Individualizing Neuro-anatomical Atlases Using a Massively Parallel Computer

Gary E. Christensen
Michael I. Miller
Michael W. Vannier
Washington University

Ulf Grenander
Brown University

Generalized electronic atlases of the head, created from MRI scans and labeled by experts, are currently available. The algorithms in this research create individualized, subject-specific atlases.

Modern medical imaging modalities such as computerized tomography (CT), magnetic resonance imaging (MRI), and positron emission tomography (PET) provide detailed 3D pictures of anatomy and function. Unfortunately, this high-dimensional voxel-based data must be interpreted manually, and quantitative comparison between individuals or across populations is exceedingly difficult without extensive manual input or gross simplification. This article presents methods that overcome these problems by transforming a generic digital neuroanatomical atlas to closely correspond with a specific individual's data set, thus creating an individualized atlas.

An anatomical atlas is an annotated volume of images, charts, or tables that systematically illustrate an anatomical part. Atlas annotations often include structure names, descriptions, locations, and functions, as well as other information specific to the atlas anatomy. Individualized digital atlases can be generated by using a computer to transform the shape of the atlas into the shape of images taken of the individual. As shown in Figure 1, the atlas transformation describes the correspondence between the coordinate system of the individual's data set and that of the atlas at each point. The atlas transformations are constrained to preserve the atlas neighborhood structure and continuity under the transformation.

Once the deformable atlas is individualized, it can be used to analyze data sets by querying for regions of interest, structure volumes, locations and magnitudes of abnormalities, and so on. The inverse transformation can also be used to map data from the individual's coordinate system back to the atlas' standard coordinate system. Transforming data sets to the shape of the atlas anatomy removes individual anatomical shape variations, which is useful in comparing brain function between individuals.

Historically, registration methods have used affine transformations that account for global translation, rotation, scale, and skew, or nonrigid transformations obtained by matching a small number of landmarks. These approaches account for global shape variations only and ignore small local variations. Our work extends these methods with a new approach that accommodates both global and local shape variation.

Potential applications of these techniques include locating areas in the brain where different functions take place, characterizing shape variations in diseases such as schizophrenia and Alzheimer's, and providing patient-specific data for planning and rehearsing surgical procedures.

PATTERN THEORY

Grenander's shape models[1] represent anatomical structure and variability in a mathematically rigorous framework that evolved from trying to represent complex knowledge.[2] Normal anatomy is represented by constructing an anatomical atlas through exhaustive segmentation and labeling of a single volumetric study. Höhne's Voxel-Man[3] is an example of an electronic atlas. Variability of shape between individuals is accommodated by defining probabilistic transformations on the atlas coordinate system. By applying high-dimensional local transformations to the atlas, a rich family of anatomies can be represented using a single atlas (see Figure 2).

Transformations on the atlas must maintain the topology—that is, the integrity of anatomic structures—and have high spatial dimension to accommodate complex anatomical details. This is achieved by constraining the set of transformations to be consistent with physical deformations of real materials such as elastic solids[4,5] and viscous fluids.[6,7]

Mathematical definitions

A deformable atlas or template is a multivalued function $\mathbf{T} = \{T_n\}_{n=1}^{N}$ defined on the ideal coordinate system $\Omega \subset R^3$. For convenience, we will assume that $\Omega = [0, 1]^3$ = {the unit cube}. The template function $\mathbf{T}$ describes how the coordinate system or tissue of the atlas anatomy looks when imaged by different imaging modalities such as MRI, CT, PET, cryosection, and segmentation. That is, $\mathbf{T}$ describes the value of the atlas anatomy for each modality in the atlas at each location in the coordinate system. For example, the value of the atlas at coordinate (0.5, 0.5, 0.5) might be expressed as $\mathbf{T}(0.5, 0.5, 0.5)$ = (150, 1000, 30, (120, 135, 45), "*ventricle*"), meaning that this coordinate has intensity 150 in an MRI image, 1,000 in a CT image, 30 in a PET image, (red = 120, green = 135, blue = 45) in an RGB cryosection image, and the label "ventricle" in a segmentation.

