AUSTRIAN GRID

A REFINED DESIGN OF THE SEE-GRID DATABASE AND PATHOLOGY FITTER

Document Identifier:	**AG-DA-1c-5-2005_v1.doc**
Workpackage:	**A1c**
Partner(s):	**Research Institute for Symbolic Computation (RISC)** **Upper Austrian Research (UAR)**
Lead Partner:	**RISC**
WP Leaders:	**Wolfgang Schreiner (RISC), Michael Buchberger (UAR)**
Privacy:	**Public**

Delivery Slip

	Name	Partner	Date	Signature
From	Károly Bósa	RISC	2005.11.29	
Verified by				
Approved by				

Document Log

Version	Date	Summary of changes	Author
1.0	2005-11-28	Initial Version	See cover on page 3

A Refined Design of the SEE-GRID Database and Pathology Fitter

Karoly Bosa
Wolfgang Schreiner

Research Institute for Symbolic Computation (RISC)
Johannes Kepler University Linz
{Karoly.Bosa, Wolfgang.Schreiner}@risc.uni-linz.ac.at

Michael Buchberger
Thomas Kaltofen

Department for Medical Informatics
Upper Austrian Research (UAR)
Thomas.Kaltofen@uar.at

November 29, 2005

Abstract

SEE-GRID is based on the SEE++ software for the biomechanical simulation of the human eye. The goal of SEE-GRID is to adapt and to extend SEE++ in several steps and to develop an efficient grid-based tool for ``Evidence Based Medicine", which supports the surgeons to choose the best/optimal surgery techniques in case of the treatments of different syndromes of strabismus.

First, we developed the "SEE++ to Grid Bridge", via which normal SEE++ clients are able to access and exploit the computational power of the Austrian Grid. We have implemented a distributed and grid-based version of the Hess-Lancaster test, which is a medical experiment for the diagnosis of strabismus and whose original sequential simulation is time consuming in SEE++. Then, we also implemented a prototype version of the grid-enabled pathology fitting algorithm, which attempts to determine (approximately) the pathological reason of strabismus in case of a patient.

In this document, we present some extended benchmark results of the parallel Hess Lancaster test, in which we used more grid resources and reached greater speedup values as before.

Next, we describe the current state of the grid-enabled distributed medical database that we started to develop in the previous phases of the project for collecting, sorting and evaluating patient's data and both real and simulated pathological cases. Then we outline the further development steps related to the SEE-GRID database.

In the last section of this document, we discuss and evaluate the possible designs of grid-based Pathology Fitting and Surgery Fitting algorithms, which we plan to implement at a later phase of the project.

1 Introduction

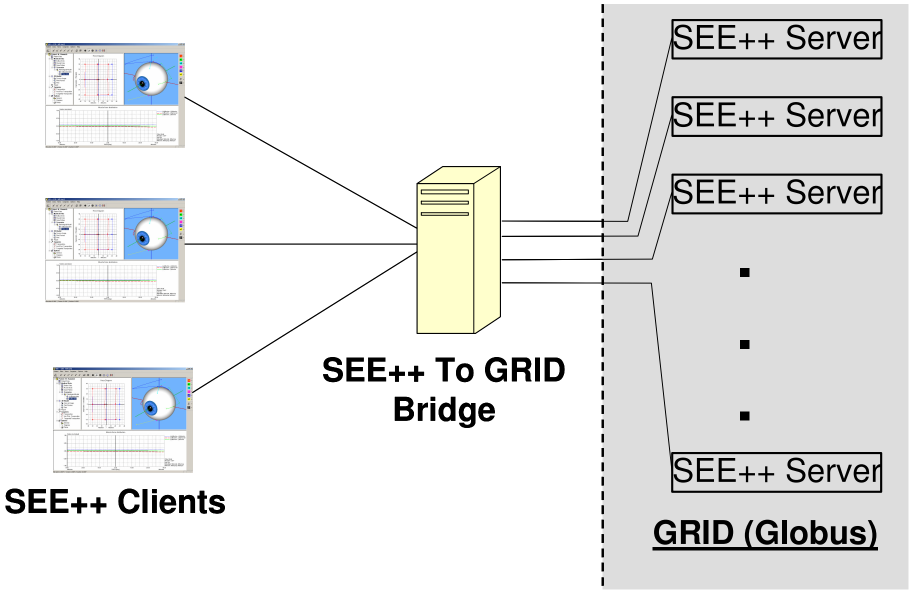

Figure 1: The Current Architecture of SEE-GRID

The design of SEE-GRID is based on the SEE++ software for the biomechanical simulation of the human eye and its muscles. SEE++ was developed in the frame of the SEE-KID project by Upper Austrian Research and the Upper Austria University of Applied Sciences [SEE-KID, Buchberger 2004, Kaltofen 2002]; it simulates the common eye muscle surgery techniques in a graphic interactive way that is familiar to an experienced surgeon. SEE++ consists of a client component for user interaction and visualization and a server component for running the actual calculations; the message protocol SOAP is used for communication between the two components.

