}\mathbf{L}^{\mathrm{T}}\big)^{-1}\mathbf{H}_i \tag{37}$$

Solving this equation is still expensive since multiple triangular systems must be solved for each right-hand side vector. Again since multiple right-hand side vectors stored in $\mathbf{H}$ can be computed independently, GPU optimizations could be considered by assigning the computation of each column to an independent multiprocessor on the GPU.

Then we use another level of parallelism where each STS is solved with several threads. Indeed, many data can potentially be treated in parallel during the solving process of each STS. This two-level strategy fits the GPU architectures where local synchronizations within a group of threads are fast, whereas global synchronizations over multiple groups are costly.

3 Registration of biomechanical models

Medical imaging is one of the main improvements of modern surgical practice and represents not only a support for diagnostics but also an actual operative instrument during therapeutic procedures. However, from a physiological point of view, organs are not static, and even the most straightforward breathing motion can induce shape deformations which requires a mental effort of three-dimensional reconstruction to coincide with the pre-operative images. However, as an approximation, FE models cannot be directly used for medical purpose.

Yet, in order to limit errors, FE models can be registered with intraoperative data extracted from medical images. In this case, biomechanical models can be used to display through Augmented Reality (AR) the internal structures (vessels, tumors...) on top of the intraoperative images that are often sparse, incomplete, and of poor quality. The advantages of relying on FE models for this task are multiple: FE models can be used to regularize the data extracted from images, being this way less sensitive to noise and tracking errors. In addition, FE models provide accurate solutions to interpolate the whole volume displacement of the organs, including the displacement of internal structures such as tumors or vessels.

In this section, we introduce a unified constraint-based formalism and solving process combining intraoperative image data with biomechanical models.

Problem definition: The non-rigid registration of a preoperative 3D model S with intra-operative data is an ill-posed problem known to be very difficult to solve. It can be formalized as finding the 3D non-rigid transformation $\mathcal{T}$ mapping S into the image frame. However, since intraoperative data are usually incomplete, errors can only be measured with the information that can practically be extracted from the operating scene (image or sensors). Let $\mathbf{m}$ be the set of data extracted intraoperatively, the non-rigid registration problem can be synthesized with:

$$\min_{\mathbf{p}}(||\overline{\mathcal{T}(S)} - \mathbf{m}||) \tag{38}$$

where $\mathbf{p}$ are the positions of the model S, and $\overline{\mathcal{T}(S)}$ corresponds to the transformed model observable in the same condition as the intraoperative data.

This process usually admits a unique solution for rigid transformation scenarios (without considering noise or other pairing issues). However, the rigid assumption is not valid for our purpose, and the deformations generate a problem having an infinite number of solutions.

To decrease the number of solutions, we assume an elastic behavior of the organ with preservation of the volume after the deformation. Therefore, we propose to rely on a non-linear biomechanical Finite Element model coupled with a set of constraints $\mathcal{H}$ to solve the registration problem of equation (38). This leads to a non-linear problem whose solution is given by the positions of the physical model providing the equilibrium between the internal forces and the external forces applied by the constraints:

$$\begin{cases} \mathcal{F}(\mathbf{p}) + \mathcal{H}(\mathbf{m}, \mathbf{p}) = \mathbf{0} & (39) \\ \mathcal{H}(\mathbf{m}, \mathbf{p}) = \boldsymbol{\delta} & (40) \end{cases}$$

where $\mathcal{F}$ provides the internal forces of the FE model, and $\boldsymbol{\delta}$ is the violation of the constraints, as described in the previous section. Lagrangian Multipliers $\boldsymbol{\lambda}$ are used to impose displacements and enforce constraints of equation (40). The number and the direction of Lagrangian Multipliers will be described below; it depends on the position of the models $\mathbf{p}$ and the data extracted from intraoperative images $\mathbf{m}$.

Note that equation (39) and (40) corresponds to the static equilibrium of the dynamic equation (23). However, in this section, the simulation time t is not considered anymore. Instead, the simulation steps correspond to the iterations of the iterative registration procedure.

Segmentation and meshing and initialization: Preoperatively, a patient-specific model is built from the images acquired before surgery. The model is usually constructed using either contrast-enhanced CT data or volume MRI sequence. The image is first segmented (using, for instance, the semi-automatic methods available in ITKSnap[4]), then a tetrahedral mesh is obtained from the segmented maps (for instance, using CGAL[5]). Relying on the numerical strategies described above, the target resolution of the FE mesh is around 2000 nodes per organ, to enforce real-time compatible computation times.

