Computer-based training system for cataract surgery

Jérémie Dequidt, Hadrien Courtecuisse, Olivier Comas, Jérémie Allard, Christian Duriez, Stéphane Cotin, Elodie Dumortier, Olivier Wavreille, Jean-Francois Rouland

Computer-Based Training System For Cataract Surgery

Jérémie Dequidt*[1,2,3], Hadrien Courtecuisse[1], Olivier Comas[1], Jérémie Allard[1,2,3], Christian Duriez[1,2,3], Stéphane Cotin[1,2,3], Élodie Dumortier[4,5], Olivier Wavreille[4,5] and Jean-François Rouland[4,5]

[1]Shacra(INRIA Lille - Nord Europe / INRIA Nancy - Grand Est / LIFL)
[2]LIFL, CNRS UMR8022
[3]University of Lille 1

[4]CHRU Lille, Hopital Claude Huriez
[5]Ophtalmologie, INSERM U86

August 30, 2013

Received: Dec. 16 2012, Revised: Apr. 12 2013, Accepted: Apr. 22 2013

Keywords: Cataract surgery simulation, real-time biomechanical models, training environment

Abstract

This paper describes a single simulation framework to perform interactive cataract surgery simulations. Contributions includes advanced bio-mechanical models and intensive use of modern graphics hardware to provide fast computation times. Surgical devices are replicated and located in a real-time thanks to infra-red tracking. Combination of a high-fidelity simulation and actual surgical tools are able to improve surgeon immersion while training. Preliminary tests have been performed by experienced ophthalmologists to qualitatively assess the face-validity of the simulator and the faithfulness of the behavior of the anatomical structures as well as the interactions with the surgical tools.

*E-mail address: jeremie.dequidt@inria.fr Contact address: Batiment INRIA A, 40 avenue Halley, 59650 Villeneuve d'Ascq, France. Phone number: +33 359.577.881. Fax number: +33 359.577.850

1 Introduction

Cataract surgery has made important advances over the past twenty years, and every year, more than five million people in the United States and in Europe undergo cataract surgery. This procedure is however complex to perform as it requires good hand-eye coordination as well as accurate microscopic manipulations of surgical instruments. Therefore a long training period is mandatory to acquire these necessary skills. Current method of training are based on companionship and this apprenticeship model depends on the availability of the instructor and potentially exposes patients to complications due to the inexperience of the operator. In this context, recent reports have demonstrated the advantages and added value of computer-based simulation over conventional training [1]. Therefore, new training simulation systems for cataract surgery have been recently developed. Typically, these simulators offer a combi-

nation of simulation software that aims at reproducing some parts of the cataract surgery procedure and hardware devices that allow the surgeons for realistic interaction with the simulation. For instance, [2] propose a Mass-Spring mechanical model of the capsule in order to simulate the capsulorhexis, whereas [3] and [4] focus on the phacoemulsification step using dedicated strategy to remove tetrahedrons from a volumetric mesh. Some other works have combined multiple cataract surgery steps in their simulators like [5] or [6]. Regarding interaction devices, some set-ups are based on 6DOF tracking systems that can possibly provide force feedbacks [6] or CCD-based optical tracking system like in EYESI simulator [1].

Simulators can provide almost limitless opportunity to practice and achieve surgical proficiency in a controlled and safe environment. However, if the training scenario is scripted in the simulator, like in a video game, then the benefit is limited. Hence, we think that the student must have the possibility to err and when it happens, it shall not be the end of the scenario. In such case, the novice must also learn to recover from the mistakes. To allow for such *freedom* in the simulation, the design of the software must rely on a physics-based simulation. If the behavior of the anatomical structures is guided by accurate deformable models, if the interactions between surgical tools and organs rely on accurate contact and friction laws, then it is possible to allow for a realistic simulation whatever the action of the user is. Obviously, the biomechanical model accuracy can not take precedence over the constraint of real-time execution: the motions of the user must quasi-immediately be transferred in the simulation. But the computational power available on today's CPUs and GPUs allows for attractive compromises between accuracy and computation time.

This paper presents our recent works in the field of cataract surgery simulator. Our main objectives are the following:

(a) be able to train on the three most important steps of the procedure: capsulorhexis, phacoemulsification, and intra-ocular lens injection which has not, to our knowledge, been treated previously

(b) develop advanced models based on biomechanics that can be easily parametrized using material properties and that can be computed quickly

(c) have a working environment and devices as close as possible to real operating rooms.

The rest of this article is divided as follows: section 2 provides a brief overview of our system, and each important step of the simulation is detailed in the following sections (*e.g.* the phacoemulsification in section 3, the capsulorhexis in section 4 and the intra-ocular lens implant in section 5). The section 6 is about the approach we use to make interactions natural to cataract surgeons.

2 Overview & Contributions

Training system: The training system described in this chapter is based on a computer-based simulator and a environment close to an actual operating room. The environment consists in a microscope where stereo glasses have been fixed in order to render the virtual scene in 3D, a full-body mannequin where replaceable silicon eyes that support cataract surgery may be inserted. Monitors are also available in the room to provide additional informations (like vital constants) or feedbacks. This serves a pedagogic purpose because it increases the feeling of immersion while allowing to train and memorize the real procedure with the same devices (microscope, pedals...) and the same surgical tools design using rapid prototyping. The tools are tracked in real-time and their 3D position is used to feed our real-time simulation. Figure 1 illustrates the various components that are part of the training system and how their are connected.

Real-time simulation: The computer-based simulation has been developed using SOFA[2] which is an open-source framework targeted at interactive medical simulation. SOFA provides a set of tools such as detection algorithms, GPU optimizations, linear

[1] http://www.vrmagic.com/simulators/eyesi-surgery-simulator/eyesi-cataract/

[2] http://www.sofa-framework.org

FEM... that offers a good trade-off between accuracy and computational cost. For improved realism, our simulation of cataract surgery relies on geometrically non-linear finite element models (FEM) using tetrahedral elements for the lens, and triangular elements for the capsule. Both models derive from Hooke's law and are based on a co-rotational method [7]. The different models have been created by a 3D artist based on anatomy handbooks. Mechanical properties are parametrized using values reported in the literature [8], and use an implicit integration scheme to enforce robustness of the deformation process. While this allows realistic and interactive deformation of the capsule and lens, additional information needs to be introduced to enable realistic tearing: the principal direction of fibers within the soft tissue. As many biological soft tissues are composite fibrous materials, the orientation of these fibers highly influences the direction of propagation of the tear in the tissue. It also introduces anisotropy in the model. State-of-the art computer graphics techniques (like Volume Rendering) are implemented in order to achieve convincing visible rendering of the human eye.

