MODELLING OF SOFT TISSUES FOR REAL TIME SIMULATIONS IN VIRTUAL SURGERY

U. Gabbert, H. Berger, J. Grochla, Y. Kaymak and C. Willberg

Institute of Mechanics, University of Magdeburg, Germany
e-mail: ulrich.gabbert@ovgu.de

Keywords: Soft Tissue Modelling, Finite Element Analysis, Virtual Surgery, Virtual Laparoscopy.

Abstract. In the paper a finite element approach and a software concept for virtual surgery of soft tissues is presented. The finite element approach includes the geometrical and physical nonlinear dynamic behavior of the organs during surgery as well as the required contact conditions between the organs and the organs with surgical instruments. Some simplifications of the standard procedure (contact search, updating rate etc) to increase the computational speed are discussed. Finally, test examples are presented to demonstrate the performance of the developed software.

1 INTRODUCTION

Operations with endoscopic techniques enable more patient conservative interventions. In endoscopic surgery the surgeon has to operate in a three-dimensional domain, only seeing a distorted two-dimensional picture on the screen. Investigations have shown that even an experienced surgeon has to be in a continuous training to operate safely with low error rate. Today an animal based training and education is mostly used. But, this is expensive and limited by tight ethical borders, and consequently, an increasing activity is observed to create suitable alternatives to using animals for teaching and developing surgical skills. Exciting potentials provide Virtual Reality (VR) simulators, which allow to navigate around and to interact in a 3D computer-simulated environment. In virtual surgery the surgeon acts in a simulated medical environment created from a virtual patient in a risk-free manner [1]. It can also be used for several different medical applications, such as pre-operative planning, robot-assisted surgery, teaching technologies, visualization of medical data, etc. There are several devices available on the market. The simulator Kylie is a virtual reality surgical simulator invented by a team at Monash University; allowing surgeons to practice by working at a computer screen in real time [2]. DellaTech has developed the Simendo Virtual Reality (VR) trainer to train the basic skills of Minimally Invasive Surgery [3]. The Xitact LS500 is a virtual trainings device for laparoscopic surgery [4]. One of the leading companies in virtual medicine is Simbionix, which developed the advanced device LapMentor [5], which allows real time laparoscopic surgery with quite realistic models of the organs in the abdomen. Another advanced device is Kistar developed at the Research Center Karlsruhe [6].

One of the main issues is the real time performance of the simulator, which still guarantees a realistic physical behavior of the organs during interaction with surgical instruments. From a mechanical point of view this interaction results in a transient movement of deformable bodies with geometrical and material nonlinear behavior (large deformations, non-linear material laws). The calculation of the deformation of different bodies consisting of soft tissues including their interaction with different constituents (surgeon/organ, organ/organ, organ by itself) needs additionally fast contact search and reaction algorithm.

One of the main disadvantages of existing surgery simulators is their unavailable or insufficient haptic feedback. Beside technical problems, inadequate physical models of the soft tissues are another issue. Some simulators have severe restrictions on the modeling of the scenarios, i.e. unavailable input options which can hinder a realistic simulation using the individual data of a patient. Some simulators produce
unrealistic deformation patterns if an organ to organ contact or a contact with surgical instrument is involved in the simulation.

These problems have stimulated the authors to contribute to the development of such software tools, which are characterized by a real time performance of more complex operation scenarios with a physical realistic behavior of soft tissues during surgery including also a realistic force feedback at the haptic devices. The real-time calculation of soft non-linear bodies performing large deformations is the main challenge of a surgery simulator. To reduce the computational effort and to guarantee the real-time calculation, it is inevitable to use extremely simplified simulation models together with efficient numerical algorithms. The calculation of large models with a nonlinear behavior in real-time is still impossible with the current micro-computer's technology. Nevertheless, in most cases extremely simplified models can still provide realistic images during an interactive real-time simulation. Since numerical calculated images do not allow proving the quality of the solution, experimental investigations are necessary. But, unfortunately, there are only few publications available reporting experimental results.

