Elastic Matching of Very Large Digital Images on High Performance Clusters

J. Modersitzki[1], W. Obelöer[2], O. Schmitt[3], and G. Lustig[2]

[1] Medical University of Lübeck, Institute of Mathematics,
Wallstraße 40, D-23560 Lübeck, Germany
`modersitzki@math.mu-luebeck.de`

[2] Medical University of Lübeck, Institute of Computer Engineering,
Ratzeburger Allee 160, D-23538 Lübeck, Germany
`obeloeer@iti.mu-luebeck.de`

[3] Medical University of Lübeck, Institute of Anatomy,
Ratzeburger Allee 160, D-23538 Lübeck, Germany
`schmitt@anat.mu-luebeck.de`

Abstract. The aim of the human neuroscanning project is to build an atlas of the human brain, based on a variety of image modalities in particular histological sections of a prepared brain. Reconstructing essential information out of deformed images is a key problem. We describe a method to correct elastic deformations. Since the method is computational expensive a parallel implementation is presented. The measurements and results shown are performed on a cluster of 48 Pentium II PCs connected via Myrinet.

1 Introduction

Reconstruction of deformed images is a basic problem within medical image processing (image registration). Especially, if images arise from a series of sections through a part of the human body, e.g. CT (computer tomography), MRI (magnetic resonance imaging), PET (positron emission tomography).

In this paper we concentrate on particular problems of the human neuroscanning project (HNSP) at the Medical University of Lübeck. The aim of the HNSP is to produce a three-dimensional map of a human brain based on different modalities, in particular cellular information. Here, the information is derived mainly from histological sections.

Typically, sectioning processes lead to deformed sections and consequently to deformed images. Although these deformations are small in general, they might become crucial for the reconstruction of cellular information in the human brain.

Two different approaches for correcting this kind of distortions are common. One approach is based on the idea of representing the unknown correction in terms of the coefficients of a fixed basis, such as piecewise linear functions or higher order splines. Typically these coefficients are determined by a least squares condition for some user prescribed landmarks (see, e.g., [3, 4]). The second approach is based on the formulation of the problem via a non-linear partial differential equation (PDE). To solve these equations no further information on the underlying images, e.g. landmarks, is needed (see, e.g. [1,17]).

We describe a method for correcting these kind of deformations based on a linear elasticity model leading to a non-linear PDE. This method is also used in other projects, e.g. [1], [2], [6], [7], [16].

To resolve details of the brain (e.g. neurons) very high resolution scans of the histological sections are needed. This leads to high dimensional problems. Due to memory and computational requirements of the method used a straightforward implementation can not be applied. Hence, we present a fast algorithm and its parallel implementation.

The HNSP project is described in the next section. The elasticity model is given in section 3, whereas section 4 presents our parallel implementation and performance measurements. The results and their medical discussion are given in section 5.

2 The Human Neuroscanning Project (HNSP)

The aim of the HNSP is the three-dimensional reconstruction of all cells of a human brain. These data should be used as the basic structure for the integration of functional data based on stochastic mapping and later on for modeling and simulation studies in such a virtual brain.

In this project a 55 year old male human post mortem brain of a voluntary donator of his corpse was fixed in a neutral buffered formaldehyde solution for 3 months. A MRI-scan of this brain was produced after fixation. Dehydration and embedding of the brain in paraffin required 3 further months. This preparatory work was followed by sectioning the brain in 20 μm thick slices (about 7000 for this brain) by using a sliding microtome. Before each sectioning process a high resolution episcopic image (1352×1795 pixels, 24 Bit, $7 \cdot 10^6$ Bytes) of the section plane was scanned.