3.1 Geometrical binding

Vision-based algorithms are traditionally used to extract geometrical primitives (2D/3D positions, partial triangular mesh...) from intraoperative images. These data provide the intraoperative shape (called **target**) on which the preoperative model (called **source**) must be registered.

A significant difficulty is related to the fact that it is complicated to relate the position of the resulting data to the FE model. Indeed, the correspondences between image data and material points on the model are usually not known.

[4] www.itksnap.org
[5] www.cgal.org

This problem is addressed with an iterative process taking as input the **target** data $\mathbf{m}$ and the **source** positions $\mathbf{p}$ of the *collision* mesh[6].

As previously, the constraint equations are linearized at each iteration $\mathcal{H}(\mathbf{p}, \mathbf{m}) \simeq \frac{\partial \mathcal{H}}{\partial \mathbf{p}} = \mathbf{H}$, providing the directions along which Lagrangian Multipliers are computed. The linearization (corresponding to the collision detection in the previous section) corresponds to the pairing operation between the control points $\mathbf{m}$ and material points on the model. The *iterative closest point* (ICP) method Rusinkiewicz and Levoy (2001) is employed to associate the target points with their respective closest points on the source mesh. The linearization is performed at the beginning of the simulation step (corresponding to the iteration t) using the current position of the model and assumed constant during the solving process.

3.1.1 Outliers and geometrical filtering:

The quality of the registration strongly depends on the quality of the pairings performed during the binding process. Indeed, incoherent or antagonistic pairings (called outliers) can appear due to the noise (image artifacts, missing data...) and must be ignored. Depending on the application, the following filters may be used:

Unique pairing: First, we enforce that control points $\mathbf{m}$ are associated with a unique element on the model; otherwise, it would lead to over-constrained problems. When several vertices of $\mathbf{m}$ are projected onto the same geometrical element of the mesh, only the nearest one is kept.

Filtering with distance: A pairing is considered as an outlier, and then ignored, if it does not satisfy the following condition:

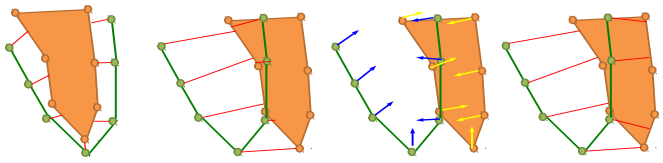
$$\mathbf{d}_j \in [\tilde{\mathbf{d}} - \mathbf{d}_t; \tilde{\mathbf{d}} + \mathbf{d}_t] \tag{41}$$

where $\tilde{\mathbf{d}}$ is the median distance computed over all pairings and $\mathbf{d}_t$, a threshold. In addition, a filtering criterion $\mathbf{d}_{\max}$ can be added in order to reject pairing above a given distance $\mathbf{d}_{\max}$.

Filtering with normals: When image data are sufficiently dense, the normals of the surface can be estimated using a simple least square plane fitting as implemented in the Point Cloud Library[7]. In this case, the method can be improved by ensuring that the normals associated with image data have the same orientation as the FE surface (Figure 3).

[6] Note that the search is usually performed with positions $\tilde{\mathbf{p}}$ of a dedicated *collision* model, either to save computation time or to have a specific representation of the structures on which constraints will be applied (vessels, contours...). However, as explained in the previous section, a linear relation $\mathbf{J}$ can be precomputed using barycentric coordinates (see equations (17) and (18)) to transfer the constraint equations to the mechanical DOFs $\mathbf{p}$. For the sake of simplicity, we omit this relation, considering that the constraints are directly applied to the mechanical model.

[7] https://pointclouds.org/documentation/tutorials/normal_estimation.html



(a) (b) (c) (d)

Figure 3: Binding process of the control points and the FE structure. 3a the control points (green) are associated (red lines) with the closer surface of the object (orange). 3b the distance criterion is not sufficient since all the control points are associated with the same side of the object and the constraints cannot be satisfied. 3c, a set of normals aiming inside the surface are computed. 3d the control points are associated with the closer triangle whose normal is oriented in the same direction.