There are two main approaches to perform physics-based simulations of the virtual surgery. One group of approaches uses simplified discrete mechanical models, such as the ChainMail approach [7], the spring mass approach [8] and the sphere-filled approach [9]. These methods apply simple but fast algorithms which can be performed in real-time without any problem. However, the adjustment of real material properties is impossible, which is one main source of partially very strange deformation patterns, and, consequently, also the force feedback behavior is far from the range of the accuracy requirements for the haptic rendering device. The second group of approaches applies continuum models, mainly based on the finite elements method (FEM), a common and powerful tool in structural and continuum mechanics today [10]. In such structural mechanics simulations the accuracy requirements of the solution are normally very high and several methods have been created to automatically guaranty the desired accuracy by an error controlled meshing and model adaptation. The computational effort of such calculations, expressed by the storage requirements and the computation time is in general very high and far away from a real time performance. A real time performance requires a strong reduction of the accuracy demands, but is limited by not losing the physical obviousness of the results. Several methods are suggested in the literature to increase the computation speed of finite element simulations to achieve a real time performance [11], [12], [6], [13],[14].

In the present paper a finite element approach as well as an adequate software concept for virtual surgery is presented. The finite element approach is based on simple tetrahedron meshes (figure 1) including geometrical and physical nonlinear dynamics behavior of the organs during surgery as well as the required contact conditions between the organs and the organs with surgical instruments.

Some simplifications of the standard procedure (contact search, updating rate etc.) to increase the computational speed are discussed. Finally, test examples are presented to demonstrate the performance of the developed software.

Figure 1: Tetrahedron meshing of a liver.
2 FINITE ELEMENT MODELLING OF SOFT TISSUES

2.1 Finite element meshing

From a structural mechanics point of view, organs are anisotropic heterogeneous materials with complex geometries. Therefore, a deformation simulation of an ensemble of organ requires a geometrical and physical nonlinear approach. Other challenging issues are simulation of the contact between the organs as well the surgical instruments, the cutting of organs, the clamping and suturing of tissues, and a lot of further technologies.

Figure 1 shows how an organ model is built up. The surface and volume of the organ is approximated by triangular and tetrahedron finite elements, respectively. Starting point of the mesh generation is a surface net of triangles resulting, e.g., from computer tomography as a set of STL-data. Normally this surface mesh consists of too much triangles and cannot be used as basis of a tetrahedron meshing, which has to be coarse enough for real time analysis. But, a fine triangular surface mesh is excellently suited as basis of a photorealistic presentation of the surgery scenario consisting of an assembly of organs and the surgical instruments. Consequently, in our analysis software we are working with coarse finite element objects of the organs, and via the so called exchange object (see Figure 7) the results are extrapolated to the fine surface mesh used for representing the scenario at the screen. One of our meshing tools starts with a regular subdivision of the 3D objects in cubes, subdivided in regular tetrahedron elements, which are then mapped in such a way, that the outer nodal points are part of the surface.

2.2 Nonlinear finite element analysis

The computation of the complex nonlinear deformation and stress-strain responses is performed by using the finite element method. Under the action of forces, such as surface tractions $\overrightarrow{q}$ and body forces $\overrightarrow{p}$, a flexible body deforms such that the stresses $\sigma$, the strains $\varepsilon$ and the displacements $\overrightarrow{u}$ are fulfilling the principal of virtual work written as

$$
\int_{V} [\sigma : \varepsilon] dV + \int_{S} [\rho \overrightarrow{u} \cdot \overrightarrow{u}] dS = \int_{V} [\overrightarrow{q} \cdot \varepsilon] dV + \int_{V} [\overrightarrow{p} \cdot \overrightarrow{u}] dV,
$$