The *target* or *individual* is characterized via a study $\mathbf{S} = \{S_n\}_{n=1}^{M}$ consisting of M characterizing data sets, or substudies. Each substudy is an examination of the target brain tissue via the sensing modality. The information in the atlas is brought into the patient's coordinates by finding the transformation registering the studies $\{S_n\}_{n=1}^{M}$ with the corresponding M modalities from the template $\{T_n\}_{n=1}^{N}$, where $M<N$.

A new atlas $\mathbf{T}_{\text{NEW}}$ is generated from the original atlas $\mathbf{T}$ by applying a transformation $\mathbf{h}(\mathbf{x}) = \mathbf{x} - \mathbf{u}(\mathbf{x})$ to the coordinate system of $\mathbf{T}$, that is, $\mathbf{T}_{\text{NEW}}(\mathbf{x}) = \mathbf{T}(\mathbf{x} - \mathbf{u}(\mathbf{x}))$. The function $\mathbf{u}$ is called the displacement field and describes the transformation in terms of a displacement at each point in the coordinate system. Registration between the atlas and study is accomplished by minimizing distance measures $D(\mathbf{u})$ such as the Gaussian sensor model

$$D(\mathbf{u}) = \frac{\gamma}{2}\int_{\Omega} \left|T(\mathbf{x}-\mathbf{u}(\mathbf{x}))-S(\mathbf{x})\right|^2 d\mathbf{x} \qquad (1)$$

where γ is a constant. (For notational convenience in this equation, T and S are assumed to be singular valued corresponding to the same sensor modality.)

The transformation from the template to the target must be smooth so that connected sets remain connected, surfaces are mapped as surfaces, and the global relationships between structures are maintained. Such properties correspond to topological properties of the transformation: continuity, differentiability, positive-definiteness of the Jacobian, and others. To ensure smoothness, the transformation $\mathbf{x} \to \mathbf{x} - \mathbf{u}(\mathbf{x})$ is estimated, which minimizes the distance measure $D(\mathbf{u})$ while satisfying smoothness constraints. These smoothness constraints are enforced on the transformation by employing a potential function $E(\mathbf{u})$ that assigns a small penalty to smooth transformations and a large penalty to nonsmooth transformations. In our work, we use potential functions based on the kinematics

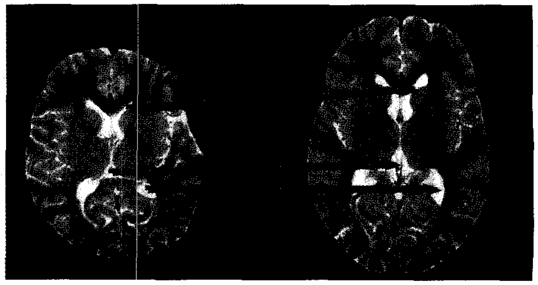

Figure 1. Two transverse magnetic resonance imaging (MRI) scans showing relatively the same anatomical structures in two subjects. The arrows show points of correspondence between the two images. Algorithms for finding high-dimensional transformations are parameterized by an arrow of correspondence at every pixel/voxel location.

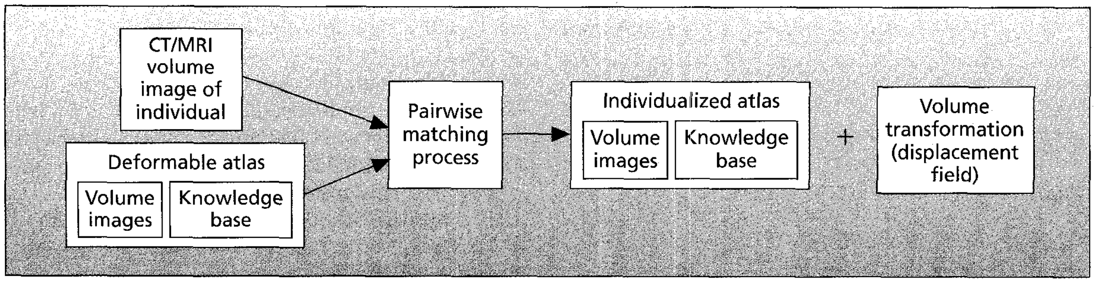
CT/MRI volume image of individual
Deformable atlas
Volume images
Knowledge base
Pairwise matching process
Individualized atlas
Volume images
Knowledge base
+
Volume transformation (displacement field)

