High Performance Implementation for Simulation of Brain Deformation

Grzegorz Soza[1,2], Roberto Grosso[1], Christopher Nimsky[2], Günther Greiner[1] and Peter Hastreiter[1,2]

[1]Computer Graphics Group, University of Erlangen-Nuremberg
[2]Neurocenter, Department of Neurosurgery, University of Erlangen-Nuremberg
Email: soza@informatik.uni-erlangen.de

Abstract. Intraoperative movement of brain tissue is a central problem in the context of accuracy of image-guided neurosurgery. One of the strategies compensating for this issue is to perform a physically-based biomechanical simulation of the occurring phenomenon. Thereby, the computational expense is of big concern. In this paper, we present a parallel framework for modeling the intraoperative brain deformation. Within this system, it is possible to perform simulations on high resolution meshes even consisting of millions of elements. Experiments with the implemented algorithm using a cluster equipped with eight processors showed a great improvement in computation time in comparison to a standard single processor implementation.

1 Introduction

The accuracy of neuronavigation systems suffers from intraoperative brain deformation known as "brain shift". Commercially available image guided surgery systems are based on extrinsic (e.g. fiducial markers) or anatomical features (e.g. skin surface points). However, with this information, only rigid transformations between pre- and intraoperative data can be calculated and subsequently used for navigation. This does not account for the highly complex non-linear deformation of brain tissue during the course of surgery. Therefore, the assessment of brain shift has been subject of intensive research of various groups in recent years. In this context, a number of methods has been developed including registration approaches [1] and simulation strategies based on biomechanical models [2, 3]. However, algorithms simulating deformations of soft tissue under the influence of real physical factors are characterized by high computation times. This makes their application difficult for intraoperative use.

In [4], a fast simulation method was presented, where volumetric calculations were reduced with the use of a condensation technique into a model considering surface nodes only. Hansen *et al.* [5] used regions of interest to select levels of accuracy between different parts of the brain. In this hybrid approach, a dynamic finite element model was applied around the surgical target point, whereas in other parts of the brain static computations were performed. A parallel implementation accelerating brain shift simulation was presented in [6]. This approach, however, suffers from imbalance of matrix assembly among processors.

In order to compensate for and analyze the brain shift phenomenon, we have already introduced a biomechanical model describing and simulating brain deformation [7]. In this paper, an extension of the previous work is presented. Aiming at the use of the simulation model during neurosurgery, a parallel simulation approach has been developed in order to accelerate the calculations. The efficiency of this parallel implementation has been evaluated on a PC cluster exhibiting a great improvement in computation time over the single processor approach. Within this framework, it is also possible to follow and visualize the direction and the magnitude of the simulated deformation of brain tissue. Furthermore, insight into volumetric changes of brain structures is obtained, thus providing additional information contributing to the improvement of surgical outcomes. The simulation model has been already applied to surgical cases, based on pre- and intraoperative MR data, demonstrating its value [7, 8]. In this work we concentrate on the details of the parallel framework.

2 Parallel Framework

In the numerical deformation model, the brain is assumed to be a linearly elastic tissue saturated by a viscous fluid. This is expressed in a coupled set of equations describing the behavior of poroelastic materials [7]. This system of constitutive equations is discretized based on the Galerkin weighted residuals method. In order to perform a numerical simulation, volumetric geometry has to be provided. For this purpose, a surface consisting of regular triangles is generated from preoperative MR brain data in a first step. Based on this representation, a tetrahedral geometry is created.

In order to perform parallel assembling of the presented equations, the generated mesh has to be partitioned. In order to optimally use computational resources of all processors, it is important to ensure that local parts of the mesh have approximately the same number of elements and corresponding vertices. On the other side, connectivity between local parts shall be taken into account in order to minimize the communication between the processors, which is a time consuming event. In this context, the optimal partitioning is accomplished using the METIS package [9]. Thereby, a unique number corresponding with one of the available CPUs is assigned to each mesh element. According to this partitioning, the elements and their corresponding vertices are distributed among all processors with the use of message passing interface (MPI) commands. The optimized partitioning contributes to a better balance in the matrix assembly and linear system solving process. However, the vertices located at the boundary of neighboring partitioning regions are accounted for in a separate way since they have to be distributed to more than one processor. These special vertices are called "ghost vertices". Additionally, local-to-global mappings are appropriately defined in order to operate efficiently on both distributed elements and vertices.

