

Surgery Simulation using Fast Finite Elements

Morten Bro-Nielsen^{1,2}

¹ Dept. of Mathematical Modelling
Technical University of Denmark, Bldg. 321
DK-2800 Lyngby, Denmark

² 3D-Lab, School of Dentistry, Univ. of Copenhagen,
Nørre Alle 20, DK-2200 Copenhagen N, Denmark
e-mail: bro@imm.dtu.dk WWW <http://www.imm.dtu.dk/~bro>

Abstract. This paper describes our recent work on real-time Surgery Simulation using Fast Finite Element models of linear elasticity [1]. In addition we discuss various improvements in terms of speed and realism.

1 Introduction

Real-time surgery simulation using general *volumetric* models has just recently become possible [2, 3, 5]. With the development of Fast Finite Element (FFE) models it is now possible to simulate the elastic behaviour of a solid with video frame rates, ie. 20 frames/second.

Other attempts have used *surface* models as the basic modeling method [6]. The problem with surface models, besides the obvious non-solid behaviour, is the lack of an defined interior when surgical procedures are modeled. The surgeon cannot cut a virtual organ modeled using a surface model since there is nothing inside the surface. Some simple cuts can be modeled, such as cutting an artery or other thin structures, but general surgical incisions are impossible.

This paper discusses real-time simulation of deformable objects using 3D solid volumetric Fast Finite Element which result in linear matrix systems. In addition we discuss parallelization of the FFE models using domain decomposition and empirical derivation of non-linear forces to account for material non-linearities.

2 Theory

In this section we describe the model which we use to simulate elastic deformation of a volumetric solid in real-time [3]. To develop the model we formulate a number of requirements that the model should fulfil:

1. Speed is everything. Deformation should be calculated in the smallest amount of time possible.
2. We do not care about the time taken for one-time pre-calculation such as setting up equations, inverting matrices etc. If something takes 24 hours extra in the pre-calculation stage, but will save 0.01 second in the simulation stage, we should do it.

3. The elastic model should be *visually* convincing. The model may be physically incorrect if it looks right.
4. In the long run we want to be able to make cuts in the model to accommodate surgical procedures. This involves changing the topology of the model and most importantly requires models that have defined interiors, ie. volumetric models.

In particular the last requirement lead us to select mesh-based 3D Finite Element (FE) models. The alternative would be parametric models such as [6]. But these models do not provide the needed freedom to perform topology changes to allow cutting. Although some of the models can handle simple cuts we aim towards being able to make completely general cuts in the models. We are convinced that only mesh-based models will allow this.

To meet the first two requirements we choose the linear elastic deformation model which is also known as Hooke's law. Using linear elasticity as the basic model involves a number of assumptions regarding the physical material that is modeled. Most importantly linear elastic models are only valid for very small deformations and strains. They are typically correct for such rigid structures as metal beams, buildings etc. Although they are used extensively in modeling, the visual result of large deformation modeling using linear elasticity is seldom satisfactory

But when used with FE these models lead to linear matrix systems $\mathbf{K}\mathbf{u} = \mathbf{f}$ which are easy to solve and fast. There is, therefore, a trade-of between the speed of the system and the visual deformation result.

Linear elastic models are used here because modeling general elastic volumetric deformation using FE is only *just* possible with todays computers. With faster computers in the future we expect more realistic models, such as incompressible Mooney-Rivlin material models [4], to be used.

2.1 Condensation

The linear matrix system $\mathbf{K}\mathbf{u} = \mathbf{f}$ models the behaviour of the *solid* object. This includes both surface nodes as well as the internal nodes of the model. But for simulation purposes we are usually only interested in the behaviour of the surface nodes since these are the only *visible* nodes. We, therefore, use condensation [9] to remove the internal nodes from the matrix equation.

The matrix equation for the condensed problem has the same size as would result from a FE *surface* model. But, it is important to understand that it shows *exactly* the same behaviour for the surface nodes as the original *solid volumetric* system.

Without loss of generality, let us assume that the nodes of the FE model have been ordered with the surface nodes first, followed by the internal nodes. Using this ordering we can rewrite the linear system as a block matrix system (surface / internal):

$$\begin{bmatrix} \mathbf{K}_{ss} & \mathbf{K}_{si} \\ \mathbf{K}_{is} & \mathbf{K}_{ii} \end{bmatrix} \begin{bmatrix} \mathbf{u}_s \\ \mathbf{u}_i \end{bmatrix} = \begin{bmatrix} \mathbf{f}_s \\ \mathbf{f}_i \end{bmatrix} \quad (1)$$

From this block matrix system we can create a new linear matrix system $\mathbf{K}_{ss}^* \mathbf{u}_s = \mathbf{f}_s^*$ which only involves the variables of the *surface* nodes:

$$\mathbf{K}_{ss}^* = \mathbf{K}_{ss} - \mathbf{K}_{si} \mathbf{K}_{ii}^{-1} \mathbf{K}_{is} \quad \mathbf{f}_s^* = \mathbf{f}_s - \mathbf{K}_{si} \mathbf{K}_{ii}^{-1} \mathbf{f}_i \quad (2)$$

The displacement of the internal nodes can still be calculated using $\mathbf{u}_i = \mathbf{K}_{ii}^{-1} (\mathbf{f}_i - \mathbf{K}_{is} \mathbf{u}_s)$. Notice, that if no forces are applied to internal nodes, $\mathbf{f}_s^* = \mathbf{f}_s$.