where $\varepsilon$ and $\overrightarrow{u}$ are the virtual strains and the virtual displacements, respectively. Introducing element wise displacement approximate functions (element shape functions) written as

$$
\overrightarrow{u}(\overrightarrow{x}) = N(\overrightarrow{x}) \overrightarrow{U},
$$

with $\overrightarrow{U}$ as the element displacement vector the strain displacement relation can be written as

$$
\varepsilon(\overrightarrow{x}) = B(\overrightarrow{x}) \overrightarrow{U},
$$

with the strain-displacement matric $B$. With the constitutive equation in form of

$$
\sigma = \overrightarrow{\varepsilon} E + \overrightarrow{\eta} \dot{\varepsilon},
$$

including a viscous material behavior the equation (1) can be written as

$$
\int_{V} \left[ \overrightarrow{U}^T B(\overrightarrow{E}U + \overrightarrow{\eta} \dot{\overrightarrow{U}}) \right] dV + \int_{S} \left[ \overrightarrow{U}^T N^T \rho N \overrightarrow{U} \right] dS = \int_{V} \left[ \overrightarrow{U}^T N \overrightarrow{q} \right] dV + \int_{V} \left[ \overrightarrow{U}^T N \overrightarrow{p} \right] dV.
$$

Assembling the element contributions and cancel out the arbitrarily virtual nodal displacements the equation of motion can be written as

$$
M \ddot{\overrightarrow{U}} + C \dot{\overrightarrow{U}} + K \overrightarrow{U} = \overrightarrow{F},
$$

with the mass matrix

$$
M = \sum_{(e)} \int_{V} [N^T \rho N] dV,
$$

where $e$ indicates the element.
the damping matrix

$$C = \sum \int [B^T \cdot \dot{B}] dV,$$

the stiffness matrix

$$K = \sum \int [B^T \cdot \dot{E} \cdot B] dV,$$

and the external load vector

$$F = \sum \left( \int [N^T \cdot \dot{q}] dS + \int [N^T \cdot \dot{p}] dV \right).$$

![Figure 2: Principle of stiffness warping.](image)

Based on the above given general approach several types of finite elements have been developed. In virtual surgery a four node tetrahedron element (see figure 1) as the simplest 3D element is applied, which realizes an approximation of the displacements inside the element with linear polynomials containing \((1, x, y, z)\). The element matrices have the format of \((12 \times 12)\) with three displacements at each nodal point as degrees of freedom. The terms of the element matrices and load vectors of equations (6) to (10) can be quite simply calculated without numerical integration, such resulting in a fast algorithm for generating the element stiffness and mass matrices, respectively. If the material behaves nonlinear and if the elements perform large displacements (rotations and stretching), which is always the case in virtual surgery, then the solution process becomes highly non-linear and requires an iterative approach, which is very time consuming. The investigation of soft tissues in surgery applications requires in most cases a nonlinear deformation strategy, which is mainly estimated by large rotation but still a small stretching of the material. We have tested different types of stretching rates, and it could be proved that the error, even in cases of relatively large stretches of about 50%, in the difference between a fully non-linear deformation analysis and a simplified linear analysis, which only takes into account the large rotation, but still a linear strain-deformation analysis is lower than 10%. From an engineering point of view, such a difference is too large in structural mechanics, but still acceptable in a real-time surgery analysis. Such approach, also known in literature as Warped Stiffness Method [15] is based on small deformations and large rotations, and can be applied within an acceptable tolerance. Large rotations necessitate a special attention to eliminate the stress free rigid-body rotations [16], which is performed within the stiffness warping. The basic idea is shown in Figure 2. Once the element rotation matrix \(R_e\) is known, first the current configuration is rotated back to reference frame \(X_0\) and then the forces are computed using the initial stiffness and the forces are rotated back to current frame \(X\).