SEE++ deals with the support of diagnosis and treatment of *strabismus*, which is the common name given to usually persistent or regularly occurring misalignment of the eyes. Strabismus is a visual defect in which eyes point in different directions. A person suffering from it may see double images due to misaligned eyes. SEE++ is able to simulate the result of the *Hess-Lancaster test*, from which the pathological reason of strabismus can be estimated. The outcome of such an examination is two gaze patterns (see Figure 3) of blue points and of red points respectively. The blue points represent the image seen by one eye and the red points the image seen by the simulated other eye, but in a pathological situation there is a deviation between the blue and the red points. The default gaze pattern that is calculated from the patient's eye data by SEE++ contains 9 points. But there exist gaze patterns with 21, 45 or more points (bigger gaze patterns provide more precise results for the decision support in case of some pathologies, but their calculations are more time consuming).

In SEE++, a third gaze pattern, a measured one (with green points) of a patient can be given as input. In this case, the goal is to take some default or estimated eye data and to modify a subset of them until the calculated gaze pattern of the simulated eye (red points) matches the measured gaze pattern. This procedure is called *pathology fitting*. The original algorithm is time consuming and gives only a more or less precise estimation for the pathology of the patient.

In the previous phases of the SEE-GRID project [SEE-GRID, 2005/1], we implemented the "SEE++ to Grid Bridge". It is the initial component of SEE-GRID, via which the normal SEE++ client can get access to the infrastructure of the Austrian Grid (see Figure 1). The SEE++ clients can access this application in the same way as in the original SEE++ system; the usage of grid resources is completely transparent to them.

The ``SEE++ to Grid Bridge" is able to split gaze pattern calculation requests of clients to independent subtasks and to distribute them among the servers. By this, we demonstrated how normal SEE++ clients are able to access via this bridge (see Figure 1) to the Austrian Grid and how a noticeable speedup can be reached in SEE++ — by applying simple data parallelism — by the exploitation of the huge computational power of the Grid. Then, we also developed a prototype version of the grid-enabled pathology fitting algorithm, whose goal is to determine (approximately) the pathological reason of strabismus in case of a patient.

In the current phase of the project, we made some extended benchmark with parallel gaze pattern calculation, see Section 2. We finished the implementation of the first version of the SEE-GRID database, which works as a Web Service application at the moment, see Section 3. At last, we analysed and made a detailed evaluation on the possible designs of grid-enabled pathology and surgery fitting algorithms, see Section 4.

2 Extended Benchmarks

Originally, we investigated the effectiveness of the parallelism in different situations where 1, 3, 5 or 9 processes of the SEE++ server were started on the grid [SEE-GRID 2005/1]. By starting 9 server processes, we speeded up the simulation of the Hess-Lancaster test by a factor of 3-4.

Machine Name	**altix1.jku.austriangrid.at**			**Altix1.jku.austriangrid.at** **hydra.gup.uni-linz.ac.at**		
Server processes / Max. number of points sent together	**1/all**	**3/3**	**9/1**	**25/1**	**30/1**	**45/1**
Changing the Total Strength of one Muscle on one Eye	**25.2703s**	**17.4387s**	**7.5788s**	**1.9498s**	**1.8533s**	**1.7831s**
Changing the Total Strengths of two Muscles on one Eye	**27.1793s**	**18.8115s**	**9.1101s**	**2.1737s**	**2.1010s**	**1.8915s**
Changing the Total Strengths of two Muscles on both Eyes	**28.6750s**	**20.0424s**	**9.7951s**	**2.2291s**	**2,1881s**	**1.9016s**

Table 1: Benchmark Results in case of the Calculation
of the Brainstem Gaze Patterns (with 45 points)

Now, we extended these test cases with some new ones, where 25, 30 or 45 server processes were started on two grid sites. As before, the maximum number of the gaze pattern points that are sent together to one server process (granularity) was "not limited", 5, 3, 2 or 1. We also used different gaze pattern sizes, like 9, 21 and 45 points. In case of 25 or more server processes (see Table 1 and Figure 2), we speeded up to 10-12 times the simulation of the Hess-Lancaster test (despite of the communication overhead). Each value located in the following tables is the median execution time of 5-7 executions.