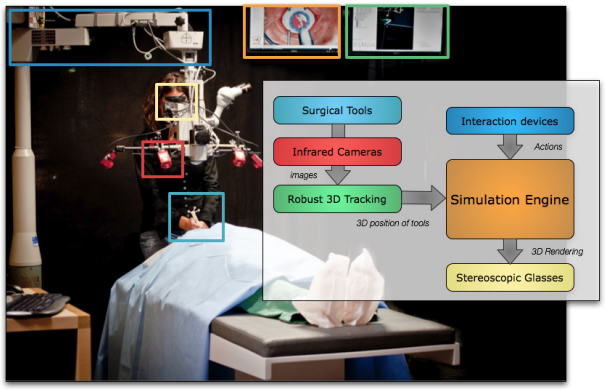


Figure 1: *Functional view and picture of the training system. The system relies on devices used in real procedures (microscope, pedals...), replicas of surgical tools that are tracked in real-time and on a simulation engine that renders a realistic and interactive simulation of cataract surgery.*

3 Lens model and phacoemulsification

This section presents the volume model used for modeling the deformations and the motions of the lens and also the interaction model developped for the phacoemulsificator.

The behavior of the lens is based on finite element modeling that allows for simulating different rigidities given the age and the pathology of the patient: typical stiffness values reported in the literature range from 1kPA (20-year old human) to 5 kPA (over 60 year old person) [9].The fragmentation and aspiration of the lens is performed in 3D and in real-time by using the phacoemulsificator's pedal which activates a local change in the geometry and behavior of the lens volumetric model. New algorithms and models have been developed to obtain this fast simulation computation without sacrificing realism.

3.1 Related Work

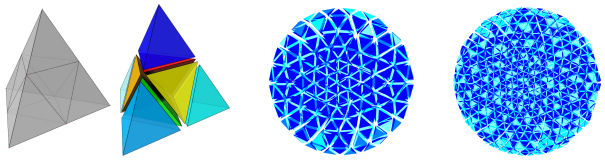
Many works focused on modeling the deformation of soft tissues. If we focus on real-time algorithms, mass-spring networks [10] is one of the most popular methods, mainly because of their simplicity and efficiency. Their main drawback appear in the difficulty to model specific material laws and parameters. However a recent work by [11] has shown that it is possible to compute springs stiffness on triangular and tetrahedral meshes relating to St Venant-Kirchhoff materials.

To model biomechanical behaviors more precisely, one can instead use Finite Element Models (FEM). This approach is more computationally intensive, and was initially used interactively by relying on preprocessing [12, 13], limiting the approach to linear small deformations without topological changes. To remove these limitations, [7] introduced the co-rotational formulation, which relies on linear elastic material but non-linear geometric deformations by computing a local rigid rotation for each element.

3.2 Tools and Methods

3.2.1 Deformable Lens Model

The deformable lens is modelled using a co-rotational FEM formulation with linear elasticity. It is represented by a tetrahedral mesh that can be tessellated to produce small elements (see Figure 2(b)). We chose an implicit time integration scheme in order to achieve large time-steps and provide a stable simulation even when subjected to challenging interactions. We use the approach detailed in [14], where the resulting system of equations is solved iteratively using a Conjugate Gradient solver implemented on the GPU.



(a) (b) (c)

Figure 2: *Tessellation algorithm that can dynamically refine tetrahedra (a). Two-levels tetrahedral topology for lens phacoemulsification (b).*

One important aspect of this method is that the mechanical matrix of the full system is not explicitly computed, but matrix-vector products are instead parallelized directly over the mesh elements. This allows to easily change the topology of this mesh, as is necessary in this application to implement the interactions with the phacoemulsifier explained below.

3.2.2 Interactions with the Phacoemulsifier

The phacoemulsifier proposes two modes of interactions to the surgeon:

Aspiration: when activated, liquids from the eye are pumped into the instrument, creating a force pushing the lens toward the tip of the phacoemulsifier.

Fragmentation: A probe at the tip vibrate at about 40,000 cycles per second, creating ultrasonic waves that break up the lens into small pieces that are sucked into the instrument by the aspiration and removed from the eye.

The aspiration aspect is handled by defining a spherical volume around the type and finding parts of the lens within it. When the aspiration is activated, an artificial force is added to pull the enclosed particles toward the tip. When fragmentation is also activated, we remove tetrahedral elements from the geometricalmodel of the lens.The removed elements should be as fine as possible to provide an accurate visual feedback to the surgeon. However, the mechanical mesh need to be relatively coarse to maintain good performances. To solve this issue, we use a finer tessellated mesh for rendering and interactions with the instrument. This is done by splitting each tetrahedron in the simulation mesh into 8 smaller tetrahedra (Fig. 2(a)). Therefore finer elements are removed due to aspiration. And if the 8 finer elements of a coarse tetrahedron are removed, the coarse tetrahedron is also removed in order to mechanically take into account the loss of matter in the lens. This allows the use of a relatively coarse mechanical mesh (Fig. 2(b)) while using a more detailed visual mesh (Fig. 2(c)) for the rendering.

3.2.3 Semi-Opaque Volume Rendering

The cristalline lens is opacified by the cataract, but it is not completely opaque. During phacoemulsification, the quantity of light that is bouncing back from the retina through the lens is an important visual cue for the surgeon to guide his action. To reproduce this effect, we rely on a volume rendering technique called *Projected Tetrahedra* as implemented by [15]. Each tetrahedron in the visual mesh is sent to the graphics card as input to complex shaders that render their contributions taking into account the light dissipation along the depth of the tetrahedra at each pixel.

3.3 Results

Figure 3 shows the resulting simulation of the phacoemulsification step. The lens is deformed when

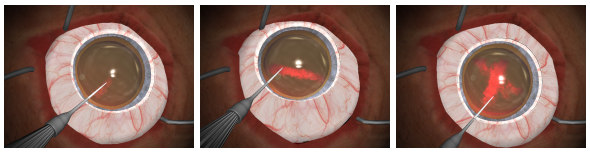


Figure 3: *Three screenshots of the phacoemulsification simulation. Local fragmentation of the lens create a cross-like pattern on the lens.*

in contact with the instrument and can thus be rotated by the surgeon to create a cross-like carving using local fragmentation and aspiration through the tip of the phacoemulsifier. This step is implemented by removing elements from a visual mesh that is tessellated at a finer resolution than the underlying mechanical mesh. This method is rather simple and do not suffer from the stability issues of other approaches based on local remeshing. The removal of these small elements progressively allows more and more light to pass through the lens. This effect is implemented through an advanced volume rendering technique. It is an important visual cue that the surgeon must learn to use in order to ensure not pushing the instrument too low within the eye, which can lead to puncturing the underlying membrane. The 3000 tetrahedra of the lens model are simulated and rendered on 2.4 GHz processor with a standard graphic card. Simulation performance is of 80 fps with an integration time-step of 0.01 seconds.

4 Lens capsule model and capsulorhexis

In this section, we focus on *Continuous Curvilinear Capsulorhexis*, a technique used to create a circular opening in the lens capsule. Once a small initial incision has been made, this technique relies essentially on the application of shear and stretch forces to create the opening. When performed correctly, the continuous curvilinear capsulorhexis creates a smooth opening, reducing the risk of tear when forces are applied to the capsule during surgery. The main risk with the continuous curvilinear capsulorhexis is to either create an opening that is too large or to create an edge that could promote a tear proceeding outwards. From a simulation and training standpoint, it is essential that the model of deformation and tear of the capsule is realistic. In [16] and [4] the lens capsule is modeled by a mass-spring network based on a triangular mesh. The main issue with these models is that they rely on discrete methods which require specific mesh structures (e.g. radial and concentric springs with different stiffnesses) to reproduce the anisotropic properties of the capsule. Also, when simulating the capsulorhexis, complex re-meshing techniques are required to maintain a similar mesh structure. The approach presented in this section relies on a continuous model based on elasticity theory for which specific fiber directions can be defined. This allows real-time anisotropic soft tissue deformation as well as realistic tearing simulation.