### 2.3 Time integration

There are several possible time integration schemes for solving the system ordinary differential equations (6), such as the explicit Euler's scheme (central difference method), the Newmark and Wilson method etc. In the following the unconditionally stable Euler's backwards approach is used for the time integration of equation (6) as
\[ \ddot{\mathbf{X}}^{(n+1)} = \ddot{\mathbf{X}}^{(n+1)} = \frac{1}{\Delta t}(\dot{\mathbf{X}}^{(n+1)} - \dot{\mathbf{X}}^{(n)}) \]
\[ \mathbf{U}^{(n+1)} = \dot{\mathbf{X}}^{(n+1)} \]
\[ \mathbf{U}^{(n+1)} = (\mathbf{X}^{(n)} + \dot{\mathbf{X}}^{(n+1)} \Delta t) - \mathbf{X}^{(0)} \]

(11)

with equation (11) the equation (6) can be written as

\[ \mathbf{A} \dot{\mathbf{X}}^{(n+1)} = \mathbf{b}^{(n+1)} \]

(12)

with the dynamic stiffness matrix \( \mathbf{A} \) and the generalized force vector \( \mathbf{b} \) as

\[ \mathbf{A} = \mathbf{M} + \mathbf{C} \Delta t + \mathbf{K}(\Delta t)^2 \]

(13)

\[ \mathbf{b}^{(n+1)} = \mathbf{M} \dot{\mathbf{v}}^{(n)} - \Delta t[\mathbf{K}(\mathbf{X}^{(n)} - \mathbf{X}_0) - \mathbf{F}^{(n+1)}] \]

(14)

The global stiffness matrix and the force vector are assembled from the element matrices and the element force vectors by using the connectivity data. The linear system of equations is solved by a preconditioned conjugate gradient method (pcg method). The rotation matrix is calculated by using the Newton iteration when the first absolute norm of the spin part of the deformation tensor exceeds certain limit. If the computed rotation matrix differs more than a given limit from the previous one, then the stiffness contribution of the element is recomputed and updated. Consequently, computing time is reduced, because the complete stiffness matrix is not updated at every time step. The algorithm has been proven to be very effective, since often only small parts of the objects under consideration perform large rotations during surgery (see marked part of a liver in figure 3).

![Figure 3: Liver before (I) and after deformation (II).](image)

![Figure 4: Different contact situation between organs.](image)

The biggest challenge in catching-up the real-time performance is to solve the equation system (12). Some authors suggest ignoring the rotational part of the deformations and recommend using a constant stiffness matrix, which is only inverted once at the beginning of the simulation [17]. However, inclusion of rotations demands a very fast solver. We use the pcg method equation solving, which has an extraordinary performance thanks to the diagonally dominant sparse structure of the stiffness matrix.

2.4 Contact problem

The virtual surgery software requires an efficient and robust contact algorithm, which includes the interaction of organs, the self contact and the contact with the boundaries (see figure 4) as well as the interaction with the surgical instruments. For the contact with instruments, the forces and moments have to be additionally computed to control the haptic device. To implement such force feedback the interaction of the instruments with the soft tissues has to be modeled properly. The contact zones cannot be initially specified, and, consequently, in each time step a search algorithm is required first to identify potential contact zones and then to calculate the contact forces in detail.

There are several ways for performing a fast global contact search in physics-based animations and particle flow simulations. The most frequently used strategies are categorized as bounding volume strategies in which the complex model is completely enclosed by a set of basic geometric shapes (usually boxes, spheres, cylinders, etc.) [18], [19]. Tree based algorithms, which work quite fast for rigid body systems (no change in the bounding volume) [20], but don’t perform well in case of large deformations [21], and self-contact [22]. The distance field concept provides advantages [23], but the efficiency of the
method is not fully investigated [24]. Unlike the bounding volume strategies, Oldenburg and Nillson suggest a fixed grid-structure for the global search [25], which does not require a tree structure. This is an advantage if cutting operations are required, because a restructuring of the search strategy is not necessary.