After sectioning, the tissue slice was stretched in warm water at 55°C. Thereafter, it was transferred onto a microscopic slide and dried over night. The sectioning, stretching and drying processes are necessary in order to get flat tissue sections. However, this methodological steps produce non-linear deformations of each section (see Fig. 3). After drying, the sections were stained in gallocyanin chrome alum and mounted under cover-glasses in order to visualize all cells (special light microscope for analyzing large sections (LMAS)) in whole brain sections. Different neuronal entities were analyzed on different structural scales i.e. from macroscopic details down to the cellular level. In order to obtain a non-deformed stack of images a so-called elastic matching method was used. Elastic matching can be used also for multimodal matching of histological sections with non-deformed MRI-scans [14] or episcopic images. The later ones might be derived from image processing before sectioning the embedded brain. The stained sections were scanned by a transparent flat bed scanner using a resolution of 800 ppcm (or 2032 ppi) in a 8 Bit gray scale mode (size of the smallest image was 5000×2000 pixels, size of the largest image was 11000×7000 pixels (about 196 MBytes). The uncompressed amount of flat bed scanned data was approximately 700 GBytes + 40 GBytes episcopic data for one human brain.

In the following, the arbitrarily chosen sections 116 and 117 out of a total of about 7000 sections were matched. In the upper left corner of Fig. 3 the reference image (116) is shown and in the upper right corner the template image (117). In these images the left and the right hemisphere of the human brain are shown. These sections were obtained from the occipital lobe.

In this stage of the HNSP the images were scaled down from 6500 × 2300 to 512 × 512 pixels in order to keep the computation times reasonable. A straightforward rescaling based on bilinear interpolation was used.

3 Modeling non-linear deformations of two consecutive sections

We are looking for an elastic deformation of the template image (T) that simultaneously minimizes the difference between the deformed and the reference image (R) and the deformation energy

$$E(u,v) = \int_{\Omega} \frac{\lambda}{2}\left(u_x + v_y\right)^2 + \mu\left(u_x^2 + v_y^2 + \frac{1}{2}(u_y + v_x)^2\right) d(x,y),$$

where the so-called deformation field $(u,v) = \big(u(x,y), v(x,y)\big)$ describes the local deformation and μ, λ are the so-called Lamé-constants, see, e.g., [10]. This approach enforces similarity of the images as well as connectivity of the tissue.

Applying the calculus of Euler-Lagrange we find that a minimizer is characterized by the so-called two-dimensional Navier-Lamé-equations (1), cf. e.g. [10],

$$\begin{pmatrix} f \\ g \end{pmatrix} = \begin{pmatrix} \mu(u_{xx} + u_{yy}) & + & (\lambda + \mu)(u_{xx} + v_{xy}) \\ \mu(v_{xx} + v_{yy}) & + & (\lambda + \mu)(u_{xy} + v_{yy}) \end{pmatrix} =: A \cdot \begin{pmatrix} u \\ v \end{pmatrix}. \qquad (1)$$

Note, $(f,g)^{\top}$ (which might be viewed as a force field) depends non-linearly on the deformation, cf. eq. (2),

$$\begin{pmatrix} f \\ g \end{pmatrix} = \begin{pmatrix} \Big(T(x-u, y-v) - R(x,y)\Big) \cdot T_x(x-u, y-v) \\ \Big(T(x-u, y-v) - R(x,y)\Big) \cdot T_y(x-u, y-v) \end{pmatrix}. \qquad (2)$$

An appropriate discretization of these equations finally leads to a fix-point type equation for the unknown deformation field, cf. eq (3),

$$A\left(u^{k+1}, v^{k+1}\right)^{\top} = \left(f(u^k, v^k), g(u^k, v^k)\right)^{\top}. \qquad (3)$$

In principle, any solver can be used to compute the solution of eq. (3). However, a discretization with $m \times n$ points results in $N = 2mn$ unknowns (e.g. for 512×512 discretization points we end up with $N = 2^{19} = 524288$) and A becomes $N \times N$. For a standard LU-decomposition one needs to store $\mathcal{O}(N^2)$ real numbers and approximately $\mathcal{O}(N^3)$ floating point operations. Thus, memory and computational requirements make a parallel implementation of an iterative solver for eq. (3) unavoidable.

4 Parallel realization and measurements

Due to a good price/performance ratio workstation clusters become an alternative to expensive dedicated parallel computers. Hence, our parallel implementation is performed on the so-called "Störtebeker Cluster" [13]. This cluster consists of 48 dual 333 MHz Pentium II nodes interconnected via Myrinet [5]. The operation system used is LINUX. For the measurements shown in Fig. 1, the parallel program uses PVM [12] as the underlying message passing system. PVM is used in order to support also different and heterogeneous platforms.

