ELSEVIER
SAUNDERS

Neurosurg Clin N Am 16 (2005) 65–75

NEUROSURGERY
CLINICS
OF NORTH AMERICA

Quasi–real-time neurosurgery support by MRI processing via grid computing

Heiko Lippmann, PD*, Frithjof Kruggel, MD

Max-Planck-Institute for Human Cognitive and Brain Sciences, Stephanstraße 1, D-04103 Leipzig, Germany

Motivation

In neurosurgery, the resection of brain tumors may be planned on the basis of high-resolution preoperative anatomic MRI scans of the patient's head. To gain more information about the function of regions adjacent to the tumor, a functional MRI (fMRI) scan may also be acquired. This information is useful in assisting the surgeon's navigation during the intervention and in minimizing the risk of potential functional damage.

After the skull is opened, several effects take place that lead to nonlinear distortions of the brain, which are collectively called the brain shift phenomenon. Thus, functional information acquired before surgery cannot be easily mapped onto anatomic images acquired during surgery. This is the major shortcoming of image-guided surgical planning based on fMRI data acquired before surgery, because the occurrence of surgically induced deformations invalidates positional information about functionally relevant areas.

Referring to observations in the article by Nabavi et al [1] and a survey by Ferrant et al [2], brain shift is understood as a nonrigid and relatively slow process. The deformation of the brain during surgery mainly occurs because of physical manipulation of the tissue: dura opening, retraction and resection, and draining and leakage of cerebrospinal fluid (CSF). Further impact on brain shift occurs from physiologic reactions of the brain to anesthetics and osmotic agents as a result of the properties of living tissue as well as conditions that are different from a normal state. The opening of the dura and the leakage of CSF cause a gravitational shift because of the disappearance of tension and pressure forces at the brain and ventricular boundaries. Further on, retraction and resection of brain tissue always affects neighboring brain tissue. The change in blood pressure as well as hydration or dehydration of the brain caused by medication administered during surgery causes the patient's brain to swell or shrink. All these factors lead to deformation of the brain during surgery.

This problem has been addressed previously in many publications. For example, Nimsky et al [3] report a manual brain shift correction procedure using intraoperative MRI. This technique is reported to be quite effective and can be done in 15 minutes. Unfortunately, it is limited to anatomic data only and is also costly.

Hastreiter et al [4] try to visualize brain shift using a deformable surface model, which they apply to pre- and intraoperative data sets to show the movement of the brain's surface. To take into account the shift of structures that lie below the brain surface, a voxel-based approach is used, which allows calculation of volume deformation. They use mutual information to perform a nonlinear registration of three-dimensional (3D) piecewise linear patches gained from the transformed data set. Although no quantitative statements referring to accuracy are made, neither of the presented techniques seems to model the brain shift extremely accurately.

Other approaches [2,5–9] try to model the brain shift using a biomechanical finite element model (FEM). These models operate with a limited

This work was supported by the European Union under the IST Programme Framework V, Project IST-2001-37153.

* Corresponding author.
E-mail address: lippmann@cbs.mpg.de (H. Lippmann).

doi:10.1016/j.nec.2004.07.009

number of nodes; the number of nodes is usually lower than the resolution of the volumetric image from which the FEM was extracted. Thus, not all the brain's fine structure can be modeled. These models use forces that are estimated from the movement of the brain surface using stereo cameras or intraoperative MRI scans to deform the biomechanical brain model. This deformation can be applied to almost any type of preoperative data for correction with respect to brain shift. Most approaches model the brain as a homogeneous tissue that only deforms elastically, however. Correct matching of the brain surface during surgery is quite difficult, and interaction between deformed brain tissue and skull is always a problem in these models. Another problem occurs if tissue (eg, tumor) is resected. Here, the surface matching is much harder; in addition, the FEM needs to be updated to reflect the changes in topology. As far as we know, when using the FEM, the only method that is capable of modeling brain shift if tissue has been resected is the one by Ferrant et al [2]. If cameras are used to track the surface's deformation as in the study by Skrinjar et al [6], problems with specularities on the wet brain surface occur, which complicates the stereo reconstruction. Despite all these problems, the accuracy of these techniques is described to be sufficient if no tissue has been resected, and computation time (caused by the use of parallel computers in some cases) is reported to be acceptable in clinical use.

A similar technique using a physics-based approach is used by Hagemann et al [10]. These investigators use a biomechanical model of the human head comprising different materials, such as bone and brain tissue, with different elasticities. Requiring manual interaction, such as tumor and resection area outlining, the method achieves good registration results using synthetic and real patient data. The disadvantage is that the method has only been tested in two-dimensional (2D) imaging and an extension to 3D imaging may be computationally expensive.

Hata et al [11] tried to measure volumetric brain deformation using intraoperative MRI with a volumetric optical flow method. Because of vulnerabilities to contrast variations in intraoperative data, they observed that the best results were achieved if the skin was segmented out. They reported an average error of 3.5 mm, determined from landmarks on a phantom test model. Unfortunately, they also applied their method only to anatomic data sets before and after the opening of the dura mater, so no discontinuities caused by tissue resection are reported.