Figure 2. An individualized neuroanatomical atlas, one that closely corresponds to an individual's specific data set, is generated by transforming the deformable atlas coordinate system until the information in the atlas corresponds to the subject's anatomy. This is accomplished automatically by pairwise matching image volumes from the patient and the atlas. Shape differences between the atlas and subject's anatomy are contained in the volume transformation.

of continuum mechanics.[8] For example, $E(\mathbf{u})$ has the form

$$E(\mathbf{u}) = \frac{1}{2}\sum_{i=1}^{3}\sum_{j=1}^{3}\int_{\Omega} \lambda\left(\frac{\partial u_i(\mathbf{x})}{\partial x_i}\right)\left(\frac{\partial u_j(\mathbf{x})}{\partial x_j}\right) + u\left(\frac{\partial u_i(\mathbf{x})}{\partial x_j} + \frac{\partial u_j(\mathbf{x})}{\partial x_i}\right)^2 d\mathbf{x} \tag{2}$$

where λ and μ are the Lamé elasticity constants for a linear elastic model.[4,5]

The estimation problem becomes that of inferring the transformation $\mathbf{x} \rightarrow \mathbf{x} - \mathbf{u}(\mathbf{x})$ of the template that minimizes both the distance measure $D(\mathbf{u})$ and the potential energy $E(\mathbf{u})$, that is, finding

$$\hat{\mathbf{u}} = \arg\min_{\mathbf{u}}(D(\mathbf{u}) + E(\mathbf{u})) \tag{3}$$

The solution $\hat{\mathbf{u}}$ corresponds to the variational minimizer of $D(\mathbf{u}) + E(\mathbf{u})$ with respect to $\mathbf{u}$.

Equation 3 can be interpreted probabilistically by assuming that $D(\mathbf{u})$ and $E(\mathbf{u})$ are potential energy functions of Gibbs probability measures, that is, of the type $p(\mathbf{u}) = (1/z)e^{-E(\mathbf{u})}$ where z is a normalizing constant. $D(\mathbf{u})$ corresponds to the likelihood because it describes the imaging sensor characteristics and $E(\mathbf{u})$ corresponds to the prior because it describes a priori knowledge about the smoothness of the transformation. The introduction of the prior distribution places the minimization problem in the class of Bayesian inference problems.[9] With this interpretation, $\hat{\mathbf{u}}$ is called the Bayesian maximum a posteriori (MAP) estimate.

Deformation models

For linear elastic solids, the restoring force holding the template together grows proportionately with the displacement from the original configuration of the template. The solution of Equation 3 for a linear elastic solid satisfies the partial differential equation (PDE)

$$\mu \nabla^2 \mathbf{u}(\mathbf{x}) + (\lambda+\mu)\, \nabla\, (\nabla \cdot \mathbf{u}(\mathbf{x})) = \mathbf{b}\,(\mathbf{x} - \mathbf{u}(\mathbf{x})) \tag{4}$$

with certain boundary conditions such as $\mathbf{u}(\mathbf{x}) = 0$ for $\mathbf{x}$ on the boundary of Ω (see Christensen, Rabbitt, and Miller[5] for other types of boundary conditions).