4.1 Tools and Methods

Using a finite element technique, one can no longer rely on the mesh topology to specify preferred directions of anisotropy. The approach presented here relies on a continuous model based on elasticity theory for which specific fiber directions can be defined (see 4.2). The underlying finite element model can handle geometrically non-linear anisotropic deformations at interactive rates. This model is combined with a novel technique for determining the fracture direction, which can be propagated along existing edges of the topology or across existing faces by using a remeshing algorithm [17].

In our approach, we propose to model the deformation and tearing of thin soft tissue (such as membranes, capsules, etc.) by using a transversely isotropic elastic model. The model is based on a corotational finite element method formulation, using triangular elements on which a specific fiber directions can be defined (see Figure 4). During deformation of the tissue, this fiber orientation, and the principal strain direction on the element are used to determine the tear propagation. Tearing or fracture within an element only occurs when a threshold is reached, which is computed using an eigenvalue decomposition of the strain tensor in each element. If the largest eigenvalue is above a given threshold

(in our experiments, the value is tuned according to the feedbacks provided by the surgeon), the element is identified as "breakable" and the eigenvector associated with this eigenvalue corresponds to the principal strain direction (see figure 4). Since tearing tends to propagate from an already fractured location, we also account for the history of tear location and direction in the overall computation of the current tear direction. Topological changes on the FEM mesh are performed at each time step. Each triangle in the neighborhood of the current tear location which strain is above our threshold will be re-meshed. This local re-meshing subdivides a triangle into new triangles. There is no particular requirement on the local re-meshing besides avoiding very small triangles which lead to ill-conditioned systems.

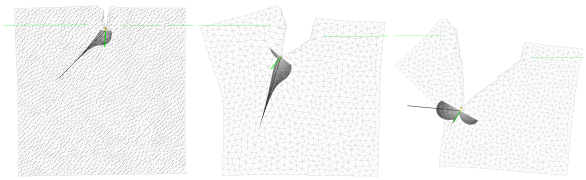


Figure 4: *From left to right: tearing propagation throughout the tissue. Fibers (not visible on the meshes since they are defined per triangles) are oriented with a 45 degree angle, and forces are applied outward, on the left and right sides of the square mesh. Our fracture criteria is represented as the shaded area. One can see on the left image how the anisotropy affects the deformation of the tissue, and on the rightmost images how the fracture direction depends on the previous fracture direction, as well as fiber direction. The square mesh is composed of 1,500 triangles. Anisotropic deformation, fracture and remeshing are performed in real-time.*

4.2 Anisotropic Soft Tissue Model

As many biological soft tissues are composite fibrous materials, additional information needs to be introduced to enable realistic tearing: the principal direction of fibers within the soft tissue. Indeed, the orientation of these fibers highly influences the direction of propagation of the tear in the tissue. It also introduces anisotropy in the model. The proposed model for describing deformation and tearing of thin soft tissue (such as membranes, capsules, etc.) relies on a transversely isotropic FEM formulation using triangular elements, in which a specific fiber direction can be defined on each element. The fiber direction on each element, called θ, is defined with respect to a local (co-rotational) reference frame (x, y). This local frame of reference is in turn defined with respect to a global frame of reference (X,Y) via a rigid transformation. The frame defined by the fiber is named (F,T), where F is along the fiber, and T is the transverse (orthogonal) direction (see Figure 5). This leads to the following definition of the anisotropic material stiffness matrix of an element:

$$\begin{bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{bmatrix} = \frac{1}{1-\nu_F\nu_T} \begin{bmatrix} K_{11} & K_{12} & K_{13} \\ K_{21} & K_{22} & K_{23} \\ K_{31} & K_{32} & K_{33} \end{bmatrix} \begin{bmatrix} \epsilon_x \\ \epsilon_y \\ \gamma_{xy} \end{bmatrix}$$

$$\begin{aligned}
K_{11} &= c^4E_F + s^4E_T + 2c^2s^2(\nu_T E_F + 2G_F) \\
K_{22} &= s^4E_F + c^4E_T + 2c^2s^2(\nu_T E_F + 2G_F) \\
K_{33} &= c^2s^2(E_F + E_T - 2\nu_T E_F) + (c^2 - s^2)G_F) \\
K_{12} &= c^2s^2(E_F + E_T - 4G_F) + (c^4 + s^4)\nu_T E_F) \\
K_{13} &= -cs[c^2E_F - s^2E_T - (c^2 - s^2)(\nu_T E_F + 2G_F)] \\
K_{23} &= -cs[s^2E_F - c^2E_T + (c^2 - s^2)(\nu_T E_F + 2G_F)]
\end{aligned}$$

where $c = cos\theta$, $s = sin\theta$, $G_F = E_F/(1+\nu_F)$ and $\nu_T/E_T = \nu_F/E_F$. Since we are dealing with the simulation of fracture and tearing, topological changes will occur. To numerically handle these changes in an optimal way, we use an iterative solver (conjugate gradient) for which there is no need to perform any matrix assembly [18].

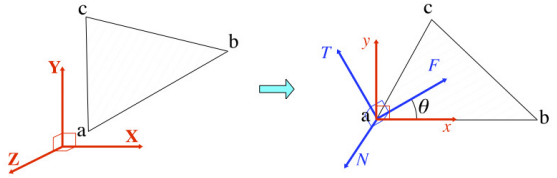


Figure 5: *Global reference frame (X,Y,Z), local frame (x,y), and frame (F,T,N) based on fiber orientation.*

4.3 Tearing Criterion

Although a significant amount of work exists in the field of finite element modeling for the simulation of fractures (see [19, 20] for instance), little has been done in the area of interactive medical simulation. Principal difficulties come from the real-time simulation requirement, and the need to account for the presence of fibers (previous work essentially deal with isotropic materials). In general, tearing or fracture occurs when a certain stress threshold is reached. For isotropic materials, this threshold is the same in every direction, and a fracture criterion can be determined using an eigenvalue decomposition of the stress tensor in each element [21]. If the largest eigenvalue is above the given threshold, the element is then fractured along a direction perpendicular to the eigenvector associated to the principal stress direction. This is however not applicable in the case of anisotropic materials, as the presence of fibers leads to preferred fracture directions. Also, the stress threshold is generally not identical along the fiber and transverse directions. Fracture occurs when the stress along the fiber or transverse to the fiber is higher to arbitrary threshold ($\bar{\sigma}_F$ and $\bar{\sigma}_T$) *e.g.* when $\sigma_F > \bar{\sigma}_F$ or $\sigma_T > \bar{\sigma}_T$ and we propose the following measure of the fracture condition, i.e. tearing occurs if:

$$c(\mathbf{d}, \boldsymbol{\sigma}, \mathbf{f}, \mathbf{p}) > 1 \tag{1}$$

where $\mathbf{d}$ is the potential direction of propagation of the fracture, $\boldsymbol{\sigma} = (\sigma_x \sigma_y \tau_{xy})$ is the stress tensor, $\mathbf{f}$ is the fiber direction, and $\mathbf{p}$ is the previous fracture direction (useful to avoid backtracking suddenly when we are continuing from an existing fracture, as otherwise the stress and fiber directions are undirected). This criterion is typically evaluated at the tip of a pre-existing fracture or at the center of each potentially fracturing element. In this work we propose a first formulation of the fracture criteria, based on the stress threshold in each direction $\bar{\sigma}$. This criterion includes a parameter limiting the angle between the previous and next directions of propagation to be at most θ_P:

$$\begin{aligned} c(\mathbf{D}, \boldsymbol{\sigma}, \mathbf{f}, \mathbf{p}) = \quad & \frac{\sigma_{D^\perp}}{\bar{\sigma}_{\mathbf{d}^\perp}} \quad H\left((\mathbf{d} \cdot \mathbf{p}) - \cos\theta_P\right) \\ \text{with } \mathbf{d}^\perp = \mathbf{d} \times \mathbf{n} & \end{aligned} \tag{2}$$

where H is the Heaviside step function: $H(a-b) = \{0 \text{ if } a < b, \quad 1 \text{ if } a \geq b\}$. The stress along a direction $\mathbf{u}$ can be computed by the coordinate transformation formula, using the angle θ_u between $\mathbf{u}$ and the x axis of the local frame of reference:

$$\sigma_u = \cos^2\theta_u \, \sigma_x + \sin^2\theta_u \, \sigma_y + 2\sin\theta_u \cos\theta_u \, \tau_{xy} \tag{3}$$

As illustrated in Eq. 1, anisotropic materials are typically characterized by two stress thresholds, $\bar{\sigma}_F$ in the direction of the fibers, and $\bar{\sigma}_T$ in the transverse direction. It is not clear what this threshold is for other directions. In our approach, we interpolate between $\bar{\sigma}_F$ and $\bar{\sigma}_T$ based on the angle between $\mathbf{u}$ and the fiber direction. To favor directions that are close to the fiber direction, it is useful to specify a peek function with a controllable steepness. This is achieved through a user-defined power α in the interpolation factor:

$$\bar{\sigma}_u = \bar{\sigma}_T + (\bar{\sigma}_L - \bar{\sigma}_T)\left(1 - \frac{2}{\pi}\cos^{-1}\left(|\mathbf{u} \cdot \mathbf{f}|\right)\right)^{\alpha} \tag{4}$$

With this formulation, the criterion along the fiber and transverse directions match the original conditions given by the anisotropic model.

Locally refining the mesh and lowering the timestep would allow a precise evaluation of the stress at the time of fracture [21]. A single direction will reach the threshold, thus a single candidate will be selected for fracturing. However, if the simulation needs to meet real-time constraints, this solution might prove prohibitive. Therefore, we propose the following approximations. First, in the case of linear materials, the computed stress inside each element does not precisely represent the real material stress, unless very fine meshes are used. To approximate the stress σ at the tip of the fracture, we propose to compute a weighted average of the stress inside each element in the neighborhood of the point of interest. Second, if large timesteps are used, several points and directions can reach the fracture threshold at the end of the timestep. To avoid substepping, which would be incompatible with real-time simulations, we propose the following heuristic: we choose as the fracture direction the direction $\mathbf{d}_{\mathbf{fracture}}$ where the criterion $c(\mathbf{d}, \boldsymbol{\sigma}, \mathbf{f}, \mathbf{p})$ is maximum, i.e. $c(\mathbf{d}_{\mathbf{fracture}}, \boldsymbol{\sigma}, \mathbf{f}, \mathbf{p}) = \max c(\mathbf{d}, \boldsymbol{\sigma}, \mathbf{f}, \mathbf{p})$. For the

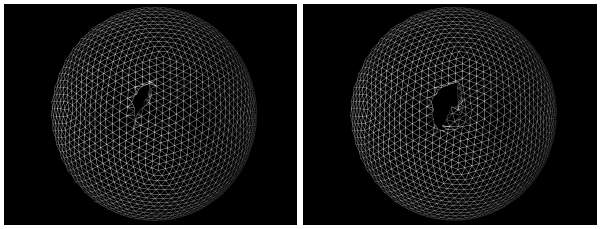
special case of an isotropic fracture criterion ($\bar{\sigma}_F = \bar{\sigma}_T$ and $\theta_\sigma = 90^o$), the solution can be computed by eigenvalue decomposition of the stress tensor. Thus, our algorithm is equivalent to the *maximum principal stress criterion* for the special case of isotropic materials. For general anisotropic surfaces, an exhaustive search on all the directions within the triangles connected to the fracture tip can be used, as the evaluation of our criterion can be implemented very efficiently.

4.4 Tearing Propagation

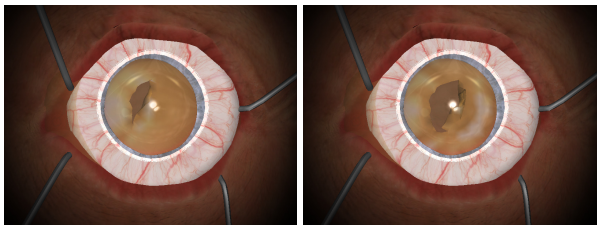
Once the fracture direction has been determined, topological changes need to be applied to the finite element mesh to describe fracture propagation. At this point, there are two possible strategies. In the case where the initial mesh is sufficiently fine, or if the edges are oriented along the fiber or transverse direction, it is possible to restrict the potential fracture direction to an existing edge. This solution is simple to implement, and the simulation cost as well as the quality of the initial mesh is preserved (as no new element is created). However the resulting fracture will be highly dependent on the original mesh. For a precise, mesh-independent, tearing propagation, it is necessary to allow for arbitrary fracture directions. Once the fracture direction $\mathbf{d}_{\mathbf{fracture}}$ has been determined, triangles need to be split to create new edges along the fracture direction. Particular care must be taken to avoid creating degenerated triangles which would otherwise negatively impact the computation of the finite element model. This issue is addressed using the work of [21].

4.5 Results

Figure 6 illustrates the tearing and deformation models in the context of a simulation of capsulorhexis. The membrane of the lens capsule is modeled as an anisotropic material with material properties similar to the actual lens capsule, and we use a co-rotational finite element formulation to numerically solve the problem. Anisotropy is produced by concentric fibers (radially spaced every 0.5mm) that are defined on the mesh to describe the actual structure of the lens capsule. An implicit integration scheme is used to enforce robustness of the deformation process. With this approach, realistic deformations can be computed in real-time, and tearing of the membrane can be simulated (see Fig. 6). The results clearly show qualitative realistic deformations accounting for the anisotropic nature of the tissue, and fracture propagation that depends on the fiber direction.