In this article, the problem is addressed by nonrigid image registration using a viscous fluid model proposed by Christensen et al [12] with respect to intraoperative MRI. The core idea is the nonlinear registration of preoperative fMRI to intraoperative MRI acquired by an open MRI scanner. Whenever an intraoperative data set is acquired, an image processing chain must be executed, which includes correction of intensity nonuniformities in the scan data; linear registration with a preoperative data set, followed by nonlinear registration to obtain a 3D deformation field; application of the deformation; overlaying of preoperative fMRI data to the intraoperative data; and, finally, transfer to a presentation device.

This chain also has a high computation cost that scales with the resolution of the processed MRI scans. To decrease computation time, parallel computers are used. We present a parallel image processing chain that is able to fulfill these requirements.

The chain is currently under development in the European grid-enabled medical simulation services (GEMSS) project [13], which examines the possibilities of the use of grid computing for medical applications. With the utilization of distributed computing over the Internet, which tries to make the power of large computation centers available to simple Internet-connected terminals, we try to achieve an execution time for this chain within 10 minutes.

Grid computing

The idea of providing the computation power of large computation centers to simple terminals over the Internet is understood as grid computing. The aim is to develop a transparent service that abstracts from the Net infrastructure. The user of such a service should not have to care about the availability of certain servers or computation centers somewhere in the Net. He simply requests a service from the grid, and the grid provides the necessary resources using software agents. Those agents are able to determine the computation centers that are capable of providing the requested service with respect to the special conditions specified by the user. Such conditions may include special requests referring to execution time, cost of the service, or security issues.

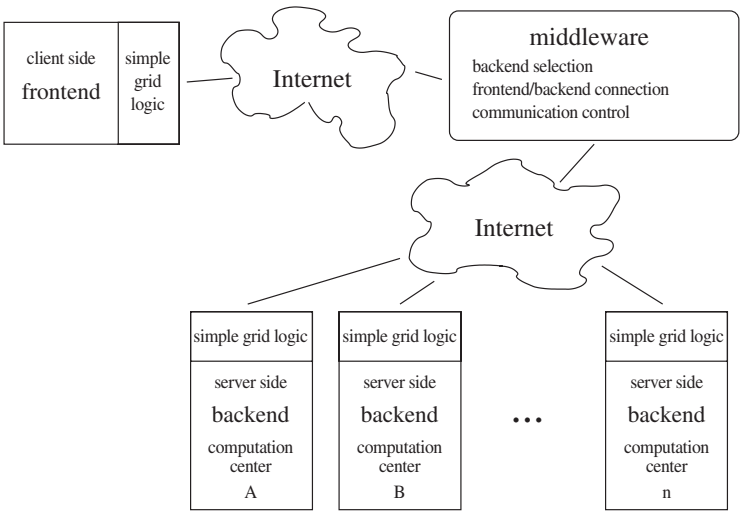

Fig. 1. Grid infrastructure within the grid-enabled medical simulation services (GEMSS) project.

Various projects such as the National Aeronautics and Space Administration's Information Power Grid or European Centre for Nuclear Research DataGrid have been launched to explore the possibilities of the realization of grid environments and to develop the necessary infrastructure. Current grid technologies (eg, Globus Toolkit, Globus Alliance, http://www.globus.org) have evolved principally to meet the needs of high-performance computing resource sharing in the academic community. Those technologies are designed for scientific purposes. When considering their use to support outsourcing of medical applications on a commercial and quasi–real-time basis, it is clear that they do not support the necessary business models, quality of service, and tightly controlled access to applications and data. The GEMSS project was initiated to meet those requirements.

Within the GEMSS project, the grid is realized in a three-tier environment consisting of the client side, middleware, and application back end. To grid-enable an application, it must be split up into two parts. The first part is usually the application front end, which is computationally insensitive, together with simple grid communication logic. On this side, all the input parameters and service requirements are defined by the user. The second part is the often parallel, computational expensive back end, which must also be able to communicate with the grid to obtain its input data and to transmit its output data. The connection between these two parts of an application forms the middleware component, which acts as a service broker. The middleware is contacted by the front end, and the service request is transmitted. Examining this request, the broker chooses and contacts an appropriate back end. Once the link between the front end and the back end has been established, all the applications communication is controlled by the middleware via a simple interface. The client simply loads up input data and waits for the job to start. If the job has been started, it can be monitored or, if necessary, canceled. If the data have been transferred, the application on the back end is invoked with the information as to where it can find its input data and where to place its output data and status information. After the completion of the back end's task, the status is set to "finished." The client detects that through the grid interface and initiates the download of the back end's output data and presents it to the user. Note that all the communication that takes place flows through the middleware over encrypted channels. Fig. 1 shows the described infrastructure schematically.