Generally the new stiffness matrix will be dense compared to the sparse structure of the original system. But, since we intend to solve the system by inverting the stiffness matrix in the pre-calculation stage, this is not important. Without loss of generality, we will understand that both the original system and the condensed system $\mathbf{K}_{ss}^* \mathbf{u}_s = \mathbf{f}_s^*$ can be used when the following text refers to the original system $\mathbf{K} \mathbf{u} = \mathbf{f}$.

2.2 Solving $\mathbf{K} \mathbf{u} = \mathbf{f}$ using Selective Matrix Vector Multiplication

Formally, solving the linear matrix system using the inverted stiffness matrix is performed using $\mathbf{u} = \mathbf{K}^{-1} \mathbf{f}$. If only a few positions of the force vector are non-zero, clearly standard matrix vector multiplication would involve a large number of superfluous multiplications. We note that

$$\mathbf{u} = \mathbf{K}^{-1} \mathbf{f} = \sum_i \mathbf{K}_{*i}^{-1} \mathbf{f}_i \quad (3)$$

where \mathbf{K}_{*i}^{-1} is the i 'th column vector of \mathbf{K}^{-1} and \mathbf{f}_i the i 'th element of \mathbf{f} . Since the majority of the \mathbf{f}_i are zero, we restrict i to run through only the positions of \mathbf{f} for which $\mathbf{f}_i \neq 0$ [3]. If n of the N positions in \mathbf{f} are non-zero this will reduce the complexity to $o(n/N)$ times the time of a normal matrix vector multiplication. We call this approach *Selective Matrix Vector Multiplication* (SMVM).

2.3 Parallelization $\mathbf{K} \mathbf{u} = \mathbf{f}$ using domain decomposition

Let us assume that the domain of the solid Ω has been decomposed into a number of non-overlapping sub-domains Ω_i with a common boundary Γ .

If we order the nodes of the global stiffness matrix \mathbf{K} with the nodes of the boundary Γ first followed by sections of nodes corresponding to the sub-domains Ω_i , we can use the condensation technique described in the previous section to separate computation of the individual sub-domains. Each sub-domain can now be handled by one processor and the result assembled on a final processor. This way we are able to parallelize solution of the linear matrix system.

3 Simulation system

In this section we describe how we generate the FE mesh model of the physical organ, limb etc. and show the simulation system that we have implemented.

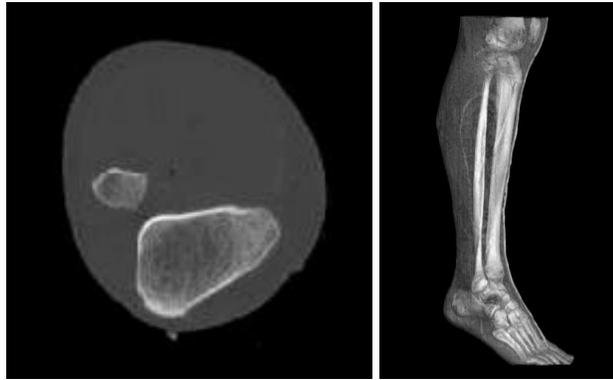


Fig. 1. Voxel data from the visible human data set.

In addition to a range of simple box-like structures we have used data from the Visible Human project [10] to make a model of the lower leg.

Since the Visible Human data set is voxel-based (see figure 1) it was necessary to generate a mesh model of it. To do this, we first used the Mvox software [1] to manually draw contours on the boundary of the skin and bone in the voxel data. We then applied the Nuages software [7] to create a 3D tetrahedral mesh model of the leg. The result was the FE mesh model shown in figure 3.



Fig. 2. Simulation system implemented using SGI Performer.

The simulation system has been implemented on an Silicon Graphics ONYX with four Mips R4400 processors using the SGI Performer graphics library. SGI Performer helps the programmer create parallel pipe-lining software by providing the basic tools for communication, shared memory etc.

Figure 2 shows a screen dump with the Virtual Operating room environment and the leg lying on the operating table. Figure 3 shows the surface of the FE mesh shown in the simulator.