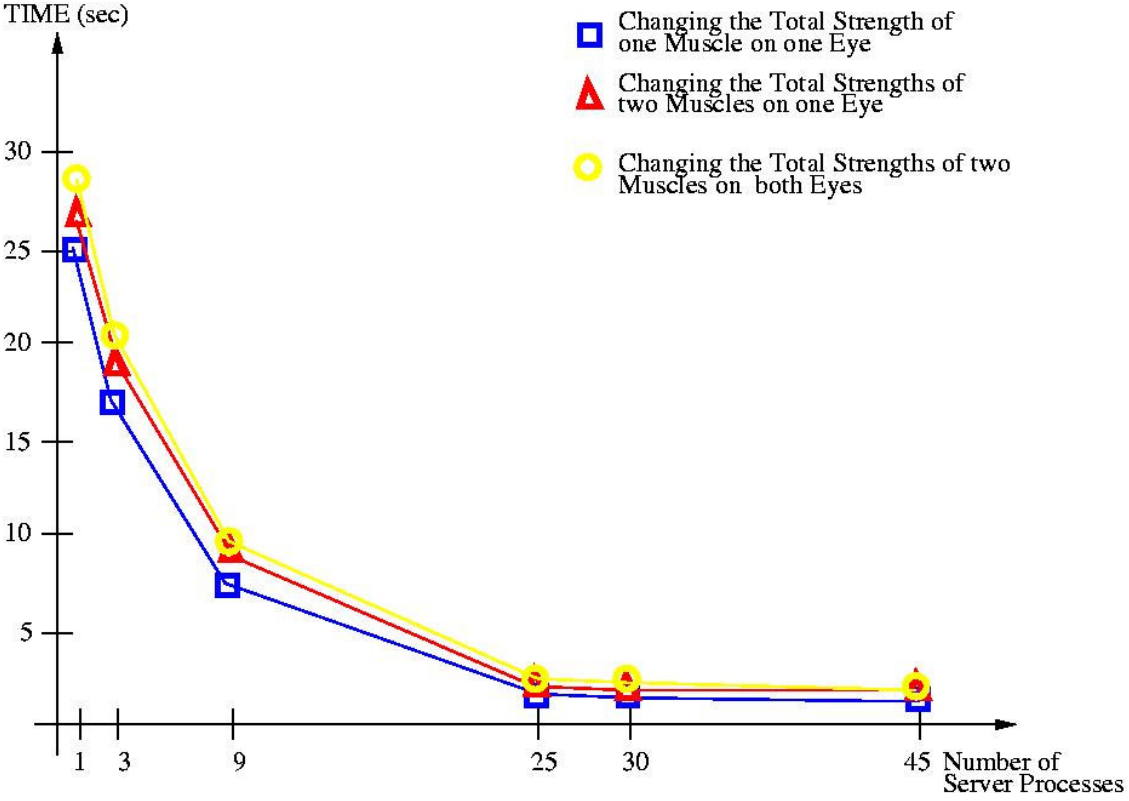

Figure 2: Speedup curves in case of the Calculation
of the Brainstem Gaze Patterns (with 45 points)

The test cases were executed on the Austrian Grid site altix1.jku.austriangrid.at, which contains 64 pieces of Itanium processors (1.4GHz). In case 25 or more server processes, we also started some SEE++ servers up to 10 server processes on another grid site called hydra.gup.uni-linz.ac.at (hydra is a cluster that contains 14 pieces of AMD Athlon 1.6GHz CPUs). But either all processes were started only on the altix1 or some of them were also started on hydra, the measured benchmark values were very similar (actually in the first case we usually got a little bit better result with 100-300 msec).

For measuring, we installed the Ethereal network protocol analyzer [Ethereal, 2004] on the machine where the SEE++ client is executed. By this software, the network traffic between the local machine and the grid portal machine was filtered and each network package sent to or received from the port of “seepp2grid” was captured. After the execution of a test case, the duration time of the calculation can be determined from the recorded capture time of the first sent and of the last received message.