3.2 Constraints definition

This section introduces the various constraints employed for our applications. At this stage, the control points $\mathbf{m}$ of the **target** shape are associated with material positions $\mathbf{p}$ on the biomechanical model. Lagrange multipliers are used to impose constraint forces on the FE model to cancel the violation of constraints.

3.2.1 Bilateral constraints

Bilateral Constraints are holonomic constraints $\boldsymbol{\Omega}(\mathbf{p}) = \mathbf{0}$, used to impose the relative positions of constrained points on the orthogonal plane defined by the constraint normal. For a given control point i, the violation of constraints $\boldsymbol{\delta}_i$ is computed with:

$$\boldsymbol{\delta}_i = \mathbf{n}_i \cdot (\mathbf{m}_i - \mathbf{p}_i^{\text{free}}) \tag{42}$$

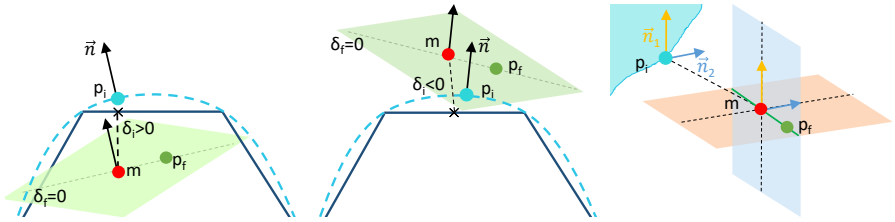
with $\cdot$ being the Euclidean dot product and $\mathbf{n}_i$ is the normal of the constraint.

Depending on the desired behavior, $\mathbf{n}_i$ may be computed in several ways. In can be either the normal of the source's surface at position $\mathbf{p}_i$, or the direction of the pairing $\mathbf{n}_j = \frac{\mathbf{m}_i - \mathbf{p}_i}{||\mathbf{m}_i - \mathbf{p}_i||}$, or simply the unit vectors $\vec{x}, \vec{y}$ or $\vec{z}$.

At the end of each simulation step, the corresponding constraint force $\boldsymbol{\lambda}_i$, applied in the direction of $\mathbf{n}_i$, is computed so that no violation remains ($\bar{\boldsymbol{\delta}}_i = 0$) in that direction. It means that the vertex i, with position $\mathbf{p}_i$, will be located on the tangential plane given by $\mathbf{n}_i$ (Figure 4). The combination of two bilateral constraints forces the points to follow a line at the intersection of the two planes (Figure 4b). Similarly, three orthogonal bilateral constraints force the point $\mathbf{p}_i$ to be located at the exact 3D location $\mathbf{m}_i$ at the end of the step. As it will be introduced in the next section, the possibility to "slide" on the surface of the FE mesh is essential to stabilize the model around the configuration minimizing the energy and satisfying the constraints.

3.2.2 Unilateral constraints

Unilateral constraints are used to simulate collision on the source mesh. It can be either collision with other FE models, the surrounding environment, or even collision with medical instruments segmented in the images. Unilateral



(a) Single bilateral constraint (b) Two bilateral constraints

Figure 4: The broken blue lines represent the Bezier surface defined over the triangles of the source mesh. $\mathbf{m}$ (in red) are paired with positions $\mathbf{p}_i$ (in blue) at the beginning of the step. After the resolution, $\mathbf{p}_f$ (in green) must be located on the tangential plane defined by the constraints. Note that $\mathbf{m}$ does not move, but instead, the model is deformed (assuming linear translations of the solution plane) to enforce the constraint.

constraint satisfies the Signorini conditions $\boldsymbol{\lambda} \perp \boldsymbol{\delta}$ as explained in the previous section (see section 2.4.1). The violation is computed the same way as in equation (42). At the end of the resolution, the constraint node i is located on the positive side of the constraint.

3.3 Constraint Solving

The linearization of equations (39) and (40), provides the following KKT system in a quasi-static scenario:

$$\begin{cases} \mathbf{A}\mathbf{x} + \mathbf{H}^T\boldsymbol{\lambda} = \mathbf{0} & (43) \\ \mathbf{H}\mathbf{x} = \boldsymbol{\delta} & (44) \end{cases}$$

Note that although we only consider static equations for the registration, the dynamic terms of equation (12) are usually conserved during the registration process, i.e. $\mathbf{A} = (1 + h\alpha)\mathbf{M} + h(h + \beta)\mathbf{K}$ as in equation (12). Indeed, dynamic terms increase the stability of the registration procedure, and since we only consider the converged state (i.e., when the velocity update $d\mathbf{v} = \mathbf{0}$) without gravity or external forces, the solution of the dynamic and static equations are identical.