A scenario similar to the one described here is applied to the image processing chain described in this article. The vision is that the information the user has to provide to run the application is reduced to a minimum. The relevant data here are the location of the patient data and the requested maximum run time. Irrelevant information for surgery, such as where the job is run, how much memory is needed, or how many processors are used, is determined automatically.

Because of the fact that the grid infrastructure developed within GEMSS project should be used

for medical applications, which often operate on the basis of patient data, security is an important aspect. In the framework of the project, weaknesses of the used Net infrastructure as well as possible attacks, together with the costs they would produce if they are left undetected, are examined. To avoid the abuse of grid services or confidential data, all grid-participating parties are equipped with standard security measures, such as firewalls, encryption methods, and secure communication protocols to authenticate communications and data and protect them from unauthorized access. Additionally, in-depth security is maintained, including standard practices like logging.

Beyond that, measures, such as intrusion detection, security audits, and the possibility of installing security updates, are examined. It is also planned to apply software security, such as virus scanners on systems, for example, on machines running clients of certain medical services, where this will be useful. On each grid site, a trust model will be created for each specific security infrastructure to determine security weaknesses. It can then be seen how weak links in a security infrastructure affect the stronger links, and action can be taken to reduce the dependency on the weak links or to improve the security strength of the weak links.

The overall goal is to enable GEMSS project participants to use the grid without excessive cost while maintaining acceptable security.

Description of the image processing chain

In this section, a detailed description of the exact application flow of the image processing chain is given. The required steps of the chain include the following:

1. Transfer of anatomic images from the scanner
2. Correction of intensity nonuniformities in the scan data caused by magnetic field inhomogeneities of the scanner
3. Linear registration of the usually low-resolution intraoperative scan with a high-resolution preoperative scan
4. Intensity adjustment between both scans to improve the results of nonlinear registration
5. Nonlinear registration of both scans to yield a 3D deformation field
6. Application of the deformation field to the preoperative fMRI data set
7. Overlay of the deformed functional information onto the intraoperative data
8. Conversion and transfer to the presentation device (eg, monitor, surgical microscope)

In this chain, steps 2, 3, and 5 are quite time-consuming. Fortunately, these steps can be parallelized.

As requirements for the chain to work, a preoperative MRI scan, together with its aligned fMRI data on the patient, is needed. During the first stage of surgery before the skull is opened, a generally low-resolution image (A) is acquired with the open MRI scanner. After the correction of possible radiofrequency (RF) field inhomogeneities, linear registration of this image with the anatomic high-resolution preoperative data takes place. The nine optimal registration parameters, which include three translation, three rotation, and three scaling parameters, are stored as a starting position for further linear registration steps. The registered image is the reference for further steps of the chain.

After the skull is opened, a sequence of intraoperative images (B) is acquired. These images are also corrected with respect to possible intensity nonuniformities and are registered with the first intraoperative data set A using the stored parameter set P as the initial position. Before the nonlinear registration step is executed, a linear intensity adjustment is performed to obtain the same intensity distribution in both input images A and B. The resulting displacement field of the nonlinear registration process is applied to the preoperative fMRI data. In the final step, the deformed fMRI scan is overlaid on the linearly registered open-skull data set B and later sent to a presentation device.

The nonlinear registration method that is based on fluid dynamics produces the best results if its input images are acquired by the same scanner. That is the reason why the first (closed skull) intraoperative image, A, is acquired and used as a reference image for further processing and not the preoperative high-resolution data, which are usually not acquired with an open MRI scanner. Fig. 2 shows the chain schematically (Fig. 3).

Elements of the image processing chain

A brief overview of the theory behind the individual steps of the image processing chain, together with their limitations and problems, is given in this section.