In a further step, a part of the global matrix is assembled by each processor based on the locally stored mesh structure and the boundary conditions are set. Subsequently, the results from all processors are summed up and the global linear

Table 1. Cluster architecture (per node)

Component	Description
CPU	2 × 2.66 GHz Intel Xeon, 533 MHz, 512 KB Cache
Memory	3 GB ECC DDR-RAM
Network	onboard dual port 10/100/1000 BASE-T Gigabit Ethernet
Hard disk	hotplug ST336607LC 36.7 GB Ultra320 SCSI
Operating system	Linux Suse 8.2

system is created. Finally, the system of equations is solved with the Portable, Extensible Toolkit for Scientific Computation (PETSC) library [10].

Based on the resulting displacement vector field, the input intensity data is recalculated and both the deformed brain volume and the visualization of the simulated displacements are provided for analysis.

3 Results

A series of computational experiments with the described method were conducted in patients undergoing craniotomy. In this context, three pairs of pre- and intraoperative T1-weighted scans consisting of 512 x 512 x 160 voxels of size 0.44 mm x 0.44 mm x 0.89 mm were considered. In a preliminary step, volumetric tetrahedral meshes of varying resolution were generated from the preoperative images rigidly registered to the corresponding intraoperative data.

The performance of the presented parallel framework was analyzed on a IA32 cluster. The architecture consisted of 4 blade servers (see Table 1) networked with 1000 Mbps Gigabit Ethernet.

For each mesh, 5 simulation runs using identical conditions were performed on one, two, four and all eight processors, respectively. Altogether 20 simulations were performed for each case, giving the number of 60 runs in all experiments, which provided sufficient data for statistical evaluation of the computation times. The mean values of the measured times (calculated from 5 experiments each) for all three series of experiments are presented in Figure 1.

In addition to the performance analysis, visualization of the simulated brain deformation was conducted showing good correlation of the obtained results and the acquired intraoperative MR scans (Figure 2).

4 Discussion

In the conducted experiments, our implementation showed scaling behavior with respect to the number of nodes used for the computation. The times required for matrix assembly and solving resulting system of equations were drastically

Fig. 1. Resulting performance of the parallel implementation with logarithmic scaling of the x-axis. Mesh with *a)* 123496, *b)* 545235 and *c)* 1691150 tetrahedra. In case of the highest resolution, the runs could only be performed on four and on eight processors due to insufficient memory for storing the mesh and additional data structures

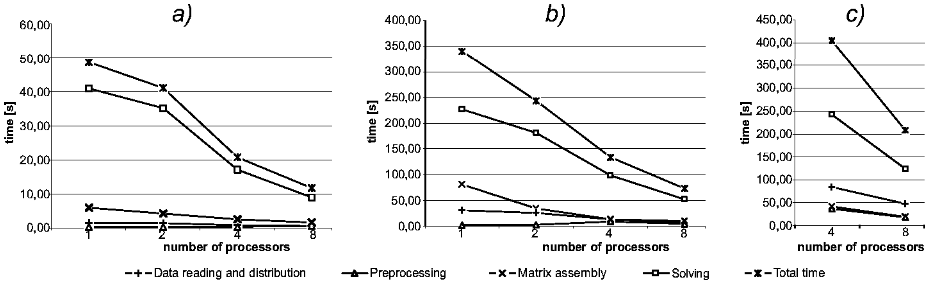
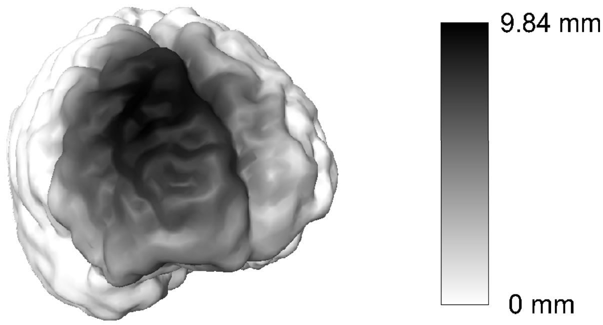


Fig. 2. Visualization of a simulated brain deformation. The color coding denotes the magnitude of displacement



reduced in comparison to a sequential version of the algorithm. Using a moderately (with respect to the underlying complex deformation) scaled tetrahedral grid, one time step for a sparse linear system consisting of 107820 equations was performed on eight CPUs within 11 seconds. Compared to this, the computation time took about 50 seconds using the single processor approach (see Figure 1a).