A new pairing procedure is called at each step to associate the control points $\mathbf{m}_i$ with their respective closest point $\mathbf{p}_i$ on the model. For each control point, a unique bilateral constraint is defined to perform the registration. Although for a single constraint, this would move the point $\mathbf{p}_i$ to the exact location $\mathbf{m}_i$, it is not necessarily the case for multiple constraints. Indeed, when several constraints are applied simultaneously on the model, the constraints are coupled through the *compliance matrix* $\mathbf{W}$. Therefore, at the end of the simulation step, $\mathbf{p}_i$ and $\mathbf{m}_i$ are located on the tangential plane defined by the normal $\mathbf{n}_i$. However, the cumulative effect of other constraints may move the points in lateral directions, allowing the model to "slide" toward the shape, minimizing the energy necessary to enforce the constraints.

During the Gauss-Seidel iterations, the constraint equations are either activated with a non-zero force or deactivated if the violation is canceled by other con-

straints. As a result, only the constraints necessary to cancel the violation are activated, and $\boldsymbol{\lambda}$ minimizes the energy to register the model. The following section will show that this observation can be used to identify the boundary conditions acting between the preoperative and intraoperative configurations. Finally, the iterative registration process is stopped when the deformable body is stabilized, i.e. when the variation of velocity between consecutive iterations t and $t+1$ is lower than a threshold $||\mathbf{x}^{(t+1)} - \mathbf{x}^{(t)}|| < \epsilon$.

3.3.1 Outliers and mechanical filtering:

The geometrical filters introduced previously may not be sufficient. Additional mechanical filters are employed to stabilize the process and facilitate the convergence.

Force clipping: During the solving phase, the constraint forces are computed as follows:

$$\boldsymbol{\lambda} \in [-\mathbf{f}_{max}; \mathbf{f}_{max}] \tag{45}$$

with $\mathbf{f}_{max}$ a parameter defining the maximal force applied on the constraints. Since the constraint forces are bounded, all the constraints might not be satisfied at the end of the resolution ($\bar{\boldsymbol{\delta}} \neq 0$). However, the cumulative effect of all the constraints will create the general motion/deformation of the model. Strong forces are only necessary if the registration constraints require significant energy to act against the internal elastic forces of the deformable body, which is most of the time attributed to outliers.

Image compliance: In the above formulation, the data extracted from images are never impacted by the biomechanical model, but instead, the model deformed to fit the data. However, as stated in the introduction, the data may be noisy, and their positions are subject to inaccuracy.

A compliance factor is associated with the constraints to improve the robustness of the method:

$$(\mathbf{W} + \mathbf{W}^{\text{soft}})\, \boldsymbol{\lambda} = \boldsymbol{\delta} \tag{46}$$

where $\mathbf{W}^{\text{soft}}$ is a diagonal matrix whose coefficients w_{soft} are chosen according to the mechanical parameters of the model and the confidence of the tracking method. If $w_{\text{soft}} = 0$, image data $\mathbf{m}$ are not influenced by the constraints, whereas the higher w_{soft}, is chosen, the more the image data will be projected onto the model rather than deforming it. Therefore, both the biomechanical model and the data $\mathbf{m}$ can be deformed during the constraints resolution.

4 Applications

Having a simulation that can handle the deformations of the anatomical structures, the collisions between tissues and surgical tools, and mechanical constraints that can fit simulation data to input (medical images), many medical applications can be designed Courtecuisse et al. (2014); Trivisonne et al. (2016); Courtecuisse et al. (2020). In this book chapter, we focus on an application that will ease the diagnoses of pelvic pathologies by adding an augmented reality set-up to 2D MRI images. 2D dynamic MRI sequences are commonly

used in clinical routines to evaluate the dynamic of organs. However, due to the limited view, subjectivity related to human perception cannot be avoided in the diagnoses. Consequently, the aim is to provide better knowledge and understanding of pathologies such as prolapsus or abnormal mobility of tissues. Our application will handle the four main structures of the female pelvic floor (bladder, vagina, uterus, and rectum).