(a) Visualization of the opening into the lens capsule



(b) Visualization of the opening into the lens capsule

Figure 6: *Simulation of capsulorhexis, a technique used to create a circular opening in the lens capsule. This technique relies essentially on the application of shear and stretch forces to propagate a fracture throughout the membrane. Top (a): the mesh used to support the computation of the deformation and topological changes. Bottom (b): final result as seen in the training system.*

5 Intra-ocular Lens Implantation

At last, the injection of the artificial lens and its deployment in the capsule in real time is performed by simulating the intra ocular lens deformation and interaction within the capsule. The artificial lens is modeled using finite element shells and allows realistic reproduction of the mechanical behavior during the exit of the injector [22, 23]. The simulation also accounts for a physics-based modeling of the contacts

of the implant with the injector and the capsule walls as well as self-contacts (contact of the implant on itself). In the following, the model used to describe the implant is explained and the obtained results are presented.

5.1 Related Work

There are two main difficulties related to the simulation of the intra-ocular lens implantation. The first one is to model the complex deformation (essentially due to bending) of the lens, and the second consists in modeling the contacts between the lens implant and the injector or the lens capsule. The deformation of thin structures whose volume is negligible compared to their surface area (like intra-ocular implants) is fairly challenging. In fact, very few models have been proposed in the field of medical simulation for simulating, in real-time, the deformation of thin structures. To allow for an interactive simulation, a computationally efficient formulation is needed. Our finite element procedure was largely developed based on physical understanding without the use of mathematical shell theories. Indeed, our shell FEM was obtained by simply superimposing a plane stress and bending energies. Although we use a simple shell formulation, our model allows large displacements nevertheless through the use of a co-rotational framework. Co-rotational approaches have been successfully applied to real-time simulation over the last few years [7]. They offer a good trade-off between computational efficiency and accuracy by allowing small deformations but large displacements

5.2 Tools and methods

The approach used to model the complex deformations of the implant during its insertion in the lens capsule is based on the use of triangular shell elements and a co-rotational formulation. The combination of both leads to an accurate, yet computationally efficient, shell finite element method featuring both membrane and bending energies. In addition, the polynomial shape functions employed to compute internal forces in our FEM formulation are also used to process contacts and interactions with the curved triangles. The benefits of our approach (fast computation and possible interactions with other objects) allow us to successfully simulate the complex step of the insertion and deployment of the intra-ocular implant.

5.2.1 Geometrical model of the implant

To simulate the insertion and deployment of an intra-ocular lens, we first created a triangulation of the lens surface. Particular care was given to the mesh, to ensure that areas where large stresses occur contain a higher density of elements (see Fig. 7). This was done

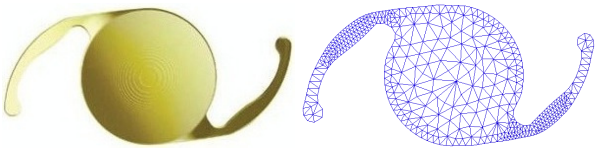


Figure 7: *An actual intra-ocular implant (courtesy of Alcon) and the triangular mesh used in our simulations. Notice the higher density of elements in areas where large deformations will take place.*

by noting the constraints applied by the surgeon to the haptics while inserting the implant within the injection device. During this stage, the haptics are folded onto the implant body, leading to high stresses at the junctions. The lens mesh contains 743 triangles and 473 nodes. Models of the injection device and the entire eye anatomy were also created. Physical parameters of the lens implant have been provided by the manufacturer Alcon. Thus, for the implant we specified a Young's modulus of $1MPa$, a Poisson's ratio of 0.42 and its mass density was set to $1.2g/cm^3$.

The first difficulty is to obtain the folded geometry of the lens within the injection device. This step is not important for the training process and does not need to be interactive. Indeed the surgeon does not always have to prepare the implant as some injection devices are readily available with a folded implant already in place. We simulate the folding process by first folding the haptics onto the implant body by applying forces on the haptics while maintaining the body fixed. While still exerting the forces on the haptics, the body was then rolled around them to

obtain the shape described in figure 8. The folded implant was then placed into the injection device.

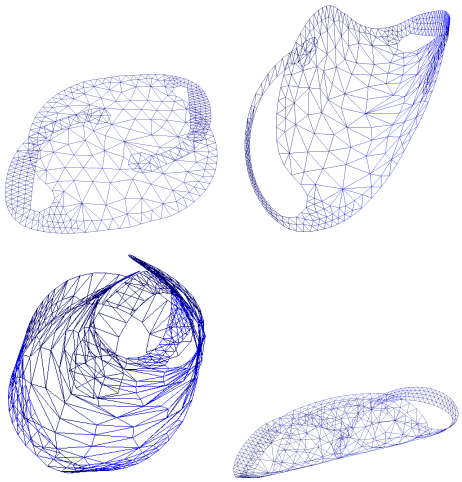


Figure 8: *Intermediary steps of the intra-ocular implant folding and fully folded implant ready to be placed into the injection device.*

5.2.2 Mechanical model of the implant

As mentioned earlier, we propose to define a triangular shell element by combining a two-dimensional in-plane membrane energy, with an off-plane energy for describing bending and twist. In the following we detail the bending stiffness computation in order to present the polynomial shape functions that are used in the shell model.

To calculate the stiffness matrix for the transverse

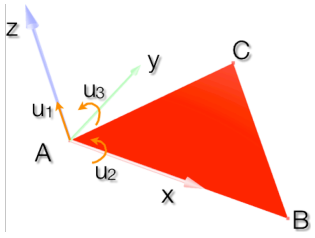


Figure 9: *The different degrees of freedom u of a triangular thin plate in bending.*

deflections and rotations shown on Fig. 9, the deflection u_z is computed using a polynomial interpolation:

$$\begin{aligned} u_z = c_1 + c_2 x + c_3 y + c_4 x^2 + c_5 xy \\ + c_6 y^2 + c_7 x^3 + c_8 xy^2 + c_9 y^3 \end{aligned} \tag{5}$$

where $c_1, \ldots, c_9$ are constants. Using a third-degree polynomial expression allows us to reach a greater precision for both the computation of the bending energy and the interpolation within the surface of the shell. Let us define the vector $\mathbf{u} = \{u_1 u_2 \ldots u_9\}$ of the displacements and slopes at the three corners of the triangular plate using the following notations:

$$\begin{aligned} u_1 &= (u_z)_{x_1,y_1} \\ u_2 &= \left(\frac{\partial u_z}{\partial y}\right)_{x_1,y_1} \\ u_3 &= -\left(\frac{\partial u_z}{\partial x}\right)_{x_1,y_1} \end{aligned} \tag{6}$$

and so on for the two other vertices and we can derive a matrix $\mathbf{C}$ such as $\mathbf{u} = \mathbf{Cc}$ where $\mathbf{c} = \{c_1 c_2 \ldots c_9\}$. We can then calculate the strains from the flat-plate theory using:

$$e_{xx} = -z\frac{\partial^2 u_z}{\partial x^2} e_{yy} = -z\frac{\partial^2 u_z}{\partial y^2} e_{xy} = -2z\frac{\partial^2 u_z}{\partial x \partial y} \tag{7}$$

Symbolically this may be expressed as $\mathbf{e} = \mathbf{Dc}$ where $\mathbf{D}$ derives from (5) and (7). Noting that $\mathbf{c} = \mathbf{C}^{-1}\mathbf{u}$, we have $\mathbf{e} = \mathbf{DC}^{-1}\mathbf{u} = \mathbf{bu}$ where the strain-displacement matrix $\mathbf{b} = \mathbf{DC}^{-1}$. The stiffness matrix $\mathbf{K}_e$ for an element is then obtained from:

$$\mathbf{K}_e = \int_v \mathbf{b}^T \chi \mathbf{b} dV \quad \text{where } \chi \text{ is the material matrix} \ . \tag{8}$$

The stiffness matrix in the global frame is eventually obtained using the rotation matrix of the element.