4.1 Parallel implementation of the elastic matching algorithm

An implementation of the parallel algorithm for p processes can be divided into three parts (quantities used only locally are denoted with a subscript $_{\text{loc}}$). The repetition of part two and three is called *outer loop*. In the current implementation there is one process per node.

1. Partition the images R and T and the initial deformation field (u^0, v^0) into p stripes and distribute these stripes to p processes. Every process sets $(f_{\text{loc}}^{-1}, g_{\text{loc}}^{-1}) = (0,0)$ and $k = 0$.
2. Every process applies the deformation $(u_{\text{loc}}^k, v_{\text{loc}}^k)$ to T_{loc} and computes the forces $(f_{\text{loc}}^k, g_{\text{loc}}^k)$*independently*.
   If the difference between the new and the old force field is sufficiently small, terminate.
3. Solve the linear system of equations (3) for the new deformation (u^{k+1}, v^{k+1}). Set $k \to k+1$ and continue with step 2.

As already pointed out, the main computational work is needed for solving eq. (3), which has to be done in any step. Here, a parallel implementation of the conjugate gradient method (CG) is used, cf. e.g. [9]. Performing this iterative scheme requires an additional so-called *inner loop*.

The basic structure of our implementation is given in Table 1. The essential computational and communication costs needed in one step of the inner CG iteration are next neighbor communication (exchange local stripes with two neighbors), two global sums (inner products $\approx 4N$ FLOPS), three local SAXPY operations (Scalar Alpha times X Plus Y, $6N/p$ FLOPS for vectors X, Y of length N/p), and one matrix vector multiplication (exploiting the special structure of the matrix A, this multiplication is $\mathcal{O}(N/p)$ FLOPS but depends on the particular discretization). Note, N is the total number of unknowns (i.e. $N = 2mn$ for $m \times n$ discretization points), p is the number of processes.

4.2 Measurements

In order to keep the measurement times reasonable, the number of steps in the outer loop as well as the maximum number of the CG steps in the inner loop is set to 50. Two measurement series with different numbers of nodes (the

Table 1. Principle phases of the parallel implementation of the CG-algorithm.

(a)	computation	matrix-vector multiplication inner product	$q_{\text{loc}} = Ap$ $\alpha_{\text{loc}} = q_{\text{loc}}^{\mathsf{T}} p_{\text{loc}}$
(b)	communication	build and distribute global sum	$\alpha = \sum \alpha_{\text{loc}}$
(c)	computation	two SAXPY's inner product	$x_{\text{loc}} = x_{\text{loc}} + \alpha p_{\text{loc}}$ $r_{\text{loc}} = r_{\text{loc}} - \alpha q_{\text{loc}}$ $\beta_{\text{loc}} = r_{\text{loc}}^{\mathsf{T}} r_{\text{loc}}$
(d)	communication	build and distribute global sum	$\beta = \sum \beta_{\text{loc}}$
(e)	computation	local vector operation	$p_{\text{loc}} = r_{\text{loc}} + \beta p_{\text{loc}}$
(f)	communication	exchange local vectors	$p_{\text{loc}} \leftrightarrow$ right and left neighbor

current implementation only supports one process per node) were performed, one matching of two 256 × 256 pixels images and a second one matching of two 512 × 512 pixels images.

The run time of the sequential version for the 256 × 256 pixels images is about 26 minutes, while the calculation of the 512 × 512 pixels images lasts about 100 minutes. Since the parallel version has a sequential part (i.e. starting the processes, initialization and distribution of the images etc.) it is clear that the speedup can not be linear (optimal). As expected, the speedup becomes better with larger problem sizes. The total run time is still very high, about 188 seconds on a 48 node cluster for the 512 × 512 pixels images.

To give inside into the program behavior three iteration steps of the CG-algorithm are shown in Figure 2. For observation and evaluation of the behavior the performance monitoring tool DELTA-T [11] is used. The *System Load* and computation/communication phases of two arbitrarily chosen processes are shown for an eight node configuration (i.e. eight processes). Figure 2 shows about 0.25 seconds of the execution time. In particular, three principle phases of the CG-algorithm as introduced in Table 1 are displayed.