In Equation 4, the divergence and Laplacian operators are

$$\nabla = \left[\frac{\partial}{\partial x_1}, \frac{\partial}{\partial x_2}, \frac{\partial}{\partial x_3}\right]^T$$

and

$$\nabla^2 = \frac{\partial^2}{\partial x_1} + \frac{\partial^2}{\partial x_2} + \frac{\partial^2}{\partial x_3}$$

and μ, λ are the Lamé constants . The left side of Equation 4 corresponds to the variational minimizer with respect to $\mathbf{u}(\mathbf{x})$ of the potential $E(\mathbf{u})$ for a linear elastic solid. The body force $\mathbf{b}(\mathbf{x} - \mathbf{u}(\mathbf{x}))$ is the driving function that deforms the template into the shape of the target data set, and it corresponds to the variational minimizer with respect to $\mathbf{u}(\mathbf{x})$ of the distance function. For the Gaussian distance measure given in Equation 1, the body force is

$$\mathbf{b}(\mathbf{x} - \mathbf{u}(\mathbf{x})) = -\gamma\,(T(\mathbf{x} - \mathbf{u}(\mathbf{x})) - S(\mathbf{x})) \cdot \nabla T|_{\mathbf{x}-\mathbf{u}(\mathbf{x})} \tag{5}$$

For viscous fluids, the force holding the template together grows proportionately to the rate of change of the transformation of the template. The PDE describing the fluid transformation of the atlas[6] is given by

$$\alpha\, \nabla^2 \mathbf{v}(\mathbf{x}, t) + (\alpha + \beta)\, \nabla\, (\nabla \cdot \mathbf{v}(\mathbf{x}, t)) = \mathbf{b}(\mathbf{x} - \mathbf{u}(\mathbf{x}, t)) \tag{6}$$

where

$$\mathbf{v}(\mathbf{x},t) = \frac{d\mathbf{u}(\mathrm{x}, t)}{dt} = \frac{\partial \mathbf{u}(\mathrm{x},t)}{\partial t} + \sum_{i=1}^{3} v_i(\mathrm{x}, t)\, \frac{\partial \mathbf{u}(\mathrm{x}, t)}{\partial x_i} \tag{7}$$

is the velocity of the transformation and α and β are viscosity constants. Note that the fluid PDE in Equation 6 was derived directly (see Christensen, Rabbit, and Miller[6]) and does not have an associated potential formulation. For the work herein, we assume that $\mathbf{v}(\mathbf{x}, t) = 0$ on the boundary of Ω. This PDE is almost identical in form to Equation 4, except that the displacement field $\mathbf{u}$ is replaced by the velocity field $\mathbf{v}$. However, these equations are very different, as can seen by substituting Equation 7 into Equation 6. The nonlinear relationship between $\mathbf{v}$ and $\mathbf{u}$ expressed by Equation 7 allows the fluid model to track long-distance, nonlinear deformations of small subregions.

Convergence of the fluid transformation algorithm occurs as time goes to infinity. This is because the driving function or body force (Equation 5) goes to zero as the template deforms into the study. As the body force goes to zero, so does the velocity field (Equation 6). As the velocity field goes to zero, so does the change in the displacement field (Equation 7), giving convergence. Practically, the algorithm is stopped after a fixed number of iterations that give the deformed template and the target image a similar appearance. The algorithm can also be stopped when the absolute difference in intensity of the deformed template and target images falls below a certain value.

3D transformation example

Figures 3-5 show an individualized deformable neuroanatomical atlas, in which one 128 × 128 × 100-voxel MRI volume was fluidly transformed into the shape of another. The study data set was preprocessed so that its histogram matched that of the atlas MRI data set. Figure 3 shows the front and side surface renderings of the atlas (left), the study (middle), and the fluidly deformed atlas (right). The goal is to deform the coordinate system of the atlas so that the anatomy of the atlas corresponds to that of the study. Notice that the deformed atlas and study appear nearly identical.