5.2.3 Mechanical interactions with the curved surface of shells

The practical interest of modelling complex behaviours such as bending and twisting would remain fairly low for medical simulation if contacts and constraints were not handled properly. Indeed, the simulation of the injection consists in pushing

the intra-ocular implant within the injection device into the lens capsule. During these stages of the simulation, complex contacts occur and consist of self collisions of the lens as well as collisions between the lens and the injector and later with the capsule. Obviously the contacts do not always occur onto the vertices of the triangles, they often occur onto the curved surface of the shell. In our case the difficulty of handling contacts comes from different sources. First the collision detection must be carried out with the curved surface of shell elements as opposed to the classic detection on plane triangles. Then the force applied onto any given point of the curved surface needs to be mapped onto the 3 vertices.

Firstly, in order to detect the collision with the bent surface, we have chosen the subdivision approach. We sample the flat surface of each element by recursively dividing each triangle into four smaller ones and the deflection of each new vertex is computed using (5) according to the displacements and slopes at the three vertices of the triangular element. This process of subdivision has two benefits. It allows us to render each shell as a curved triangle (Fig. 10 (a) and (b)) and to detect any collision with the curved surface of the shell using any of the classic collision detection algorithms working on flat triangles.

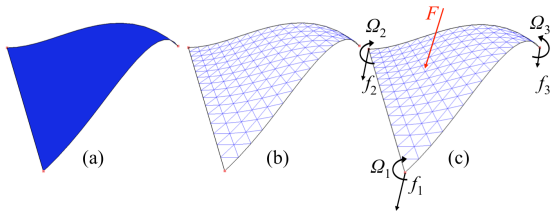


Figure 10: *(a) The triangle formed by the three vertices of the shell has been recursively subdivided 3 times and the deflection of each new vertex was computed according to the same deflection function used in our shell FEM. (b) Sampling the actual surface of the shell allows more accurate rendering and collision detection. (c) The shape function is used to distribute an external force F onto the triangle nodes.*

Secondly, once a collision has been detected, it must be processed by distributing the normal component of the force applied to the bent surface into a force and a torque at each of the three nodes (Fig. 10 (c)). This process is carried out by merely inverting our shell FEM formulation.

In a nutshell, the same polynomial interpolation function (5) chosen to compute the bending energy in our FEM formulation is also useful for handling the curved surface of our shells (regarding both visualisation and interactions with other objects). As adhesions between the haptics and the body is often observed in surgery, friction also needs to be taken into account in the contact response process. To solve the contacts we use the contact warping method proposed by Saupin *et al.* [24] as it offers an efficient way to compute physically correct contact responses in the case of co-rotational models.

5.2.4 Simulation results

Results of our simulation are illustrated in figure 11. We can notice the progressive deployment of the implant when it exits the injector. The shape of the intra-ocular lens remains very close to that of a real one during all stages of the simulation: within the injector, during the ejection phase, and when in place within the capsule. Due to the high stiffness and low mass of the lens, a direct sparse solver was used at each time step (dt $= 0.01$ s) rather than an iterative solver, resulting in a more accurate and more stable simulation, to the detriment of computation times (about 5 FPS for the complete simulation, and about 10 FPS for the deformation only, on a 2.4 GHz processor).

6 Interaction Devices

The way of interacting with the computer-based simulator is important especially if the system targets teaching and practising purposes. Moreover accuracy and stability of the interaction system is essential in a medical context. Our proposition combines replicas of real surgical tools as well as fast and robust 3D tracking to accurately track the manipulations of the surgeon.

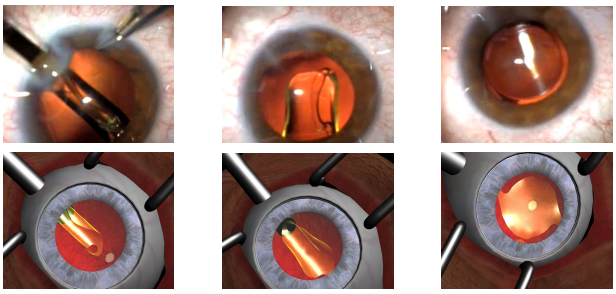


Figure 11: *Three steps of the simulation of the intra-ocular lens implant injection and its deployment within the lens capsule. Top: images from a real cataract surgery, courtesy of Dr. Tarek Youssef, ophtalmologist in Canada. Below: our simulation of the implant's deployment.*

6.1 Related Work

The use of a stylus attached to an articulated arm allows good precision and force feedback and is a popular interaction device for computer-based simulations [6, 3]. However it can hardly handle multiple tools and interaction modality (a stylus with a single button) is quite far from cataract surgery tools. Optical tracking offers a bigger workspace but may require time-consuming algorithms to provide 3D tracking of tools [16].

6.2 Methods

Our approach is based on infra-red optical tracking which combines the benefits of optical tracking with fast algorithms. Indeed the image segmentation is faster because only the reflective markers need to be detected. The rest of this section describes how the tracking is done and connected with the simulation engine.

Real surgical tools could have been used to interact with the simulation, however they suffer from two important drawbacks. First they are exepensive and second they are metal-based which makes the tracking more difficult and lower the accuracy (diffuse reflection area may appear on the images). Therefore, the use of replicas of surgical tools based on polyamide (nylon) was made. They were first modelled using standard CAD software and then were built using fast prototyping. Figure 12 shows the comparison between some of the real surgical tools and the ones used in our simulation. Reflective markers may disturb the clinician, especially for grasping tasks, but the markers are carefully placed to limit disturbance.