4 Conclusion

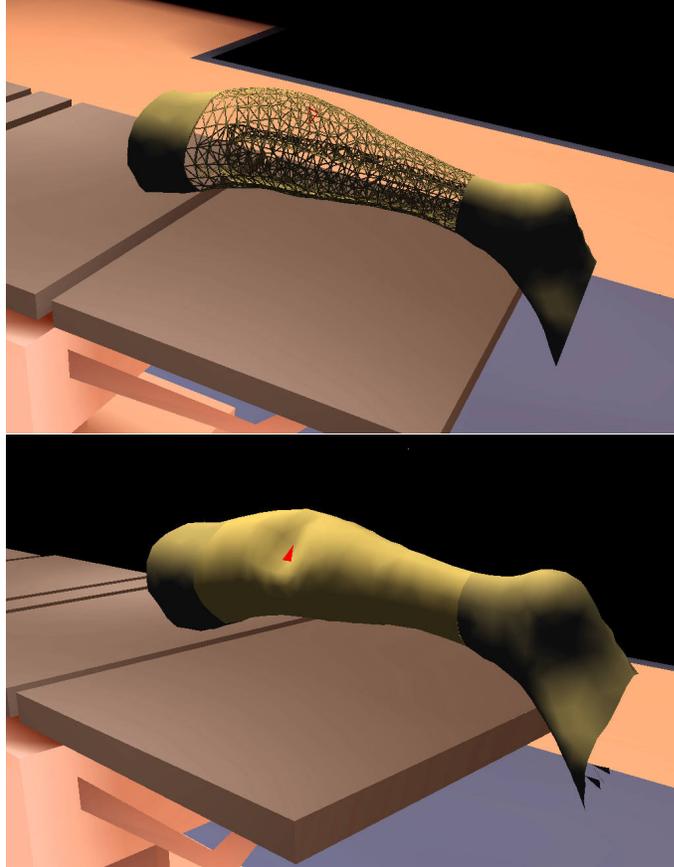


Fig. 3. Top: Wireframe model of lower leg in simulator. Bottom: Simulation of pushing on a the lower leg.

In this paper we have described a method for real-time simulation of elastic deformation of a volumetric solid based on linear elastic Fast Finite Elements

(FFE). We have discussed how solution of the linear matrix system $\mathbf{K}\mathbf{u} = \mathbf{f}$ can be implemented in parallel using domain decomposition. The simulation system we have developed for surgery simulation was finally described.

The example using a leg from the Visible Human data set with 700 system nodes (condensed system with only surface nodes) ran comfortably with a single processor using only 1/3 of a frame (20 frames/second) when forces were applied to 3 nodes. This included calculation of the deformation and also basic processing. So although both more nodes and more surface nodes with forces applied would increase the time requirement, we believe bigger models could be accommodated using the SMVM method.

Although we have shown that real-time simulation of solid volumetric deformable models is possible there is still much work to be done before realistic surgery simulation can be performed. Most importantly we are currently working on the implementation of cutting in a FE mesh.

In addition to more realistic tissue models we also need detailed segmentation of the organs, limbs etc. to allow different material properties and models to be used. The current parallel research in digital atlases such as the VoxelMan atlas [8] is a significant step in this direction.

References

1. M. Bro-Nielsen, *Mvox: Interactive 2-4D medical image and graphics visualization software*, submitted to CAR'96, 1996
2. M. Bro-Nielsen, *Modelling elasticity in solids using Active Cubes - Application to simulated operations*, Proc. Computer Vision, Virtual Reality and Robotics in Medicine (CVRMed'95), pp. 535-541, 1995
3. M. Bro-Nielsen and S. Cotin, *Real-time Volumetric Deformable Models for Surgery Simulation using Finite Elements and Condensation*, Proc. Eurographics'96, 1996
4. P.G. Ciarlet, *Mathematical elasticity, vol. I: Three-dimensional elasticity*, North-Holland, ISBN 0-444-70529-8, 1988
5. S. Cotin, H. Delingette, M. Bro-Nielsen, N. Ayache, J.M. Clément, V. Tassetti and J. Marescaux, *Geometric and Physical representations for a simulator of hepatic surgery*, Proc. Medicine Meets Virtual Reality, 1996
6. S.A. Cover, N.F. Ezquerra and J.F. O'Brien, R. Rowe, T. Gadacz and E. Palm, *Interactively deformable models for surgery simulation*, IEEE Computer Graphics & Applications, pp. 68-75, Nov. 1993
7. B. Geiger: *Three-dimensional modeling of human organs and its application to diagnosis and surgical planning*, INRIA Tech. Rep. 2105, Dec. 1993
8. K.H. Höhne, M. Bomans, M. Riemer, R. Schubert, U. Tiede and W. Lierse, *A 3D anatomical atlas based on a volume model*, IEEE Computer Graphics Applications, 12(4):72-78, 1992
9. H. Kardestuncer, *Finite element handbook*, McGraw-Hill, ISBN 0-07-033305-X, 1987
10. *The Visible Human Project*, WWW
http://www.nlm.nih.gov/extramural_research.dir/visible_human.html