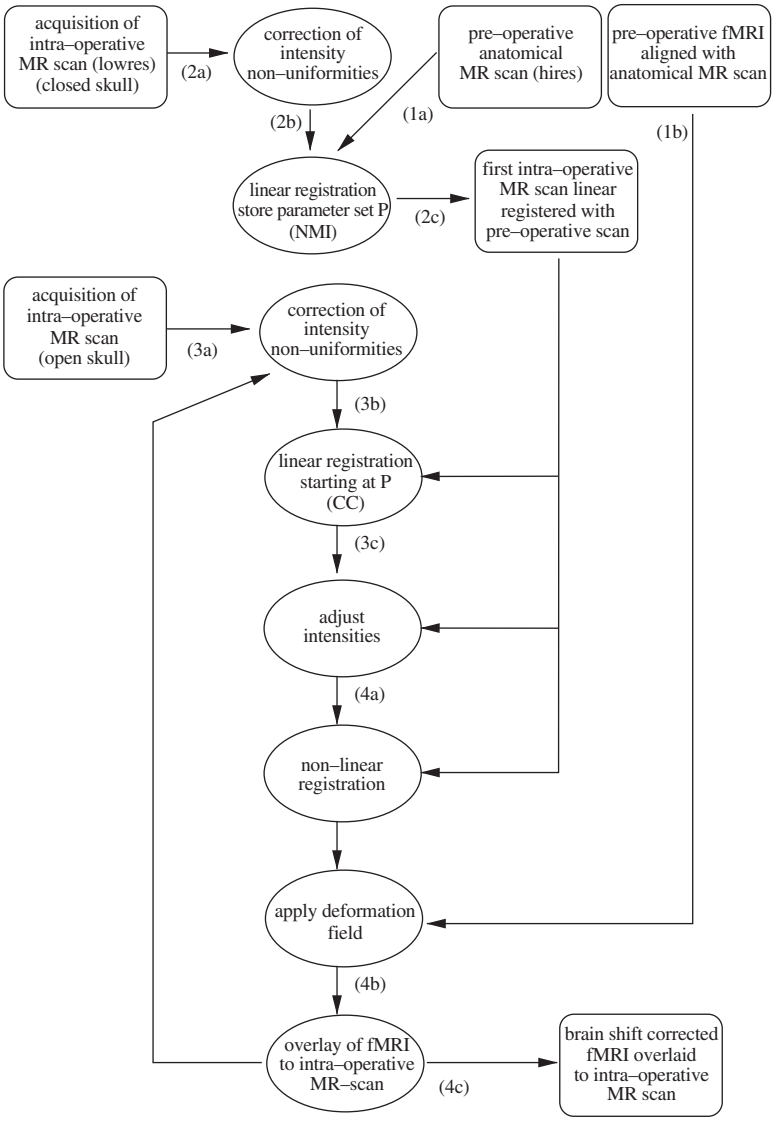

Fig. 2. Image processing chain. The result of each step written in brackets is shown in the corresponding subfigure of Fig. 3.

Data transfer and conversion

Currently, an implementation of this step does not exist. Instead, input data are expected to be in files in a machine-independent format. Later, a direct transfer from the open MRI scanner to the application for intraoperative data is planned.

Correction of intensity nonuniformities

With today's MRI techniques, acquired data often contain inhomogeneities that may be caused by nonuniformities in the RF field of the scanner during acquisition. These inhomogeneities can cause serious misregistration during later processing steps. To address this problem, an adaptive fuzzy C-means (AFCM) algorithm [14,15] is used. Although this algorithm was originally a segmentation method, it can be applied to obtain intensity-corrected MRI scans, because it has been shown [16] that correction of inhomogeneities is strongly coupled with segmentation. The AFCM algorithm segments an MRI scan while estimating a multiplicative bias field that can be used to correct image intensities.