The upper curve in Figure 2 shows the *System Load.* Here, 100% indicates that all eight processors are working concurrently. An average utilization of nearly 100% is reached, i.e. the parallel implementation of the CG algorithm is able to use almost the full system power. The two Gantt graphs of processes four and eight show calculation phases (black) and communication phases (white). Typically, more than 50% of the execution time of one CG step is taken by phase (a). Building the global sums (phases (b) and (d)) takes more time than exchanging the local vectors in phase (f), although much more data have to be transferred in phase (f).

If more computation nodes are used, the calculation time of phases (a), (c), and (e) is reduced. In contrast, the time for the communication phases (b) and (d)

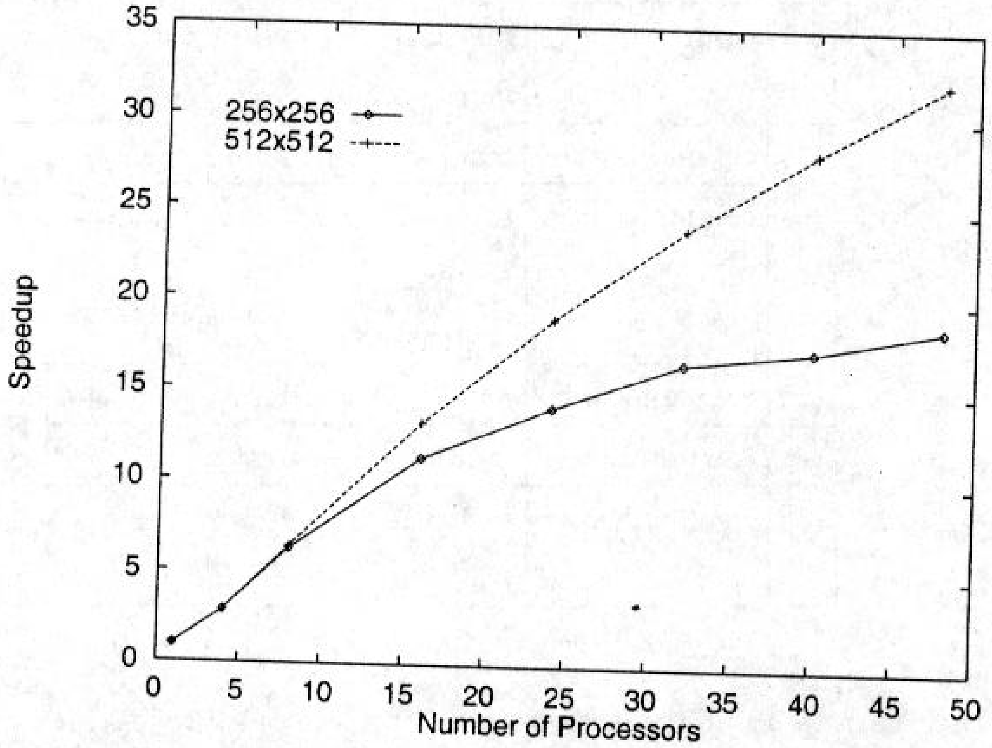


Fig. 1. Speedup for matches of 256 × 256 pixels images and 512 × 512 pixels images.