A successful experiment was performed for a huge mesh consisting of 1691150 tetrahedra, which resulted in 1357332 linear equations and a parallel computation time of about 3.5 minutes. In addition to accelerated computation times, the parallel implementation was the only approach which allowed conducting simulation based on a very high resolution mesh that would not fit into address space of a single processor machine.

The overall time required for an entire simulation must also comprise brain segmentation and mesh generation. This can be, however, conducted in a preprocessing step and is therefore not a limitation for a practical application during surgery. After simulation and computation of the respective deformation field, the corresponding image data has to be reconstructed accordingly. Up to now, this step is performed with a sequential approach requiring a few seconds only.

5 Conclusion

We presented a parallel framework for biomechanical simulation of brain deformation. The presented approach allows predicting the intraoperative shift of brain tissue. Employing a parallel architecture, low computation times were achieved, which makes clinical application possible. The approach was evaluated with preoperative MR data. The results of the simulation were compared to intraoperatively acquired data, showing the practical value of the method.

Acknowledgment: We are grateful to Stefan Zachow (Zuse Institue, Berlin, Germany) for generating high quality tetrahedral grids. This work was funded by Deutsche Forschungsgemeinschaft in the context of the project Gr 796/2-3.

References

1. Rueckert D, Sonoda LI, C Hayes DLGHill, Leach MO, Hawkes DJ. Nonrigid Registration Using Free-Form Deformations: Application to Breast MR Images. IEEE Trans Med Img 1999;18(8):712–721.
2. Miga M, Paulsen K, Hoopes P, Kennedy F, Hartov A, Roberts D. In Vivo Quantification of a Homogeneous Brain Deformation Model for Updating Preoperative Images during Surgery. IEEE Trans Biomed Eng 2000;47(2):266–273.
3. Ferrant M, Nabavi A, Macq B, Black PM, Jolesz FA, Kikinis R, et al. Serial Registration of Intraoperative MR Images of the Brain. Med Image Anal 2002;6(4):337–359.
4. Bro-Nielsen M. Surgery Simulation Using Fast Finite Elements. In: Proc. Visualization in Biomedical Computing. No. 1131 in Lect Notes Comp Sc. Springer; 1996. p. 529–534.
5. Hansen KV, Larsen OV. Using Region-of-Interest Based Finite Element Modeling for Brain-Surgery Simulation. In: Proc. MICCAI. Springer; 1998. p. 305–316.
6. Warfield S, Talos F, Tei A, Bharatha A, Nabavi A, Ferrant M, et al. Real-Time Registration of Volumetric Brain MRI by Biomechanical Simulation of Deformation during Image Guided Neurosurgery. Comput Visual Sci 2002;5:3–11.
7. Soza G, Grosso R, Hastreiter P, Labsik U, Nimsky Ch, Fahlbusch R, et al. Fast and Adaptive Finite Element Approach for Modeling Brain Shift. Comput Aided Surg 2004;8(5):241–246.
8. Soza G, Grosso R, Nimsky Ch, Greiner G, Hastreiter P. Estimating Mechanical Brain Tissue Properties with Simulation and Registration. In: Proc. MICCAI. vol. 2. Lect Notes Comput Sc, Springer; 2004. p. 276–283.
9. Karypis G, Kumar V. Multilevel k-Way Partitioning Scheme for Irregular Graphs. J Parallel Distr Comp 1998;48(1):96–129.
10. Balay S, Buschelman K, Eijkhout V, Gropp WD, Kaushik D, Knepley MG, et al. PETSc Users Manual. Argonne National Laboratory; 2004.