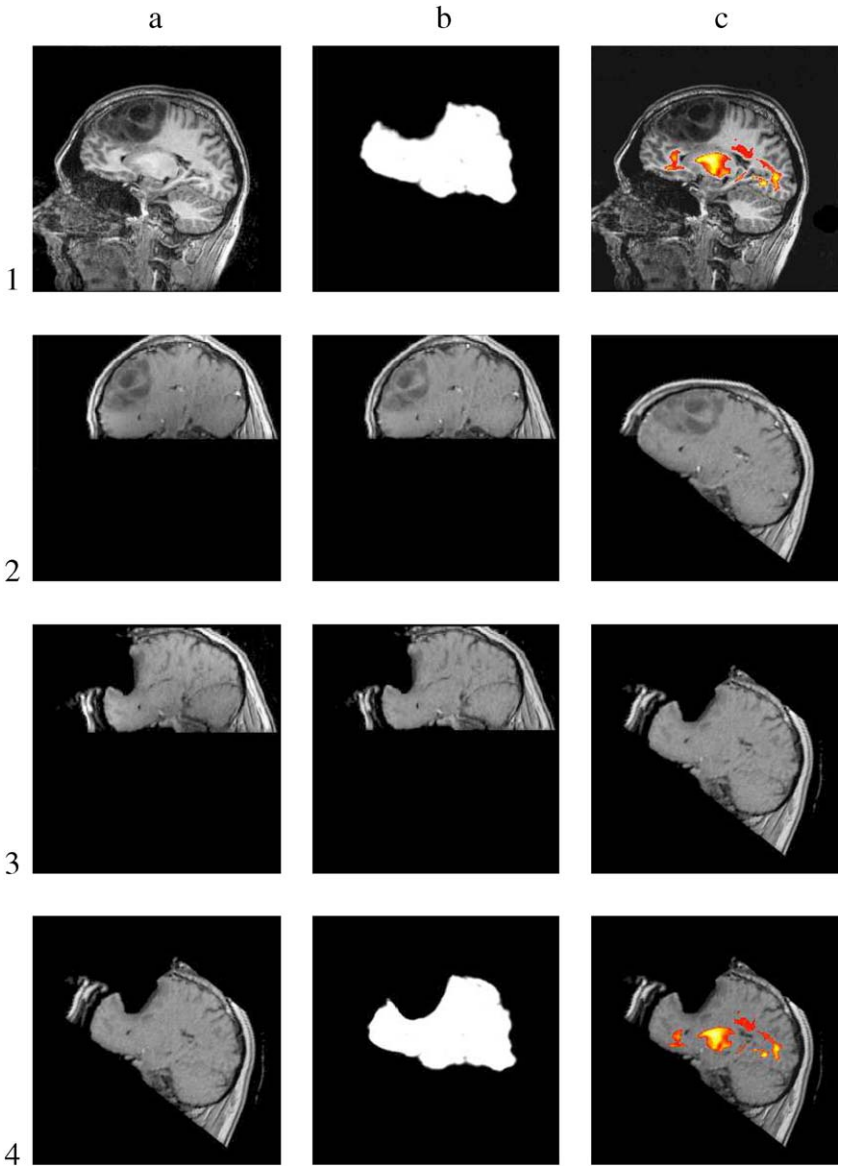

Fig. 3. Single steps of the image processing chain. Subfigures correspond to the results of the step written in brackets in Fig. 2. Subfigure 1*C* shows overlaid preoperative anatomic MRI and functional MRI data. Because of the lack of corresponding fMRI data, a synthetically generated data set was used.

Application of the AFCM algorithm is time-consuming because of the solution of a huge linear system. To improve speed, the linear system is only coarsely solved within a multiresolution framework, which is sufficient to estimate a proper bias field. Further speed improvement is obtained by restricting the analysis to a head mask extracted from the data set. The greatest improvement in performance is obtained by computing the solution of the linear system in parallel. The algorithm was enhanced to work in a multiprocessor shared memory environment with nearly linear speed-up. This was realized by an overlap of iterations. Thus, the next iteration is starting as soon as all necessary data have been computed, although the current iteration is not yet finished. This is the reason why the maximum number of usable processors is limited by the number of slices in the MRI scan.

A different strategy is applied in a distributed memory environment. Here, the data set is partitioned into blocks, and each block is solved independently of the others. If the volume is split up into n blocks, n−1 additional blocks are processed, which are centered over the borders of two neighboring blocks. If the algorithm converged for each block, the n blocks are put together again; on the borders of those blocks, where the convergence is usually worse than near the center of a block, slices from the additional n−1 blocks are filled in. Finally, the last iteration of the algorithm runs serially on the full volume to correct remains of the artifacts in the data caused by the partitioning.

Linear registration

The registration of the low-resolution intraoperative data set to a high-resolution preoperative image is realized by maximizing the normalized mutual information (NMI) [17] or the cross-correlation (CC), respectively. NMI is used when data from different scanners are registered, and CC is used for data sets originating from the same scanner. Fourier-Mellin transform-based methods [18,19] cannot be applied, because the Fourier spectra of intraoperative and preoperative images can vary dramatically because of their possibly different structures (eg, a few intraoperative slices should be registered with a full preoperative image of the head).

Linear registration has to embrace translation, rotation, and scaling that define a nine-dimensional search space in 3D space. To achieve a fast convergence, the downhill simplex optimization algorithm is used, which performs well [20] and does not require any gradient information. Computation time of the gradient of the NMI cost function, for example, would be nine times greater than the evaluation time of the cost function itself. To improve the speed of the registration, a parallel evaluated speculative downhill simplex that converges exactly like the original method but twice as fast was developed. Further nearly linear speed-up was gained by evaluating the cost function (NMI and CC) in parallel by partitioning the data into blocks and assigning each block to a single processor. The results of the processed subvolumes are totalled to obtain the cost function value. All these parallel computed parts of the linear registration can be used in a shared and distributed memory environment.

Currently, the success of linear registration depends on the initial orientation of both images, because it cannot be assumed that the global optimum is found during optimization. A strategy needs to be developed to increase the probability of convergence to the global optimum.

Intensity adjustment of two scans

The result of the nonlinear registration step of the processing chain depends, among other things, on the similarity of the intensities of both images. To achieve this, a codomain of tissue voxels in both images is computed. The mean and standard deviation of voxel intensities of the source image are then adjusted to match those of the reference.

This step is optional because it is often not necessary. Because it is not time-consuming, it can be performed serially.

Nonlinear registration

To obtain a deformation field that can be applied to an fMRI scan, nonlinear registration is required. In this chain, a method based on fluid mechanics [12,21] is applied. The time-consuming part here is, once again, the solution of a huge linear system. To speed this up and to avoid local minima, the system is solved using a multiresolution approach. Further speed-up is achieved by solving the system in parallel in a shared memory environment. Here, each processor operates on a single "slice" of the system.

If there is only a distributed memory environment available, all multiresolution steps except the last one, which needs more computation time than all the other steps before combined, are computed serially. For the last resolution level, the data are again partitioned into blocks, and each block is processed by its own processor in parallel. The blocks are computed independently of each other. To avoid partitioning artifacts, the blocks have an overlap of two slices on each border.

In Fig. 4, the brain's surface within the intraoperative data set with the opened skull, together with the deformation field obtained from the nonrigid registration with the intraoperative data set with the closed skull, is visualized. Regions colored in red mean a contraction, and regions colored in blue mean an expansion of the brain tissue. Additionally, small arrows indicate the direction of the shift of the tissue.