We used a multiresolution approach to estimate the transformation. This increases the convergence rate and avoids local minimum corresponding to incorrect matches. Figure 4 shows a sequence of 3D surface-rendered MRI images of the template as it deformed into the study. The initial or global transformation of the atlas was estimated assuming that the template was a deformable cube constructed of a linear elastic material.[5] The linear elastic transformation was constrained to give a global nonrigid transformation by allowing deformations of only the 192 lowest frequency modes of vibration. The modes of vibration correspond to the eigenfunctions $\Phi_i(\mathbf{x})$ that satisfy $\nabla^2 \Phi_i(\mathbf{x}) + \nabla(\nabla \cdot \Phi_i(\mathbf{x})) = \kappa_i \Phi_i(\mathbf{x})$, where κ_i are the eigenvalues corresponding to $\Phi_i(\mathbf{x})$. (See Equation 6 of Christensen, Rabbitt, and Miller.[5])

Local deformation of the atlas was performed by assuming that the elastically deformed template could deform as a cube of compressible fluid material as described by Equations 6 and 7. The fluid model allowed the template to fully deform into the shape of the study without being "held back," as was the case with the linear elastic model. The fluid transformation was parameterized by one translation vector and one velocity vector for each voxel, giving a total of $6 \times 128 \times 128 \times 100 \approx 9.8 \times 10^6$ parameters. This high-dimensional parameterization allows enough degrees of freedom to accommodate the local anatomical shape variation. The normalized squared difference was 56,000 initially, 20,000 after the elastic transformation, and 1,800 after the fluid transformation. (The normalized squared difference is computed by normalizing the voxel intensities of the template and study between 0 and 1, subtracting the deformed template from the study, and summing the squares of the voxel differences.)

The top and bottom rows of Figure 5 show sagittal MRI slices 74 and 62 from the atlas (left), study (middle), and fluidly deformed atlas (right). The differences between the atlas and study for these slices illustrate the need to accommodate anatomical shape differences in all three coordinate directions. For example, notice how different the shapes of the constituent structures of the study (top middle) are from those in the atlas (top left) and how similar they are to those in the deformed atlas (top right). Likewise, the bottom row of Figure 5 shows close correspondence between the anatomical features after deformation.

The atlas deformation algorithms are computationally intensive due to the large number of parameters estimated and the iterative nature of the deformation algorithms. Therefore, estimating the transformation that individualizes the deformable atlas requires a supercomputer.

MASSIVELY PARALLEL IMPLEMENTATION

We implemented the elastic and fluid deformation algorithms on a MasPar 128 × 128 massively parallel* SIMD computer (Model MP2).[10] This specialized hardware architecture let us compute the complex atlas transformations in realistic time frames, that is, a few minutes or hours instead of a few days or weeks.

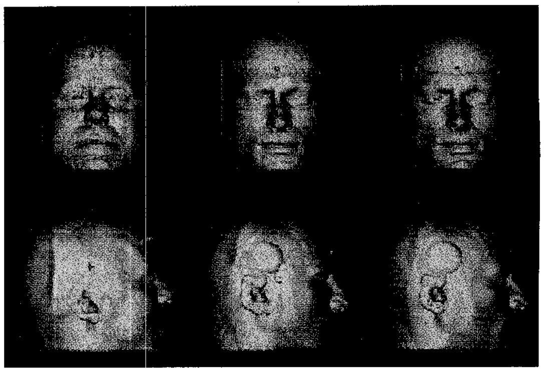

Figure 3. Front and side 3D views of facial surfaces reconstructed from MRI data of the deformable atlas anatomy (left), the study (middle), and fluidly deformed atlas (right). Note that a small part of the ear and scalp were excluded from the sample volume, which resulted in the rendering artifact on the side of the head in the study and deformed atlas.

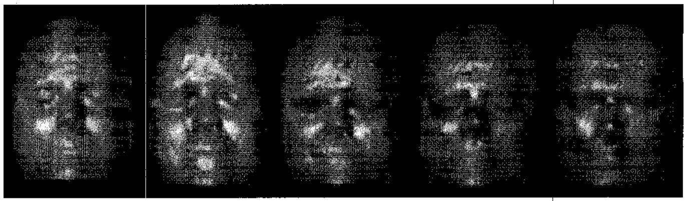

Figure 4. Facial-surface 3D views reconstructed from MRI data of the deformable atlas anatomy as it is automatically deformed into the shape of the study anatomy. The undeformed atlas is shown in the leftmost image, and the final deformed state is shown at far right.