In those medical tests, where not only one but two gaze patterns are used at the same time (each of them is assigned to left or to the right eye) for diagnostic purposes, more speedup may be reachable by further enlarging the number of the server processes running on the grid.

3 The Design of the SEE-GRID Database

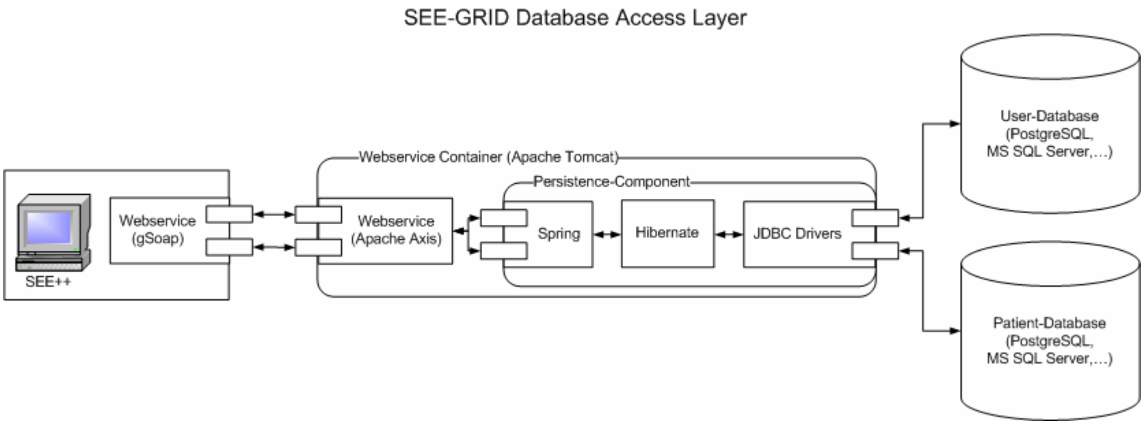
SEE-GRID Database Access Layer
SEE++
Webservice (gSoap)
Webservice Container (Apache Tomcat)
Webservice (Apache Axis)
Persistence-Component
Spring
Hibernate
JDBC Drivers
User-Database (PostgreSQL, MS SQL Server,...)
Patient-Database (PostgreSQL, MS SQL Server,...)

Figure 3: SEE-GRID Database Access Layer

In SEE-GRID, a distributed grid-based database is going to be used for storing and sorting patient data with gaze patterns and eye data.

3.1 Current State

In the first step, a medical database for SEE++ was designed [Mitterdorfer, 2005] and developed as a Web Service application (see Figure 3). The SEE++ client interacts with the database via the SOAP protocol. Also the communication protocol of SEE++ was extended with some additional SOAP messages used by this database application. The Web Service fuctionality on the server side is implemented and provided by *Apache Axis*. Later, this component can be substituted by a grid-enabled database interface component (see Section 3.2).

For mapping the implemented object-oriented data structures to relational data structures, an open source tool, called *Hibernate* is used. Hibernate aims to be a transparent source *Object/Relational* (O/R) mapping framework, which means that the objects need not implement specific interfaces or extend a special base class. For easily accomplishing this O/R mapping, we used up the predefined Hibernate functionality contained by the application framework *Spring*. Philosophy of the Spring framework is not to create new solutions for problems already solved but to integrate existing solutions and simplify their usage. For directly communicating with the data basis, JDBC database drivers are used.

The medical data of SEE++ (e.g.: patient's data, simulated and measured gaze patterns, result of medical experiments, etc.) are stored in the Patient-database (see Figure 3). The metamodel does not only support SEE++, it was designed for supporting general medical database [Mitterdorfer, 2005].

Since the SEE-GRID database is designed for storing patient records, security is a very important aspect. The user database (see Figure 3) contains the user authentication and authorization information of the system. The security implementation ensures that every Web Service call is secured appropriately by checking the caller's identity. Furthermore, the persistence component employs many techniques to maximize security:

- encrypting any network transfer via HTTPS or SSL,
- applying certificates on client and server side to ensure mutual trust,
- applying strong encryption of user passwords with a SHA-512 salted hash.

All cryptographic algorithms are based on proven standards to maximize security. The security component is not tied to the persistence component at all. Therefore, it can be maintained separately and used for other purposes.

3.2 Next Development Steps

The proposed grid database will be based either on G-SDAM (Grid Seamless Data Access Middleware) architecture [G-SDAM, 2005] or on the Web Service technologies applied in Globus 4. Since both of them are able to communicate via the SOAP protocol with other grid-based applications, our database implementation is flexible enough and it can be easily adapted to them.