4.0.1 Data acquisition

Patient-specific data are generated from two data sets of two volunteers (designed as T005 and T007) presenting no pelvic pathology. The experimental data inputs have been acquired following a close protocol to a standard clinical routine (see fig. 5).

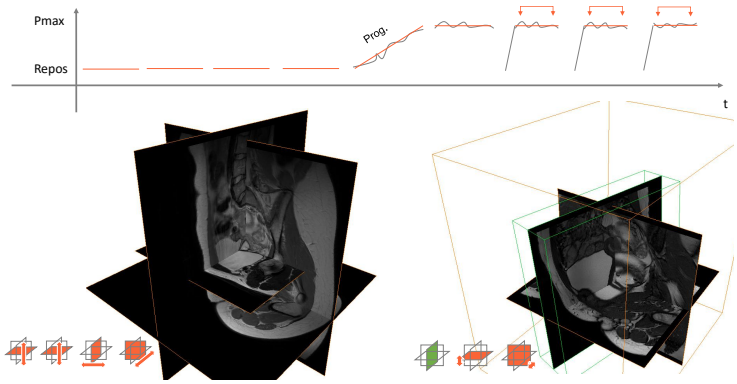


Figure 5: Protocol for MRI data acquisition composed of three steps. TSE: Volume acquisition is performed providing the geometry of organs. DYN: a dynamic 2D sequence is acquired during breathing and contraction of muscles. FFE: A sparse volume acquisition is performed for the validation, while the volunteer maintains a maximal pressure of organs.

Pelvic MRI requires between 20 and 50 minutes to obtain static volume images with sufficient organ resolution (TSE). The procedure is followed by a dynamic acquisition (DYN) while the patient contracts and relaxes the pelvic muscles. The DYN sequence is performed in the middle plane, representing organ motion the most. Data were obtained with a 3 Tesla MRI (Philips Achieva 3.0T TX), providing up to 2 of images per second in dynamic mode. TSE and DYN are already part of standard clinical routines, but a final step (FFE) is added to assess the accuracy and validate our registration method. Volunteers are asked to maintain a maximal contraction of the pelvic muscles without moving or breathing. It significantly limits the time allocated for acquisition, and 15 of sparse sagittal slices uniformly distributed in the volume were acquired. Although the FFE is not required to perform the registration, it provides the only data available to assess the accuracy of our approach.

4.1 Contour tracking

The problem of outline organs' tracking in DYN MR images is ill-defined because it requires finding tangential information that may vary due to off-plane motion.

To follow material points over time, especially to avoid drifting along the contour boundaries, we propose to apply additional constraints for the computation of these points to track the material points better and to ensure the plausibility of the computed motion.

Based on the initial B-spline model, each organ is redefined by a composite Bézier curve (a series of segments attached end to end). The beginning and ending points of Bézier segments are considered material points, selected automatically based on the criteria of high curvature. The "corner" points are identifiable during the deformation over time. Then the conversion from B-spline to composite Bézier is carried out using a least squares minimization.

The whole organ contour is initialized on the first image. Then, the contour is deformed automatically to fit the other images in the DYN sequence. The material points are first registered from image t to $t+1$ using an Optical Flow method. Then, the energy E_{image} is minimized for each Bézier segment in the neighborhood of the curve. The energy is evaluated from the gray image level, the normal direction of the Bézier curve, and a *level set* function (more details can be found in Jiang et al. (2015)).

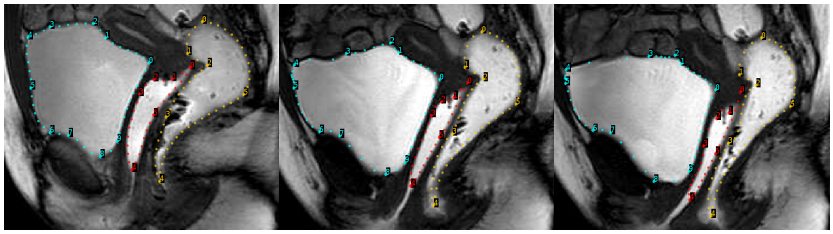


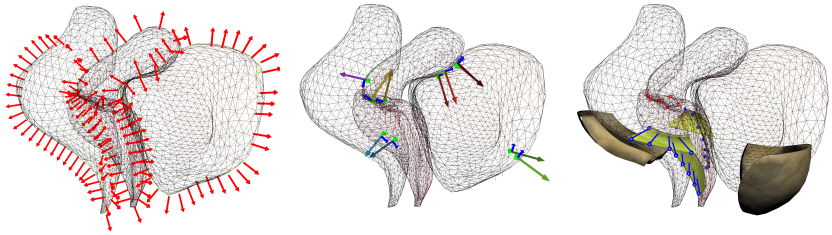
Figure 6: Outline's tracking with material points in the sequence of dynamic MR images.