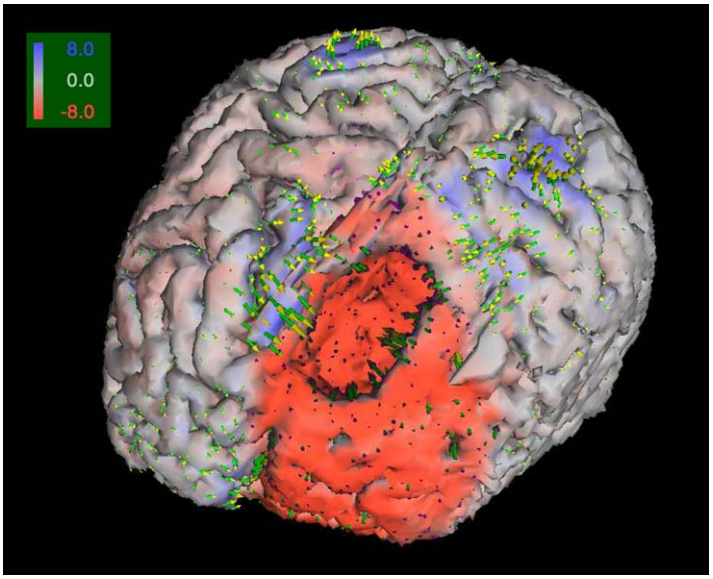

Fig. 4. Surface of the brain within the intraoperative data set with an opened skull, together with the deformation field obtained by nonrigid registration. Red means contraction, and blue means expansion of the brain volume. Arrows indicate the direction of the tissue shift.

Application of a deformation field to functional MRI data

The deformation field obtained by nonlinear registration is applied to an fMRI data set by shifting each voxel of the data set by its corresponding vector from the displacement field. This step is not time-consuming and can be performed serially.

Overlay of the deformed functional MRI data with intraoperative data

In this step, which is again performed serially, the deformed preoperative functional data set that was initially aligned with the preoperative anatomic MRI scan is overlaid to the rigidly registered intraoperative image to show regions of activation with respect to the brain shift.

Conversion and transfer to presentation device

This step has also not been implemented yet. Data are converted into a format that can be viewed with a visualization tool. As in step 1, automatic transfer of the data to a chosen presentation device is imaginable.

Executing the chain over the grid

To decrease the computation time of an unoptimized serial processing chain, which needed approximately 4 hours on a Pentium III single processor with 500 MHz, to a clinically acceptable 10 minutes, the software experienced a lot of optimization. In addition, we will probably need a parallel machine with 10 to 12 processors at a clock rate of 1.6 GHz to achieve this goal. This would mean that a personal computer (PC) cluster has to be available not far away from the MRI scanner to compute a usable result within 10 minutes.

Because of the fact that parallel computers are expensive, need maintenance, and are often underloaded, the idea of using external computation centers on demand is attractive. This is the main idea behind grid computing. The presented image processing chain will be implemented to work within the grid infrastructure provided by GEMSS project.

A common use for our application could be that before an operation begins, an available computation center is determined by the GEMSS project middleware. After automatically negotiating terms of necessary resources, payment, and security issues, the preoperative as well as closed-skull intraoperative anonymous patient data are transferred to the target computation center. Right after that, the preprocessing can start. After the acquisition of another intraoperative MRI scan, now with the skull opened, the new data are

transferred to the center and the main part of the image processing chain is executed remotely. After the computation has finished, the result is transferred to the operating room. There, the data can be displayed in a surgical microscope or monitor. Now, further intraoperative data sets can be uploaded, and the main part of the chain can be restarted.

The only computation hardware that is required in the operating room is an inexpensive terminal connected to the MRI scanner, the Internet, and a presentation device. Of course, the grid service is not free of charge, but compared with the setup and maintenance of a local PC cluster, which will probably need a special room with a controlled climate, this may be an interesting alternative.

Quality of service

The middleware that is under development within the GEMSS project provides access to computation centers using common Web transport protocols like hypertext transfer protocol (HTTP). For security reasons, the secure socket layer-encrypted version of HTTPs is used. The client side does not require a special network environment; specifically, network ports do not need to be opened, which always entails a potential security risk. If a Web browser operates correctly, communication with the middleware is ensured. This framework also does not need any modification of current firewall configurations. The only requirement is a fast Internet connection to have low transfer times for the MRI data sets. The size of one data set can vary from 4 to 12 MB for an anatomic data setup to 50 MB for an fMRI data set.

Before any transfer over the Internet is initiated, patient data can be made anonymous, if necessary, to protect the identity of the patient.

The server side of the application, which can be installed in all the computation centers that plan to provide this special service, contains the described image processing chain controlled by the middleware. In its final version, the middleware estimates resource requirements of the current application using abstract performance models. Such a performance model may, for example, predict the application's memory use and its execution time, depending on the current input parameters and the current load of the computation center. These parameters as well as the price for the requested service are compared with the requirements stated by the client. Based on this negotiation, a suitable computation center is chosen.