In the following a grid-structure methodology, which we have applied successfully, is briefly presented. The simulation space is considered as a grid-structure, where each grid cell represents a cube. The cubes are numbered in the x-, y-, and z-directions, respectively. In principle, the size of the cubes can be chosen arbitrarily. However, a too small size disregards some of the contact zones while a too large size slows down the algorithm due to increased number of local checks. Fung et al [26] suggest that the size of the cube can be chosen as the longest side length of the enclosing prisms of each element.

![Cubes and Search strategy](image)

Figure 5: Efficient contact search in a 2D case.

For getting the potential contact zones one has to run a loop two times over all nodes. Figure 5 demonstrates the procedure by looking at node P1. Around the coordinates of P1 a box is added. The corners of this box are in different cubes. The result of this example is that P1 is in potential contact with P2 and P3. P5 cannot be in contact with P1, because P1 and P5 are the nodes of the same element. The nodes may belong to the same object potentially causing self-contact or different objects potentially causing an object-object contact. At the end of the global search a list of nodes with potential contact partners exists, which is used for a local contact search algorithm. This principally performs an inside-outside test based on local coordinates. The coordinates of a nodal point $P$ are inside a finite element, if the equation

$$L_{1P} + L_{2P} + L_{3P} + L_{4P} = 1,$$  \hspace{1cm} (15)

is fulfilled, where $L_{ip}$ are the tetrahedron coordinates of the element under consideration [16]. If equation (15) is not fulfilled, then the node is outside the element and no contact occurs with this element. The contact forces are calculated according to Seydel [27] and included by a penalty approach.

## 3 SOFTWARE CONCEPT

In the following a brief summary of the main principles of the software concept of our Virtual Surgery Trainer (VST) is presented. The software consists of different encapsulated parts, which have access to a consistent data basis via software functions provided by a data management system. A overview about the VST software is given in figure 6. The main parts are (i) the initialization process, (ii) the finite element analysis process, and (iii) the control process, respectively. The consistent data keeping, organized in virtual data array, the data access, the data exchange operations, and several other functionalities and operations at the data are provided by a data manager, which supplies the programmer also with user routines for interacting with the data. The data are partitioned in objects, such as the finite element objects, the exchange objects and the control objects. The main task of the exchange objects is to provide the exchange between the internally used finite element objects (volume models) and the objects providing the information for the photorealistic graphical presentation (refined surface model) as well as the data for controlling the hardware devices (force feedback of the surgical instruments). The finite element object includes the environmental object (surgical scenery), the manipulation objects (the surgical in-
4 CONCLUSION

In the paper an overview about our developments of a virtual surgery trainer is presented with a focus on the development of fast algorithms to increase the calculation speed without losing the physical obviousness of the results. The quality of the simulation has been evaluated by comparing the results with the solutions provided by commercial software tools of structural mechanics (e.g. ABAQUS) calculated with refined finite element meshes. Figure 7 shows one of the tested scenarios. To evaluate the run time performance a milt meshed with 600 tetrahedron elements has been tested with two computers, an AMD with an Athlon 64 Processor 3200+ with 2.2 GHz and 1 Gigabyte RAM, and an Intel Core Duo T7250 2GHz with 2GB RAM. All simulations were performed using a single core only without any parallelization. An acceptable and stable solution is achieved with a time step length of $\Delta t=0.003$s. With our software tool 222 frames per second are measured at the first computer and 335 frames per second at the second one, respectively. With a parallelized software version a much higher performance can be achieved, which is required if more complex models and surgical operation have to be simulated with a proper accuracy.

ACKNOWLEDGEMENT

The work has been supported by the German State of Saxony-Anhalt, which is gratefully acknowledged.

REFERENCES


[27] Seydel M., Solving the contact problem within the FEM simulation of sheet forming (in German), Forschung im Ingenieurwesen, Düsseldorf, Bd. 54, Nr. 1, 1988.