Figure 6 shows the semi-automatic tracking of three highlighted organs with material points: bladder (red), vagina (blue) and rectum (green). Concerning the mobility of different areas, the distribution of points matches the organ shape and can provide more physical information for the following 3D registration.

4.2 Boundary condition

The original contribution was to combine the 3D FE models, image data, and a priori knowledge of boundary conditions. Boundary conditions include contacts between structures and an approximative definition of ligaments and attached structures (Figure 7).

According to the knowledge of anatomy, anatomical constraints are added to limit organs' off-plane motion (see fig. 7, center). The vagina is attached on each side by ligaments modeled as stiff springs (blue), allowing vertical mobility but preventing lateral motions. The uterus is attached to the vagina through bilateral constraints (red), linking the displacement of both structures. In addition, fasciae constrain lateral motions of the bladder and the rectum on each side. Friction contacts are taken into account to limit the mobility of the tissues. To not generate unsolvable problems (where the outline of the organs overlaps



(a) Outline (b) Contact (c) Anatomical

Figure 7: Constraints set used in the simulation: Outline constraints (left), Contact Constraints (center), anatomical constraints (right).

fixed faciae geometries), these structures are considered deformable with fixed constraints on their extremity. Therefore, faciae remains in contact with tissues preventing lateral motion thanks to contact simulation, while deformations are allowed to satisfy image constraints.

The exact patient-specific modeling of these conditions is complex (since boundary conditions are usually not visible in medical images). Therefore the precise location and geometries of fasciae are unknown and may change due to patient motions. However, in the next section we show that even such approximate information significantly improves the registration accuracy.

4.3 Evaluation

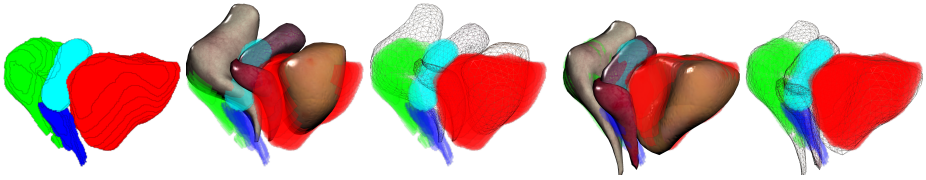
Since Boundary conditions and mechanical parameters are not accurately known for patient specific data, all the following simulations have been run 100 times, where the mechanical parameters of each model have been randomized within a range of 10% around the values given in the table 1. In addition, the location of springs (used to attach the vagina and model ligaments) have been randomized around the ideal position segmented by an expert. The following results include the average error and the standard deviation of this simulations set.

	DoF (T005)	DoF (T007)	E (MPa)
Vagina	928	1789	0.66
Uterus	740	523	0.6
Bladder	806	772	0.24
Rectum	894	2205	0.54
Fascia Pelvis	170	352	0.03
Fascia Sacrum	288	326	0.03
Ligaments	199	233	0.78

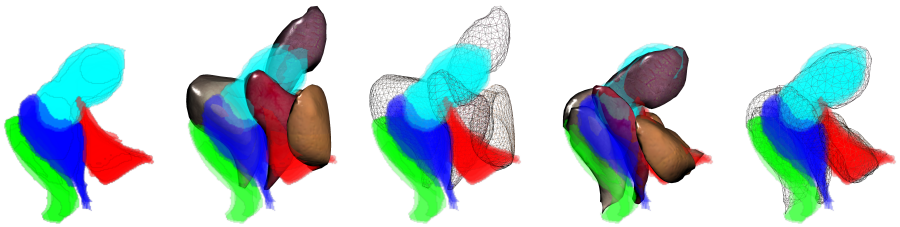
Table 1: Number of degrees of freedom (DoF) and Young's modulus (E) of the FE models used for the registration.