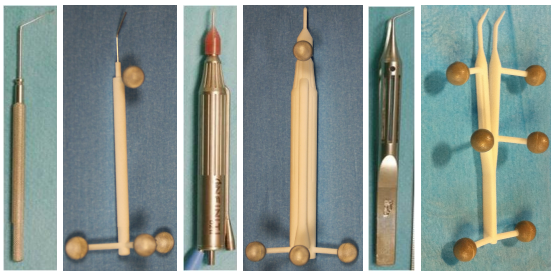


Figure 12: *Examples of tools used in cataract surgery and the ones we built for interacting with the simulation. From left to right: the phaco chopper, the phacoemulsification handpiece and the capsulorhexis forceps.*

6.3 Robust 3D Tracking

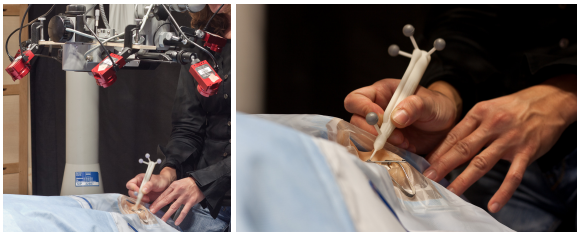


Figure 13: *Overview of our setup: mannequin and microscope where 6 infrared cameras are attached. These cameras allow to track in real-time the motion of the surgical tools.*

Our setup includes 6 infra-red cameras that targets the area around the eye. Two cameras may be sufficient to provide a 3D tracking but this is not a robust set-up. Indeed, intensive manipulations in that area may completely or partially hide instruments preventing the real-time tracking to work. With 6 cameras, the operative area is well covered and during any step of the procedure at least two cameras

are able to track the instruments. The setup is shown on figure 13. The redundancy involved with the 6 cameras make the calibration more time-consuming but once it has been performed, the tracking algorithm is able to provide 3D position of markers in real time. We are also using a feature proposed by *Tracking Tools*[3] software which allows to track rigid bodies. Each rigid body is defined as a set of reflective markers and the distance between markers do not change over time and the tracking is then more simple. 3 markers may be sufficient to define a 3D position but again for robustness issues, we introduce redundancy by placing 4 markers on the instrument. Using different *signatures* (*e.g.* the way markers are placed on the tools), we can manage to track several tools simultaneously (see figure 14) which is a prerequisite for several steps of the cataract surgery procedure.

For a better balance of the computation power needed to run our simulation system, two computers are used. The first one, which does not require large computational resources, is connected to the cameras and runs the tracking software while the second one runs the simulation engine. The two computers shared tracking informations by using the VRPN library [4] which is a lightweight and low-latency client-server protocol dedicated to virtual reality applications. The simulation engine is therefore on a single computer (with high-performance graphics cards) and can provide fast response to user inputs.

7 Conclusion

The results presented in the previous sections described original contributions which, combined in a single simulation framework, allow to perform interactive cataract surgery simulations. These contributions rely on advanced bio-mechanical models and intensive use of modern graphics hardware to provide fast computation times. Moreover, surgical devices, that are tracked in real-time thanks to infra-red cameras, have been reproduced in order to faithfully mimic the operating room environment and the actual procedures. Preliminary tests have been performed by experienced ophthalmologists to qualitatively look at the realism of the simulation and the faithfulness of the interactions with the surgical tools.

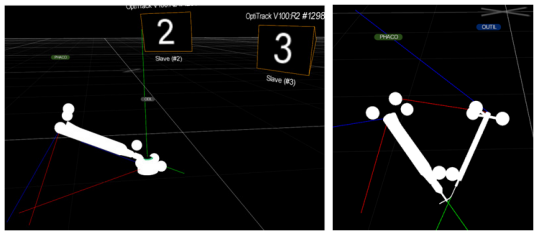


Figure 14: *Screenshots of the tracking of surgical tools. With the reflective markers, the tracking system is able to compute the 3D position of the tool in real-time and using different marker signatures, we can manage to detect several tools simultaneously without ambiguity.*

As these feedbacks were positive a double-blind experiment is conducted among ophthalmologist interns to assess the effectiveness of simulation training compared to current companionship training methods. This experiment is based on the following protocol: first, a common metric system has to be define in order to provide a common ground to evaluate surgeons performance in a real procedure or in a simulated procedure; second, the comparison of performance between interns that were exclusively trained on simulators will be confronted with interns that were trained according to companionship method.

The common ground used to measure surgeons performance during cataract surgery is based on [25] and is an objective performance rating tool derived from the Objective Structured Assessment of Technical Skill (OSATS) [26]. OSATS has also been validated for other surgical specialities. The work of Saleh *et. al.* on OSACSS (Objective Structured Assessment of Cataract Surgical Skill) exhibits construct validity and is therefore a valuable tool to assess the surgical skill of trainees [25]. OSACSS is based on a set of 20 measurements or observations that can be programmed into the simulation engine. For instance, indices in-

[3] http://www.naturalpoint.com/optitrack/products/tracking-tools/

[4] http://www.cs.unc.edu/Research/vrpn/

clude *Eye Positioned Centrally Within Microscope View*, *Iris Protection* or *Capsulorhexis: Formation and Circular Completion* and can be easily translated into computer instructions. All of these criteria are based on geometric criteria, contact analysis (pressure on some fragile anatomical structures) or procedure step completion. The experiment has started where a series of trainees practice in the virtual environment and are evaluated by both experts and the simulator. Scores will be compared and we do expect that the simulation provides scores that are close to those grant by the expert surgeons.

Once this step is achieved settling that our simulation provides comparable and objective scores that assess skills in cataract surgery, future works will require a second and longer experiment. This experiment will be conducted to determine the possible outcomes of the simulator and if practicing in our virtual environment is a viable training method that is complementary to companionship training.

References

[1] Linda T. Kohn, Janet M. Corrigan, and Molla S. Donaldson. *To Err Is Human: Building a Safer Health System*. National Academy press, 1999.

[2] K. Weber, C. Wagner, and R. Mnner. Simulation of the continuous curvilinear capsulorhexis procedure. In *International Symposium on Biomedical Simulations*, pages 113–121, 2006.

[3] Kup-Sze Choi, Sophia Soo, and Fu-Lai Chung. A virtual training simulator for learning cataract surgery with phacoemulsification. *Computers in Biology and Medicine*, 39(11):1020 – 1031, 2009.

[4] N Santerre, F Blondel, F Racoussot, G Laverdure, S Karpf, P Dubois, and J-F Rouland. A teaching medical simulator: phacoemulsification in virtual reality. *Journal francais d'ophtalmologie*, 30(6):621–626, 2007.

[5] Yousuf M. Khalifa, David Bogorad, Vincent Gibson, John Peifer, and Julian Nussbaum. Virtual reality in ophthalmology training. *Survey of ophthalmology*, 51.(3.):259–273, May 2006.

[6] Marco Agus, Enrico Gobbetti, Giovanni Pintore, Gianluigi Zanetti, and Antonio Zorcolo. Real-time cataract surgery simulation for training. In *Eurographics Italian Chapter Conference'06*, pages 183–187, 2006.

[7] C. A. Felippa. A systematic approach to the element independent corotational dynamics of finite elements. Technical Report CU-CAS-00-03, Center for Aerospace Structures, 2000.

[8] Ahmed Elsheikh, Defu Wang, and David Pye. Youngs modulus of elasticity for the human cornea. *Journal of cataract and refractive surgery*, 20(6):672, 2007.

[9] Theo G. van Kooten, Steven Koopmans, Thom Terwee, Sverker Norrby, J.M.M. Hooymans, and Henk J. Busscher. Development of an accommodating intra-ocular lensin vitro prevention of re-growth of pig and rabbit lens capsule epithelial cells. *Biomaterials*, 27(32):5554 – 5560, 2006.

[10] K. Montgomery, C. Bruyns, J. Brown, G. Thonier, A. Tellier, and J.-C. Latombe. Spring: A general framework for collaborative, real-time surgical simulation. In *Medicine Meets Virtual Reality (MMVR02)*, 2002.