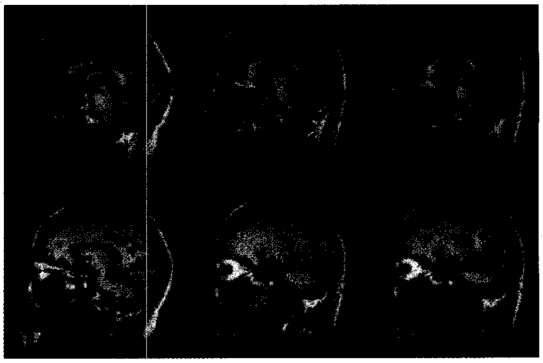

Figure 5. The top and bottom rows show sagittal MRI slices 74 and 62 from the MRI atlas (left), study (middle), and fluidly deformed atlas (right).

*A massively parallel computer has more than 1,000 processor elements (PEs) and is classified as having either a SIMD or a MIMD parallel architecture. SIMD (single instruction, multiple data) systems are used for problems that require a large number of computations in which all the PEs perform the same operation in parallel. On each clock cycle, each PE performs the same operation on its private data. MIMD (multiple instruction, multiple data) systems are used for algorithms that can be broken up into separate, independent parts to solve. Each part is assigned to a separate processor and all the parts are solved simultaneously. In general, the PEs used in SIMD machines are much less complicated to design than those used in MIMD machines and therefore require less area on a computer chip for layout. As a result, SIMD computers generally have more—but simpler—processors than MIMD computers for the same price. Due to the simplicity of the PE design, operations take more clock cycles to compute on a SIMD computer than a MIMD computer. However, the speed advantage of the SIMD computer comes from a large number of processors computing in parallel.

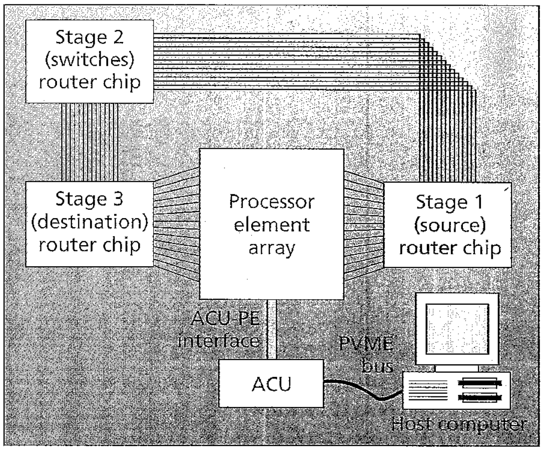

Figure 6. MasPar system block diagram. The array control unit (ACU) executes the same instruction on each processor element (PE) in the PE array.

The MasPar computer's 16,384 processors permit breaking the algorithm into simple commands that can be executed in parallel. The PEs in the MasPar are connected in a 128 × 128 toroidal mesh for efficient local communication and by a global router for nonlocal communication (see Figures 6 and 7). The PE mesh connection, called the xnet, allows each PE to send/ receive data (communicate) to/from its eight nearest neighboring PEs—a valuable feature for solving problems such as partial differential equations that require local PE communication. The global router allows PEs to send/receive data to/from other PEs that are separated by long distances. It also allows the PEs to communicate in nonregular patterns in parallel, which is important for computing the deformed template.