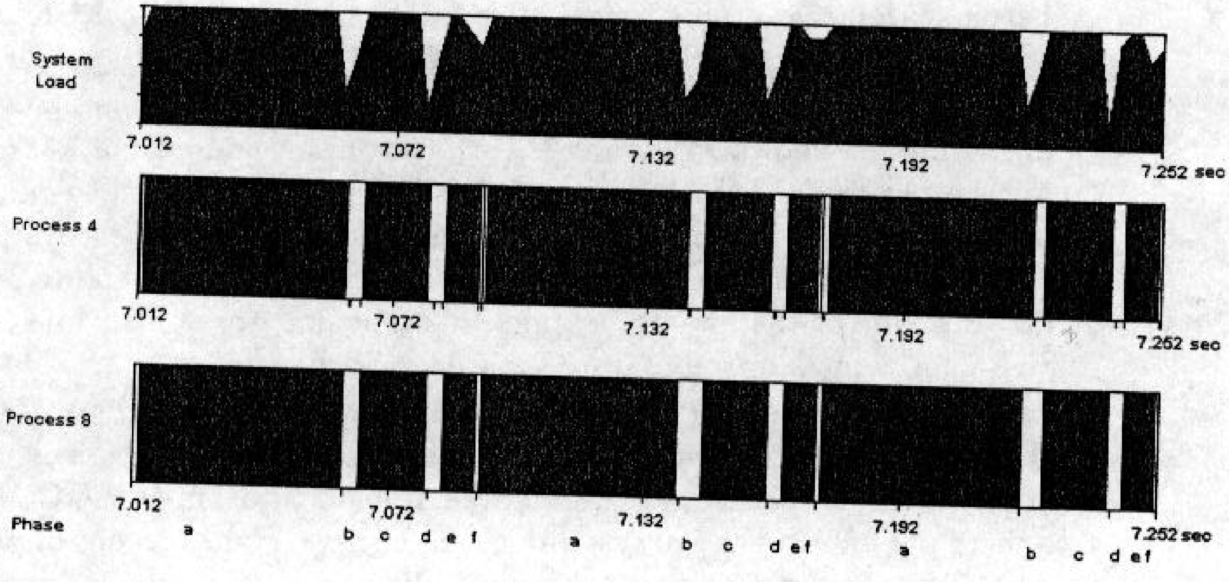


Fig. 2. *System Load* and computation/communication phases of the arbitrarily chosen process 4 and process 8 for an eight node configuration.

is increased. For larger numbers of nodes this leads to a lower efficiency. This might become crucial while matching two images of relatively small sizes. However, the images to be matched in the HNSP project have a high resolution and thus large sizes.

5 Results

Comparing the template 117 with the reference image 116 shows a difference of $||T - R||_2 \approx 10361$ in the Euclidean norm, cf. Fig. 3. After applying 500 steps of the outer loop of the elastic matching algorithm the difference has been reduced to $||T_{500} - R||_2 \approx 3123$, which is ca. 30.1%.

¿From the morphological point of view this can be considered as an adequate result. Irrespectively to the fact that there exists no other method for solving this kind of deformation problem, this technique can be considered as a sophisticated procedure to match images of tissue sections of the whole human brain.

6 Conclusion

The presented method allows the reconstruction of deformed images. The technique is also useful in order to align images from other modalities like PET, MRI, EEG (electroencephalography). In addition this method can be adapted easily to multimodal image matching. Moreover, this method can be extended to a variety of matching problems.