[11] Hervé Delingette. Biquadratic and quadratic springs for modeling st venant kirchhoff materials. In *Proceedings of ISBMS 2008*, pages 40–48, July 2008.

[12] M. Bro-Nielsen and S. Cotin. Real-time volumetric deformable models for surgery simulation using finite elements and condensation. In *Computer Graphics Forum*, volume 15, pages 57–66, 1996.

[13] Stéphane Cotin, Hervé Delingette, and Nicholas Ayache. Real-time elastic deformations of soft tissues for surgery simulation. *IEEE Transactions on Visualization and Computer Graphics*, 5(1):62–73, 1999.

[14] Hadrien Courtecuisse, Hoeryong Jung, Jérémie Allard, Christian Duriez, Doo Yong Lee, and Stéphane Cotin. GPU-based real-time soft tissue

deformation with cutting and haptic feedback. *Progress in Biophysics and Molecular Biology*, 103(2-3):159–168, December 2010. Special Issue on Soft Tissue Modelling.

[15] Sébastien Barbier and Georges-Pierre Bonneau. GPU improvements on the sorting and projection of tetrahedral meshes for direct volume rendering. In *Pacific Visualization Symposium 2008, PacificVis 2008*, Kyoto, Japon, March 2008. IEEE VGTC. Poster session.

[16] Roger Webster, Joseph Sassani, Rod Shenk, Matt Harris, Jesse Gerber, Aaron Benson, John Blumenstock, Chad Billman, and Randy Haluck. Simulating the continuous curvilinear capsulorhexis procedure during cataract surgery on the eyesi system. *Stud Health Technol Inform*, 111:592–5, Jan 2005.

[17] Jérémie Allard, Maud Marchal, and Stéphane Cotin. Fiber-based fracture model for simulating soft tissue tearing. In *Medicine Meets Virtual Reality 17 (MMVR)*, pages 13–18. IOS Press, jan 2009.

[18] J. Allard, S. Cotin, F. Faure, P.J. Bensoussan, F. Poyer, C. Duriez, H. Delingette, and L. Grisoni. Sofa–an open source framework for medical simulation. In *MMVR conference*, 2007.

[19] Jeffrey Smith, Andrew Witkin, and David Baraff. Fast and controllable simulation of the shattering of brittle objects. In *In Graphics Interface*, pages 27–34. Blackwell Publishing, 2000.

[20] James F. O'Brien, Adam W. Bargteil, and Jessica K. Hodgins. Graphical modeling and animation of ductile fracture. *ACM Trans. Graph.*, 21:291–294, July 2002.

[21] Pavol Federl. *Modeling fracture formation on growing surfaces*. PhD thesis, Calgary, Alta., Canada, Canada, 2003. AAINQ87032.

[22] Olivier. Comas, Stéphane Cotin, and Christian Duriez. A shell model for real-time simulation of intra-ocular implant deployment. In *International Symposium on Computational Models for Biomedical Simulation*, volume 5958 of *Lecture Notes in Computer Science*, pages 160–170. Springer, 2010.

[23] Nadia Boubchir, Stéphane Cotin, Christian Duriez, Jérémie Dequidt, Jérémie Allard, and Jean-Francois Rouland. Computer-based simulation of iol injection: Toward a full featured cataract surgery training system. In *Meeting of the American Academy of Ophthalmology (AAO 2009)*. Unknown, 2009.

[24] Guillaume Saupin, Christian Duriez, and Stephane Cotin. Contact model for haptic medical simulations. In *Biomedical Simulation*, pages 157–165. Springer, 2008.

[25] Georges M. Saleh, V. Gauba, Arijit Mitra, Andre S. Litwin, Andrew K. K. Chung, and Larry Benjamin. Objective structured assessment of cataract surgical skill. *Archives of Ophthalmology*, 125:363–366, 2007.

[26] J. A. Martin, G. Regehr, R. Reznick, H. MacRae, J. Murnaghan, Hutchison C., and Brown M. Objective structured assessment of cataract surgical skill. *The British journal of surgery*, 84:273–278, 1997.

8 Authors' biographies

- **Jérémie Dequidt** completed a Ph.D. thesis in Computer Science at the University of Lille 1 in 2005. As a post-doctoral fellow he joined the SimGroup at CIMIT (Boston, MA) and then INRIA Alcove team, working on interventional radiology simulations. Since september 2008, he is an Assistant Professor in Computer Science at Polytech'Lille. He also is a member of the Inria research-team Shacra. His research works include collision detection and response, mechanical / geometric / adaptive modeling and experimental validation of the simulations.

- **Hadrien Courtecuisse** is a research engineer at IHU Strasbourg. He received his Ph.D. in computer science from Lille University in 2011. He

works in the field of real-time medical simulation since 2008. His research interests include finite element methods, GPU parallel processing using CUDA, Sparse Linear Algebra, Collision detection, and modelisation of the interactions with soft bodies.

- **Olivier Comas** is a research associate at National Research Council Canada. He received his Ph.D. in computer science from Lille University in 2010 and his M.S. in medical image processing and robotics from Strasbourg University in 2006. He works in the field of real-time medical simulation since 2006. His research interests include include finite element methods and GPU parallel processing using CUDA.

- **Jérémie Allard** received a PhD degree in Computer Science from University of Grenoble in 2005. Then, he got a postdoctoral position at the CIMIT SimGroup in Boston. He joined INRIA in 2007 where he gets a research scientist position. His research interests include new highly parallel architectures such as GPUs in the context of interactive simulations for medical applications.

- **Christian Duriez** received a PhD degree in robotics from University of Evry in 2004 for a research work done at CEA Robotics lab on haptic rendering of collision between deformable solids. Then, he got a postdoctoral position at the CIMIT SimGroup in Boston. He arrived at INRIA in 2006 where he gets a research scientist position. His research work concerns finite element methods computed in real-time, robot control by simulation, contact modeling and haptic rendering.

- **Stéphane Cotin** joined INRIA in 2007 to hold a position of Research Director. Since January 2010 he also leads the SHACRA group, a multidisciplinary team of scientists involved in the field on medical simulation. He also manages the development of a Large Scale Initiative on Medical Simulation, a collaborative research program involving scientists from six different teams. Before joining INRIA, he was Research Lead for the Medical Simulation Group at CIMIT, where he managed a team of 12 permanent researchers and post-doctoral fellows. His main research interests are in physics-based simulation, real-time simulation of soft-tissue deformations, and medical applications of this research.

- **Élodie Dumortier** is a Senior Registrar of the Universities and a Medical Assistant at the Hospitals of Lille. Her research focuses on the validation of simulation to train cataract surgery procedures.

- **Olivier Wavreille** is a Senior Registrar of the Universities and a Medical Assistant at the Hospitals of Lille. His research focuses on the palpebral portion of the lacrimal gland and on the tear film.

- **Jean-François Rouland** is a University Professor at University of Lille and is also the Head of Ophthalmology Department in Lille Hospital. He is also the General Secretary of the French Society of Glaucoma Expert of the French Ministry of Health.