Performance

Currently, the performance and the quality of the results of the chain are examined on a local PC cluster. The cluster contains four nodes with two Advanced Micro Devices AthlonMP (Advanced Micro Devices, Sunnyvale, California) 1800+ processors at a clock rate of 1.6 GHz. Nodes are equipped with 1 GByte of random-access memory and connected via a 100-MBit/s Ethernet network. Each component of the chain uses a message passing interface framework to communicate with other processes in this distributed memory environment. Table 1 shows the execution times of the chain for a different number of used processors compared with the serial execution of the chain on one processor compared with the run time in a two-processor shared memory environment (SHR) provided by a single node of the cluster and compared with a large PC cluster used over the grid. The last column of the table lists transfer times for the data sets, including data packing and unpacking, but does not include any overhead introduced by communication over the grid middleware. The values are gained for the processing of only one real data set because of the lack of further patient data. The preoperative MRI scans have a resolution of $256 \times 192 \times 256$ voxels, and the intraoperative data sets have a resolution of $256 \times 256 \times 60$ voxels.

The poor speed-up gained by using 10 processors instead of 1 is caused by the high communication overhead and because of the fact that each parallel component contains serial parts that cannot be parallelized. In a distributed memory environment, data must be transferred multiply between the nodes. This overhead grows with an increasing number of processors. In a SHR, which is used seldom because of its high price, all processors can directly access the data, and this problem does not appear.

Varying the number of processors always produces a different partitioning of the data. This can result in the effect that with one partitioning, the finding of the optimum requires more iterations than with another partitioning. Thus, it can occur that although there are more processors used now than before, a special step of the chain requires more computation time, although there are more processors used as seen in the

Table 1
Performance of the chain

	SER (l)	SHR (2)	MPI (4)	MPI (6)	MPI (8)	GRD (10)
Preprocessing						
Upload	0 min 0 s	0 min 0 s	0 min 0 s	0 min 0 s	0 min 0 s	0 min 44 s
Inhom	2 min 51 s	2 min 15 s	2 min 36 s	2 min 39 s	2 min 29 s	2 min 16 s
Linreg	4 min 21 s	3 min 4 s	2 min 14 s	2 min 13 s	2 min 5 s	1 min 47 s
Σ	6 min 12 s	5 min 19 s	4 min 50 s	4 min 52 s	4 min 34 s	4 min 47 s
Processing						
Upload	0 min 0 s	0 min 0 s	0 min 0 s	0 min 0 s	0 min 0 s	0 min 8 s
Inhom	2 min 58 s	2 min 20 s	2 min 44 s	2 min 28 s	2 min 19 s	2 min 1 s
Linreg	8 min 16 s	5 min 13 s	3 min 42 s	2 min 33 s	2 min 31 s	1 min 50 s
Intens	0 min 4 s	0 min 4 s	0 min 4 s	0 min 4 s	0 min 4 s	0 min 3 s
Fluidreg	6 min 56 s	4 min 31 s	3 min 57 s	4 min 2 s	2 min 39 s	2 min 34 s
Shift	0 min 21 s	0 min 16 s	0 min 12 s	0 min 16 s	0 min 16 s	0 min 20 s
Overlay	0 min 4 s	0 min 4 s	0 min 4 s	0 min 4 s	0 min 4 s	0 min 3 s
Download	0 min 0 s	0 min 0 s	0 min 0 s	0 min 0 s	0 min 0 s	0 min 9 s
Σ	18 min 39 s	12 min 28 s	10 min 43 s	9 min 27 s	7 min 53 s	7 min 8 s
Overall Σ	24 min 51 s	17 min 47 s	15 min 33 s	14 min 19 s	12 min 27 s	11 min 55 s

Abbreviations: Fluidreg, nonlinear registration; GRD, grid; Inhom, correction of inhomogeneities; Intens, intensity adjustment; Linreg, linear registration; min, minutes; Overlay, overlay of fMRI to anatomic MRI data; s, seconds; SER, serial execution; Shift, apply deformation field; SHR, shared memory environment.

The number of processors is shown in brackets. The values for GRD have been obtained on a different cluster.

preprocessing section of Table 1 between the columns showing the results for four and six processors as well as in the processing section for fluid registration.

As already mentioned, the required number of processors is approximately 10 to 12 to complete the chain in 10 minutes in a grid scenario, including transfer times and input/output times of every single software component. In praxis it has to be taken into account that on a grid site, input/output times can vary significantly, depending on the current load of the cluster that might be produced by other tasks.

Summary

One problem of providing time-critical medical services over the grid is always its dependency on the Internet. It cannot be assumed that transfer of a certain amount of data over the Internet is always achieved during a specified period. Such a requirement cannot be fulfilled by the infrastructure of the Web. There is always the risk of a network delay or even an overload. Because of this, another goal of this project is the evaluation of grid services versus the use of local services.