Since no off-plane data are available during the DYN sequence, validation can only be performed at the end of the dynamic acquisition using FFE slices. The 15 FFE Sagittal images have been segmented, providing a coarse representation

of the organs' geometries at maximal pressure state (see fig. 8a).



(a) T005 FFE (b) T005 TSE (c) T005 FFE



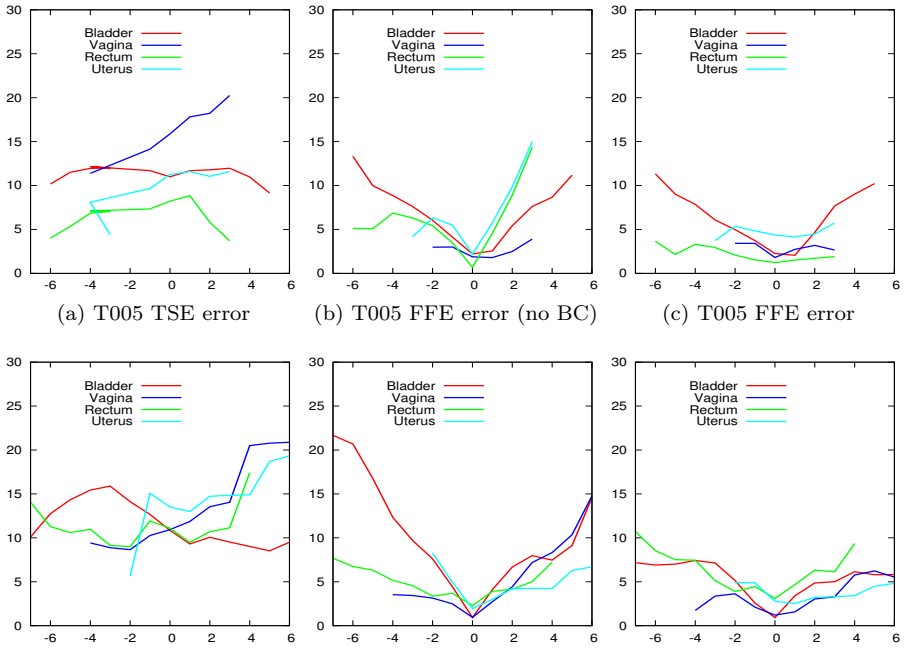
(d) T007 FFE (e) T007 TSE (f) T007 FFE

Figure 8: Overlapping between sparse FFE segmentation 8a and organs' geometries for both before 8b and after 8c the dynamic registration.

Overlapping with the organs' geometries is shown in figure 8. Results show that complex motion and deformations happened, enhancing the need for biomechanical models during the registration: For T005, the vagina has been significantly displaced and contracted, leading to the compression of the rectum and sliding of the uterus along the bladder. For T007, the bladder is compressed by surrounding tissues and the fasciae, which significantly modify the organ's geometry. Our method provides realistic transitions, including far from the image plane.

The average distance error according to FFE segmentation is reported in figure 9. Results show that our biomechanical registration significantly reduces the error in the 2D DYN plane (index 0) where constraints are applied. Still, it can also provide an accurate extrapolation for off-plane motion. The vagina undergoes the most significant displacement. The average error between TSE and FFE surfaces is $15mn$ initially and reduced to $3mm$ using our method. The remaining part of the error is mainly due to a significant amount of contrast gel expelled during the dynamic acquisition (due to compression of muscles), which is not considered in current FE models.

Fig shows a comparison with registration without boundary conditions to enhance the importance of anatomical constraints. 9. Although the method provides similar accuracy results in the dynamical plane, the error increases significantly faster for off-plane motion due to nonrealistic rotations of organs around the constrained plane. More importantly, without any boundary conditions, the method leads to the interpenetration of organs that are not physically possible.



(a) T005 TSE error (b) T005 FFE error (no BC) (c) T005 FFE error

(d) T007 TSE error (e) T007 FFE error (no BC) (f) T007 FFE error

Figure 9: Average distance error (mm) according to the index of FFE Sagittal planes. Index 0 corresponds to the location of DYN slices (i.e., where constraints are applied).

Finally, the total number of tetrahedral elements is shown in table 1. Each simulation step involves 30 nonlinear friction contact constraints with 70 outline constraints. Apart from the segmentation and meshing of TSE images, the method is compatible with real-time (i.e. 1 second is simulated in 1 second).