The data sets of the database may be collected by manual insertion of patient data (respectively by automatic transfer of data entered in local databases into the grid base) as well as by automatic insertion of the computed simulation data.

By the SEE-GRID database, a huge number of medical cases will be easily available for users/surgeons, but also the proposed parallel pathology fitter will be based on it (see Section 3.2). By searching in this database for corresponding input eye data sets for the pathology fitter and by starting concurrent pathology fitting processes on some grid sites,

- We may get better solutions than in the case of the existing algorithm;
- We may get more then one solution which may relevant to the actual pathological situation of the patient;
- The execution of the solutions may take less time, since we will have good estimations at the very beginning.

Since we intend to distribute the implementation of the database over multiple grid nodes, we must develop a parallel/distributed search algorithm so that computational processes will be able to access and to collect the necessary and most relevant information from this distributed grid database for the pathology fitting.

4 Evaluation for Further Development

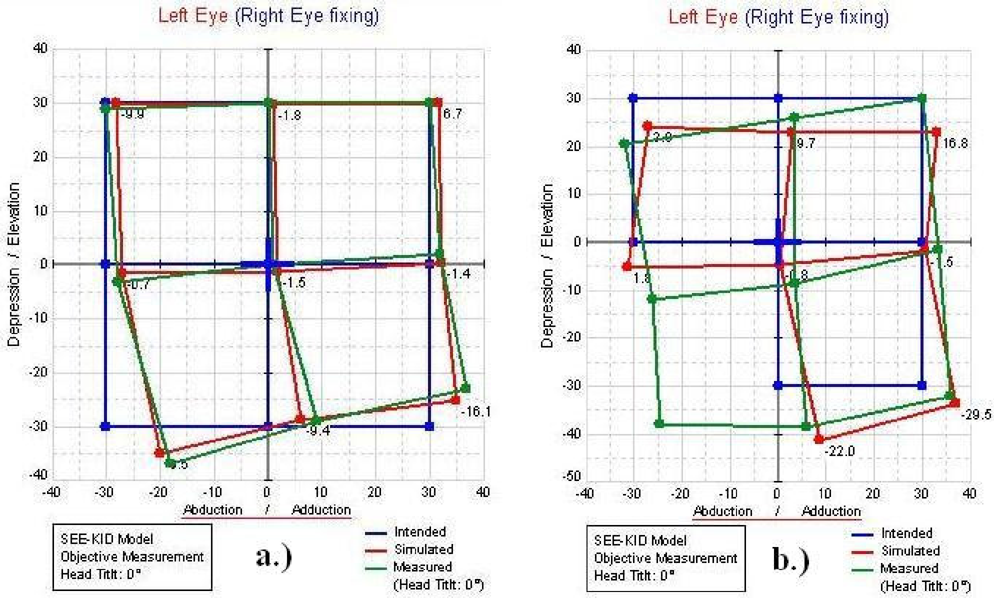

Figure 4: Examples for Gaze Patterns: Intended (blue lines), Measured (green lines) and Simulated (red lines)

This section contains an evaluation and design analysis of the grid-based pathology fitting and surgery fitting algorithms, which will be developed in the later phases of the SEE-GRID project.

4.1 Overview on the Pathology Fitting

The goal of the pathology fitting is to determine (approximately) the pathological cause of strabismus from which a patient suffers. A pathology fitting process takes an initial parameterization of both eyes and gradually “improves” it (by modifying the different kind of parameters of the eye muscles) until the gaze pattern calculated with the biomechanical eye model matches the measured pattern of the patient (see Figure 4). Since the measuring of gaze pattern is not perfect and precise usually, the simulated gaze patterns almost never will be completely the same as the measured one.

Unfortunately, a gaze pattern does not uniquely determine the values of eye model parameters. Hence, a new term, called *strategy* is introduced in the SEE++ software system, which is derived from some other medical examinations (besides the Hess-Lancaster test) by the doctors. The strategy works as some kind of heuristic and it can estimate which eye data parameters may be most effected in the current strabismus syndrome from which a patient suffers (the reasons of different syndromes are the disease of different sets of muscle parameters). The strategy is a list of particular muscle parameters in a specific order for the pathology fitter in order to exclude most of the possible incorrect solutions. Only these given kinds of muscle parameters are allowed the pathology fitter to modify.