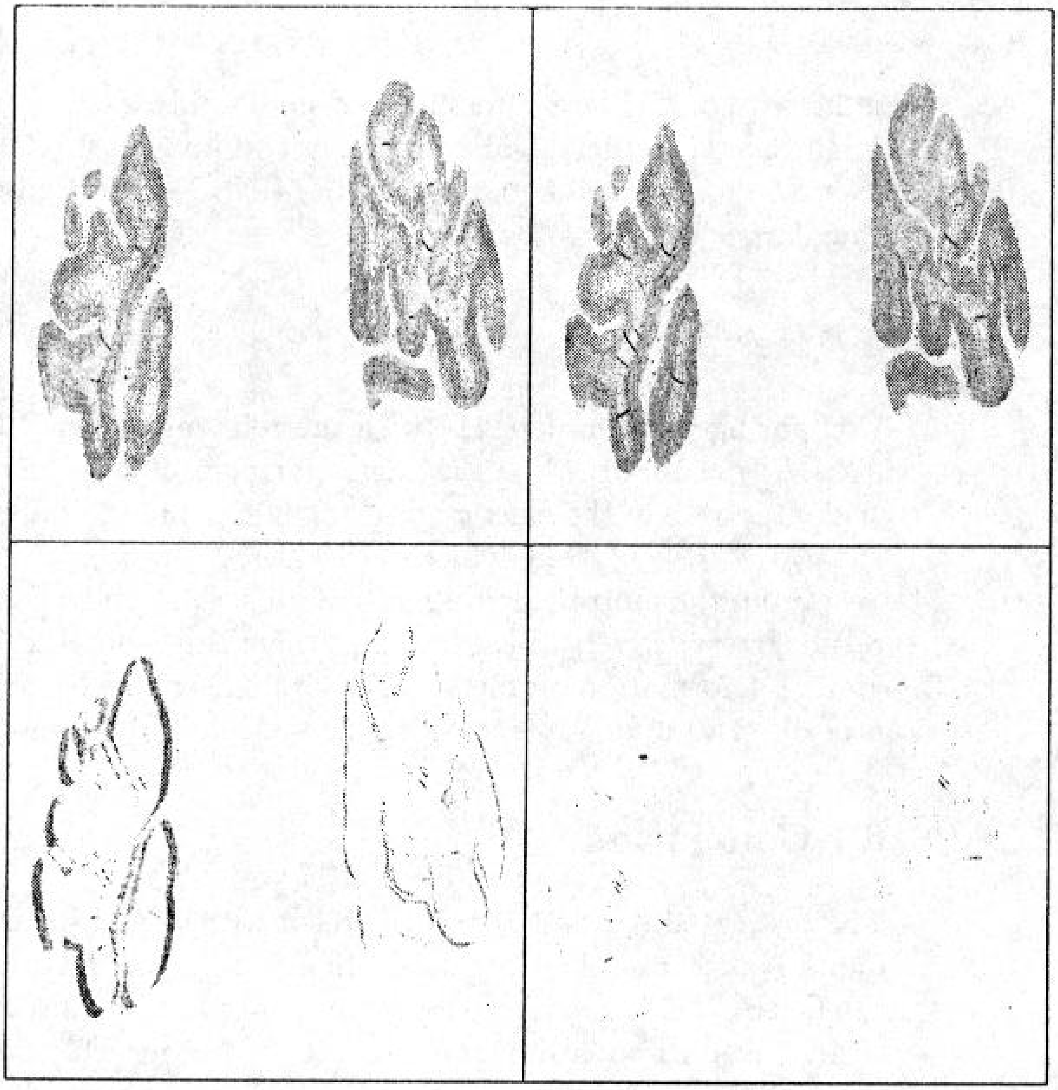
The parallel implementation makes the approach applicable and attractive for medical image processing applications, especially those from the human neuroscanning project. The parallel implementation allows the matching of images resulting from high resolution sections. Thus, linear artifacts arising from downscaling and bilinear interpolation can be reduced using large sized images in the discrete non-linear elasticity model.

¿From the medical point of view, further work has to be performed by investigating and interpreting the results. Up to now we do not know what is actually the best match in medical sense, what is undermatched and which result is overmatched (corrected template and reference are identical). ¿From the mathematical point of view a convergence proof of the overall algorithm is under work.

However, the method is a promising tool for the reconstruction process. In the future we will also parallelize the method using a multigrid solver and in particular a direct solver based on fast Fourier-type techniques (FFT) [8]. Therefore, different moduls for solving the system of linear equations with appropriate parallelization strategies have to be supplied.

Our implementation demonstrates that the elastic matching algorithm produces promising results in a reasonable amount of time on a high performance cluster system.

Acknowledgements We are indepted to A. Folkers and J.-M. Frahm for implementing parts of the algortithm.



$||T - R||_2 \approx 10361$ $||T_{200} - R||_2 \approx 3123$

Fig. 3. Upper left: 116 (Reference R), upper right 117 (Template T), (both 512×512 pixels, 256 gray levels, note: differences can hardly be seen by human eyes), lower left: $|T - R|$, lower right: $|T_{500} - R|$. Here, only small differences in comparison to the first subtraction image remained, reduction: $||T_{500} - R|| / ||T - R|| \approx 30.1\%$.