5 Conclusion

A new constraint-based formalism was introduced for non-rigid registration between a preoperative model and the intraoperative configuration. The method relies on the registration of 3D FE models to 2D dynamic MRI slices, providing a biomechanical extrapolation for the off-plane dynamic motion of organs. We showed that *a priori* knowledge of boundary conditions can be combined with intraoperative image-based data solve the ill-posed 2D/3D registration problem and provide a 3D extrapolation of the dynamic motion observed in a single 2D MRI slice.

The methodology follows a protocol compatible with current clinical constraints presenting potential short-term medical applications in this way. The registration accuracy was evaluated with two patient-specific data sets of volunteers presenting no pelvic pathology, and a sensitivity study is performed using synthetic data. We showed that the method is fast, robust, and compatible with intraoperative constraints. The resulting simulations provide a 3D extrapola-

tion of the dynamic of the organs observed in a single 2D MRI slice, facilitating diagnosis compared to 2D sequences. As future work, a model of loss of organs' volume related to the expulsion of contrast gel should be added to the biomechanical models.

References

[Baraff and Witkin, 1998] D. Baraff and A. Witkin. *Large steps in cloth simulation.* In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1998*, pages 43–54, ACM, 1998. 5, 6, 24

[Courtecuisse et al., 2014] H. Courtecuisse, I. Peterlík, R. Trivisonne, C. Duriez and S. Cotin. *Constraint-Based Simulation for Non-Rigid Real-Time Registration.* Medicine Meets Virtual Reality, vol. 196, pages 76–82, 2014. 18, 24

[Courtecuisse et al., 2020] H. Courtecuisse, Z. Jiang, O. Mayeur, J. F. Witz, P. Lecomte-Grosbras, M. Cosson, M. Brieu and S. Cotin. *Three-dimensional physics-based registration of pelvic system using 2D dynamic magnetic resonance imaging slices.* Strain, vol. 56, no. 3, page e12,339, 2020. 18, 24

[Duriez et al., 2006] C. Duriez, F. Dubois, A. A. A. Kheddar and C. Andriot. *Realistic Haptic Rendering of Interactive Deformable Objects in Virtual Environments.* IEEE Transactions on Visualization and Computer Graphics, vol. 12, no. 1, pages 36–47, 2006. 12, 24

[Felippa, 2000] C. Felippa. *A systematic approach to the element-independent corotational dynamics of finite elements.* Tech. Rep. January, Center for Aerospace Structures, 2000. 3, 24

[Jiang et al., 2015] Z. Jiang, J. F. Witz, P. Lecomte-Grosbras, J. Dequidt, C. Duriez, M. Cosson, S. Cotin and M. Brieu. *B-spline based multi-organ detection in magnetic resonance imaging.* Strain, vol. 51, no. 3, pages 235–247, 2015. 20, 24

[Nesme et al., 2005] M. Nesme, M. Marchal, E. Promayon, M. Chabanas, Y. Payan and F. Faure. *Physically realistic interactive simulation for biological soft tissues.* Recent Research Developments in Biomechanics, vol. 661, no. 2, pages 11–22, 2005. 4, 24

[Rusinkiewicz and Levoy, 2001] S. Rusinkiewicz and M. Levoy. *Efficient variants of the ICP algorithm.* In *Proceedings of International Conference on 3-D Digital Imaging and Modeling, 3DIM*, pages 145–152, IEEE, 2001. 15, 24

[Teschner et al., 2005] M. Teschner, S. Kimmerle, B. Heidelberger, G. Zachmann, L. Raghupathi, A. Fuhrmann, M. P. Cani, F. Faure, N. Magnenat-Thalmann, W. Strasser and P. Volino. *Collision detection for deformable objects.* Computer Graphics Forum, vol. 24, no. 1, pages 61–81, 2005. 6, 24

[Trivisonne et al., 2016] R. Trivisonne, I. Peterlik, S. Cotin, H. Courtecuisse, I. Peterlík, H. Courtecuisse, S. Cotin, H. Courtecuisse, I. Peterlik, S. Cotin and

H. Courtecuisse. *3D Physics-Based Registration of 2D Dynamic MRI Data*. In *Medicine Meets Virtual Reality*, vol. 220, page 432, IOS Press, 2016. 18, 24