Roughly, the pathology fitting works in the following way [SEE-GRID, 2005/2]:

- On the highest level, the algorithm selects the different kind of eye muscle parameters contained by the strategy one by one. The simulated eye model updated by the last calculated values of the currently modified parameters is used as input for the optimization of the next kind of parameters (if and only if the fitting of the currently modified parameter yields any improvement).
- On the lower level, a non-linear optimization algorithm (currently Levenberg-Marquard is used) modifies the muscle parameters selected by the strategy in several iterative optimization steps. The same kind of muscle parameters is always modified together.
- On the level of the optimization steps, the algorithm performs some computations by which it tries to determinate the next improvement values of the given data (Jacobian and Hessian matrices are computed among others).
- At the end of each optimization step, a gaze pattern is calculated with the modified eye data for the evaluation of the improvement comparing with the previous state.

4.2 Possibilities for Speeding up the Pathology Fitting

The next two subsections contain a detailed discussion about what kind of parallel or sequential strategies we have investigated for speeding up the existing algorithm.

4.2.1 Parallelization of the Existing Algorithm

We investigated the possibilities how we can improve the pathology fitting algorithm by parallelization:

- On the highest level the different kinds of eye data parameters have to be modified sequentially in a specific order given by the strategy. Hence, the only possibility for parallelizing the algorithm on this level is to find a non-sequential strategy/heuristic instead of the current one.
- The optimization algorithm itself is an iterative algorithm, in which the last computed result is always used in the next optimization step. Hence, there is no possibility to parallelize the algorithm on this level.
- On the level of the optimization steps, we can parallelize the computation of each optimization step (parallelizing the computations of the Jacobian and Hessian matrices) as it is described in [Parallel LevMarq.]. But the algorithm uses too small

parameter vectors (the sizes of the previously mentioned matrices are small, therefore, their computation takes much less than 1 second) and it has too many iterative optimization steps. Because of the communication overhead, we may not be able to reach any speedup.

- On the lowest level at the end of each optimization step, a gaze pattern is calculated with the modified eye data. Since a pathology fitting process often requires the calculation of approx. 60-100 gaze patterns, we combined the sequential pathology fitting algorithm with the parallel gaze pattern calculation in previous project work. By this, we could improve the algorithm and reach some limited speedups [SEE-GRID, 2005/2].

4.2.2 Speeding up the Sequential Algorithm

To improve the optimization algorithm, it may be also possible to execute the optimization of the parameters given in the strategy concurrently by for instance some *weighted optimization* (however this will not result parallel processes, which can be distributed to independent resources, just less iterative optimization steps).

The basic idea is to assign different numbers to different kind of muscle parameters (greater numbers to those parameters that usually have more important effects for the supposed pathological syndrome). By assigning the same numbers to some different kind of muscle parameter, we also can cluster some parameters, if the order of their modification is not strictly determined.

We introduce some small values like $\varepsilon_1 > \varepsilon_2 > \varepsilon_3 \dots$ etc. First, only the muscle parameters with greatest weight are altered until the deviation between the computed and the measured gaze pattern is less then ε_1. Then we start to modify the next kind of parameters with smaller weight until the deviation between the gaze patterns is less than ε_2. The process proceeds analogously until the algorithm modifies each kind of eye muscle parameters given in the strategy.

By this weighted optimization, we might get similar result as in the case of the current algorithm, but with fewer sequential steps.

4.3 Possibilities for Finding Better Solutions

Since a gaze pattern does not uniquely determine a simulation model and the current algorithm may not find always the best solution, we can exploit the grid infrastructure to attempt to find better solution by some techniques which helps us to avoid local minima during the optimization (e.g.: simulated annealing).

The pathology fitting process may not always search for the global minimum, since the given strategy works like a heuristic and may lead the algorithm to some desired local minimum. However, if we mix the simulated annealing technique [Sim. Annealing] and the heuristic, it can also help us to find better local minima.

In this case, the application may give a list of the best solutions found and the users/doctors can choose the most relevant one for the actual pathological situation.