References

1. Amit, Y., Grenander, U., Piccioni, M.: Structural Image Restoration Through Deformable Templates, Journal of the American Statistical Association, 86(414) (1991) 376–387
2. Bajcsy, R., Kovačič, S.: Toward an Individualized Brain Atlas Elastic Matching, MS-CIS-86-71 Grasp Lap 76, Dept. of Computer and Information Science, Moore School, University of Philadelphia (1986)
3. Bookstein, F.L.: A statistical Method for Biological Shape Comparisons, J. theor. Biol. 107, (1984) 475–520
4. Bookstein, F.L.: Size and Shape Spaces for Landmark Data in Two Dimensions, Stat. Sci. 1, (1986) 181–242
5. Boden, N.J., Cohen, D., Felderman, R.E., Kulawik, A.E., Seitz, C.L., Seizovic, J.N., Su, W.-K.: Myrinet: A Gigabit-per-Second Local-Area Network. IEEE Micro 15(1) (1995) 119–128

6. Bro-Nielsen, M.: Medical Image Registration and Surgery Simulation. PhD thesis, IMM, Technical University of Denmark (1996)
7. Christensen, G.E.: Deformable Shape Models for Anatomy. PhD thesis, Sever Institute of Technology, Washington University (1994)
8. Fischer, B., Modersitzki, J.: Fast Inversion of Matrices Arising in Image Processing. Tech. Report A-98-29, Medical University of Lübeck (1998)
9. Golub, G.H., van Loan, C.F.: Matrix Computations. The Johns Hopkins University Press, Baltimore, Second edition (1989)
10. Gurtin, M.E.: An Introduction to Continuum Mechanics. Academic Press, Orlando (1981)
11. Obelöer, W., Maehle, E.: Integration of Debugging and Performance Optimization. Proc. 2nd Sino-German Workshop Advanced Parallel Processing Technologies - APPT, 17–24, Koblenz (1997)
12. Geist, A, et al.: PVM 3 Users Guide and Reference Manual. Technical Report ORNL/TM-12187, Oak Ridge National Laboratory (1993)
13. Medical University of Lübeck, Institute of Computer Engineering, Störtebeker Cluster Project. http://www.iti.mu-luebeck.de/cluster (1998)
14. Schormann, T., Henn, S., Zilles, K.: A New Approach to Fast Elastic Alignment with Applications to Human Brains. LNCS Vol. 1131 (1996) 337–342
15. Schormann, T., Dabringhaus, A., and Zilles, K.: Statistics of Deformations in Histology and Application to Improved Alignment with MRI, IEEE Transactions on Medical Imaging, 14(1) (1995) 25–35
16. Schormann, T., von Matthey, M., Dabringhaus, A., Zilles, K.: Alignment of 3-D Brain Data Sets Originating from MR and Histology, Bioimaging, Vol. 1 (1993) 119–128
17. Thirion, J.-P.: Non-Rigid Matching Using Demons, *in IEEE, Correspondence on Computer Vision and Pattern Recognition*, 1996, 245–251

Volume Editors

Peter Sloot
Alfons Hoekstra
Bob Hertzberger
University of Amsterdam
Faculty of Mathematics, Computer Science, Physics, and Astronomy
Kruislaan 403, 1098 SJ Amsterdam, The Netherlands
E-mail: {sloot, alfons, bob}@wins.uva.nl

Marian Bubak
University of Mining and Metallurgy (AGH)
Institute of Computer Science and Academic Computer Center CYFRONET
al. Mickiewicza 30, 30-059 Cracow, Poland
E-mail: bubak@uci.agh.edu.pl

Cataloging-in-Publication data applied for

Die Deutsche Bibliothek - CIP-Einheitsaufnahme

High performance computing and networking : 7th international conference / HPCN Europe 1999, Amsterdam, The Netherlands, April 12 - 14, 1999. Peter Sloot ... (ed.). - Berlin ; Heidelberg ; New York ; Barcelona ; Hong Kong ; London ; Milan ; Paris ; Singapore ; Tokyo : Springer, 1999
(Lecture notes in computer science ; Vol. 1593)
ISBN 3-540-65821-1

CR Subject Classification (1998): C.2.4, D.1-2, E.4, F.2, G.1-2, J.1-2, J.3, J.6, K.6

ISSN 0302-9743
ISBN 3-540-65821-1 Springer-Verlag Berlin Heidelberg New York

Printed in Germany

Typesetting: Camera-ready by author
SPIN: 10704672 06/3142 – 5 4 3 2 1 0 Printed on acid-free paper