A further point for future research related to the chain has to deal with the optimization approach for the linear registration step. Because the optimization uses the downhill simplex algorithm in a nine-dimensional search space, the number of iterations needed to find the optimum can vary dramatically. This makes linear registration the most unpredictable step of the chain in terms of execution time. It cannot be assured that the global optimum is found.

Additional work has to be done in validating the registration accuracy, including the examination of the influence of intensity variations between intraoperative images as well as the influence of tumor resection and the presence of the opened skull versus the closed skull in the fluid-based registration.

References

[1] Nabavi A, Black PM, Gering DT, Westin CF, Mehta V, Pergolizzi PS, et al. Serial intraoperative MR imaging of brain shift. Neurosurgery 2001;48: 787–98.

[2] 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;6:337–59.

[3] Nimsky C, Ganslandt O, Hastreiter P, Fahlbusch R. Intraoperative compensation for brain shift. Surg Neurol 2001;56:357–65.

[4] Hastreiter P, Rezk-Salama C, Nimsky C, Lürig C, Greiner G, Ertl T. Registration techniques for the

analysis of the brain shift in neurosurgery. Comput Graph 2002;24:385–9.

[5] Ferrant M, Warfield SK, Nabavi A, Jolesz FA, Kikinis R. Registration of 3D intraoperative MR images of the brain using a finite element biomechanical model. Delp SL, DiGioia AM, Jaramaz B, editors. In: Medical image computing and computer-assisted intervention—MICCAI 2000. Berlin: Springer; 2000. p. 19–28.

[6] Skrinjar O, Nabavi A, Duncan J. Model-driven brain shift compensation. Med Image Anal 2002;6:361–74.

[7] Warfield SK, Ferrant M, Gallez X, Nabavi A, Jolesz FA, Kikinis R. Real-time biomechanical simulation of volumetric brain deformation for image guided neurosurgery. In: High performance networking and computing conference—SC 2000. IEEE Computer Society: Washington, DC; 2000. p. 1–16.

[8] Paulsen K, Miga M, Kennedy F, Hoopes P, Hartov A, Roberts D. A computational model for tracking subsurface tissue deformation during stereotactic neurosurgery. IEEE Trans Biomech Eng 1999;46:213–25.

[9] Clatz O, Delingette H, Bardinet E, Dormont D, Ayache N. Patient specific biomechanical model of the brain: application to Parkinson's disease procedure. In: International symposium on surgery simulation and soft tissue modeling (IS4TM) 2003, vol. 2673. Lecture Notes in Computer Science, Institut National de Recherche en Informatique et en Automatique Sophia Antipolis. Springer: Heidelburg; 2003. p. 321–31.

[10] Hagemann A, Rohr K, Stiel HS, Spetzger U, Gilsbach JM. Non-rigid matching of tomographic images based on a biomechanical model of the human head. In: Hanson E, editor. SPIE medical imaging 1999. Proceedings of the SPIE International Symposium. Society of Photo-optical Instrumentation Engineers: Bellingham, Washington; 1999. p. 583–92.

[11] Hata N, Nabavi A, Wells WM, Warfield SK, Kikinis R, Black PM, et al. Three-dimensional optical flow method for measurement of volumetric brain deformation from intraoperative MR images. J Comput Assist Tomogr 2000;24:531–8.

[12] Christensen G, Joshi S, Miller M. Volumetric transformation of brain anatomy. IEEE Trans Med Imaging 1997;16:864–77.

[13] GEMSS project. Grid-enabled medical simulation services. Available at: http://www.gemss.de. Accessed September 8, 2004.

[14] Pham DL, Prince JL. An adaptive fuzzy c-means algorithm for image segmentation in the presence of intensity inhomogeneities. Patt Recog Lett 1999; 20:57–68.

[15] Pham DL, Prince JL. An adaptive fuzzy segmentation algorithm for three-dimensional magnetic resonance images. Information Processing in Medical Imaging. Lecture Notes in Computer Science 1999; 1613:140–53.

[16] Styner M, Brechbühler C, Szekely G, Gerig G. Parametric estimate of intensity inhomogeneities applied to MRI. IEEE Trans Med Imaging 2000; 19:153–65.

[17] Pluim JPW, Maintz JBA, Viergever MA. Mutual information based registration of medical images: a survey. IEEE Trans Med Imaging 2003;22: 986–1004.

[18] Chen Q, Defrise M, Deconinck F. Symmetric phase-only matched filtering of Fourier-Mellin transforms for image registration and recognition. Transactions on Pattern Analysis and Machine Intelligence 1994; 16:1156–68.

[19] Reddy BS, Chatterji BN. An FFT-based technique for translation, rotation, and scale-invariant image registration. IEEE Trans Image Process 1996;5: 1266–71.

[20] Maes F, Vandermeulen D, Suetens P. Comparative evaluation of multiresolution optimization strategies for multimodality image registration by maximization of mutual information. Med Image Anal 1999;3:373–86.

[21] Wollny G, Kruggel F. Computational cost of non-rigid registration algorithms based on fluid dynamics. IEEE Trans Med Imaging 2002;11:946–52.