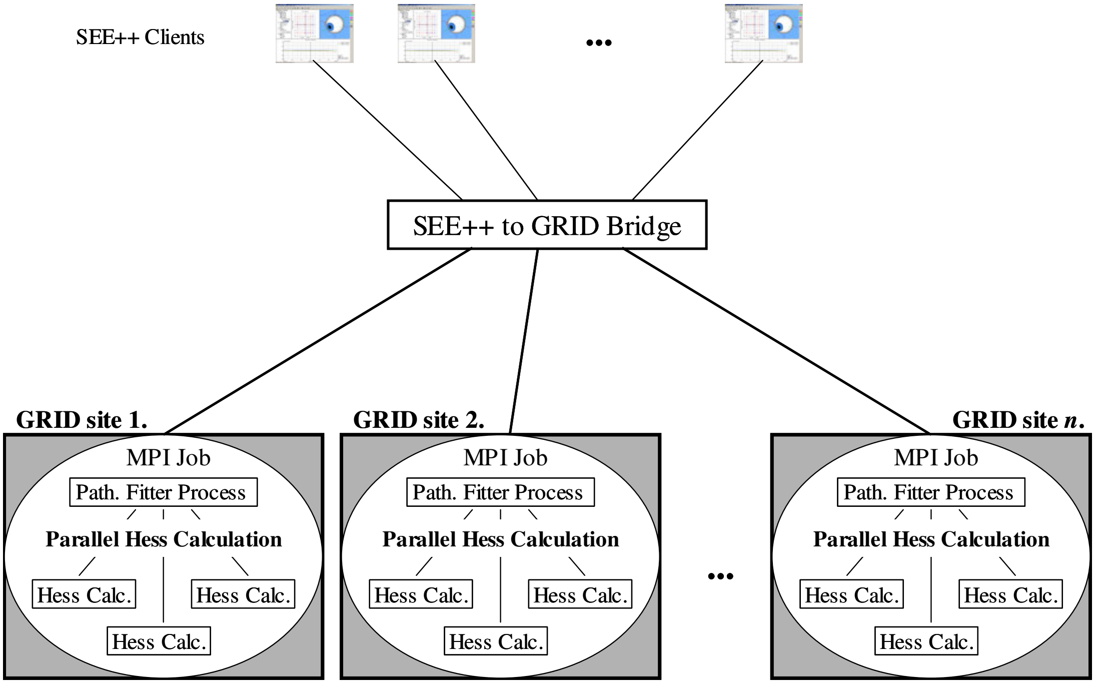

Figure 5: Encapsulating the pathology fitting and parallel gaze pattern calculation into MPI jobs

Another possibility for improving the output of the algorithm is to search in the SEE-GRID database in order to find similar cases by taking the measured gaze pattern (and maybe the name of the supposed syndrome from which the current patient suffers) as input. Then, we can start concurrent and independent pathology fitting processes on some grid nodes with the founded eye models as initial estimations for the optimizations. Furthermore, the computed results will also be stored in the database as feedbacks for providing better and better initial estimations for later computations.

Later, we intend to encapsulate the combined pathology fitting and parallel gaze pattern calculation into a MPI process and execute multiple such fitting processes on different grid sites (see Figure 5).

4.4 Surgery Fitting

At the last phase of the project, we plan to develop a surgery fitting method, which is able to support the doctors to find the best/optimal surgery technique to correct efficiently the vision of the patients. The problem of surgery fitting is similar to pathology fitting, but this case the result of the pathology fitting is taken as input and it is modified until the calculated gaze pattern closely matches the gaze pattern of the healthy eye. Further differences are:

- Since only few eye parameters (e.g.: length of the muscles, muscle insertion on the eye globe, etc.) can be changed by surgery, the set of those eye parameters, which the surgery fitting algorithm can modify, is only a subset of the parameters which could be modified by the pathology fitting.

- The final goal of the surgery fitting is to correct the (stereo) vision of a patient as much as possible (finding a global minimum). Since the intended optimal state can be uniquely determined by a particular gaze pattern, we do not have to apply a sequential heuristic as in the case of pathology fitting.

- There is only one possible input data set, this is the fitted pathology (or if the pathology fitter returned more than one solutions, then the doctor has to choose one).

From these it can be seen, that we do not need to execute the same application with different input parameters on the grid multiple times as in case of the pathology fitting. However we have to be sure that the optimization algorithm avoids the local minima. Therefore, during the surgery fitting we should branch the calculation sometimes and start some other fitting processes on the grid with some kinds of neighbourhood values of the reached intermediate improvement of the eye data (simulated annealing [Sim. Annealing]).

Furthermore, since we need not use any sequential heuristic on the highest level of the algorithm, we may have a chance to speed up the algorithm itself by implementing some parallel search technique (e.g.: parallel Branch and Bound [Aida et. al. 2003, Aida Osumi 2005, Filho et. al., 2003]) on the grid.

5 Conclusions

Based on the results and investigations described in this paper, our ongoing research work has three main directions:

- **Algorithmic improvements:** we have started to investigate the strategies outlined in Section 4.2.1 and Section 4.2.2 in order to achieve a better understanding of the fundamental pathology fitting algorithm and thus to devise new, faster sequential or parallel algorithms.

- **Implementation of the grid-based database:** we currently work on the adaptation of the existing prototype implementation of the SEE-GRID database to the Austrian Grid.

- **Development of a grid-based parallel search technique for pathology fitting:** by using the SEE-GRID database, we are going to implement a variant of pathology fitting. This variant will apply a grid-based parallel search technique to find cases in the SEE-GRID database that are similar to measured patient data; it will then execute multiple independent pathology fitting processes on the grid.

References

[Aida et. al., 2003] Kento Aida, Wataru Natsume, Yoshiaki Futakata. *Distributed Computing with Hierarchical Master-worker Paradigm for Parallel Branch and Bound Algorithm*, 3rd International Symposium on Cluster Computing and the Grid, May 12 - 15, 2003, Tokyo, Japan. http://csdl.computer.org/comp/proceedings/ccgrid/2003/1919/00/ 19190156abs.htm

[Aida, Osumi, 2005] Kento Aida, Tomotaka Osumi, "A Case Study in Running a Parallel Branch and Bound Application on the Grid," Proc. IEEE/IPSJ The 2005 Symposium on Applications & the Internet (SAINT2005), pp.164-173, Feb. 2005.
http://www.alab.ip.titech.ac.jp/papers/saint2005.pdf

[SEE-GRID 2005/1] Károly Bósa, Wolfgang Schreiner, Michael Buchberger, Thomas Kaltofen. *The Initial Version of SEE-GRID*. Austrian Grid Deliverable A1c-1-2005, Research Institute for Symbolic Computation (RISC), Johannes Kepler University, Linz, March 2005.

[SEE-GRID, 2005/2] Karoly Bosa, Wolfgang Schreiner, Michael Buchberger, Thomas Kaltofen. *A Prototype of the SEE-GRID Pathology Fitter*. Austrian Grid Deliverable A1c-3-2005, Research Institute for Symbolic Computation (RISC), Johannes Kepler University, Linz, July 2005.

[Buchberger, 2004] Michael Buchberger. *Biomechanical Modelling of the Human Eye*.
Ph.D. thesis, Johannes Kepler University, Linz, Austria, March 2004.
http://www.see-kid.at/download/Dissertation_MB.pdf

[G-SDAM, 2005] *A Report on a Unified Grid-aware Access Layer for SEE-GRID Data Sets.* Austrian Grid Deliverable M-4aA-1c, FAW Institute and RISC-Linz Institute, Johannes Kepler University, Linz, August 2005.

[Kaltofen, 2002] Thomas Kaltofen. *Design and Implementation of a Mathematical Pulley Model for Biomechanical Eye Surgery*. Diploma thesis, Upper Austria University of Applied Sciences, Hagenberg, June 2002.
http://www.see-kid.at/download/Pulley_Model_Thesis.pdf

[Mitterdorfer, 2005] Daniel Mitterdorfer. *Grid-Capable Persistance Based on a Metamodel for Medical Decision Support*. Diploma thesis, Upper Austria University of Applied Sciences, Hagenberg, July 2005.

[Ethereal, 2004] Ethereal Network Protocol Analyser. http://www.ethereal.com

[Parallel LevMarq.] N. N. R. Ranga Suri, Dipti Deodhare, P. Nagabhushan. *Parallel Levenberg-Marquardt-Based Neural Network Training on Linux Clusters.* ICVGIP 2002, Ahmadabad, India.

[Sim. Annealing] M. Miki, T. Hiroyasu, M. Kasai, K. Ono, T. Jitta, Temperature Parallel Simulated Annealing with Adaptive Neighbourhood for Continuous Optimization Problem. International Journal of Computational Intelligence and Applications, 2002, pp 149-154.

[Filho et. al., 2003] J. Viterbo Filho, L. M. A. Drummond, E. Uchoa e M. C. S. Castro. *Towards a Grid Enabled Branch-and-Bound Algorithm*. Report RT-05/03, Department of Computer Science -- Fluminense Federal University, 2003.
http://www.ic.uff.br/PosGrad/RelatTec/reltec03.html