Solving partial differential equations

Partial differential equations (PDEs) are central for individualizing the atlas because they constrain the transformations to be smooth—that is, they ensure atlas structure connections and preserve the atlas neighborhood structure under the transformation.[7] It turns out that the PDEs in Equations 4 and 6 must be solved at each iteration of the deformation algorithm.[4-6]

Mesh-connected, massively parallel computers exploit the local, regular PE communication required to solve PDEs. For example, consider the 2D partial differential equation

$$\frac{\partial^2 u(x,y)}{\partial x^2} + \frac{\partial^2 u(x,y)}{\partial y^2} = f(x,y) \tag{8}$$

with constant forcing function $f(x, y)$ defined on the unit square $\Omega = [0, 1]^2$. Rewriting this equation using symmetric difference formulas[11] for the second derivatives gives

$$\frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{\Delta^2} + \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{\Delta^2} = f_{i,j} \tag{9}$$

where $u_{i,j} = u(x_i, y_j)$, $f_{i,j} = f(x_i, y_j)$, $x_i = i\Delta$, $y_j = j\Delta$, and $\Delta = 1/N$. Solving for $u_{i,j}$ gives the desired result

$$u_{i,j} = \frac{1}{4}(u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} - \Delta^2 f_{i,j}) \tag{10}$$

Equation 8 is solved numerically for u, using Equation 10, by repeated calculation of $u_{i,j}$ at each location (i,j) until the values $u_{i,j}$ converge. This is done by mapping each pair $u_{i,j}$ and $f_{i,j}$ to a separate PE at location (i, j) in the mesh. Notice that the data required to compute Equation 10 at each PE is either stored locally ($f_{i,j}$) or in a neighboring PE ($u_{i+1,j}, u_{i-1,j}, u_{i,j+1}, u_{i,j-1}$). The data from neighboring PEs is fetched by all PEs in parallel by shifting the u array north, south, east, and west.

Transforming the template

Besides calculating PDEs in parallel, the MasPar is well

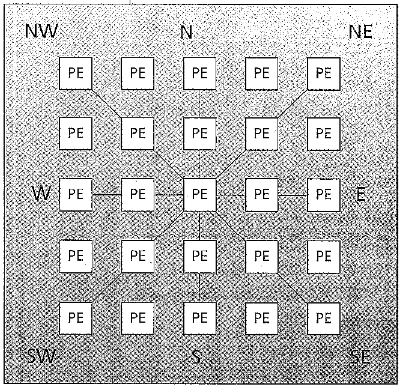

Figure 7. The MasPar is well suited for solving partial differential equations because each processor can send/receive data to/from its eight nearest neighbors in parallel.

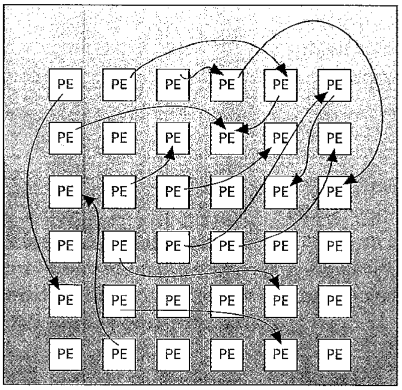

Figure 8. Computing the template transformation requires a nonregular pattern of PE communication. The MasPar's global router transfers data from source to destination PEs as indicated by the parallel arrows.

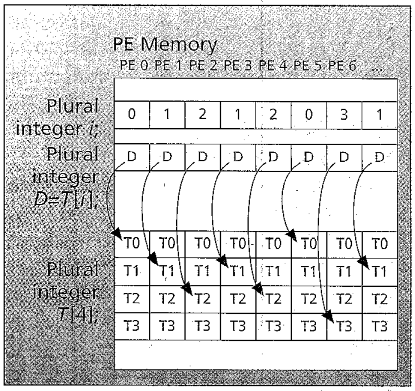

Figure 9. Parallel indirect addressing. The value of *D* at each PE is set equal to the value of *T* offset by the index *i* in parallel.

Table 1. Typical program execution times on the 128 x 128 MasPar for 2D and 3D transformations. The linear elastic transformations were constrained by the number of low-frequency fundamental modes of vibration equal to the number of parameters estimated.

Algorithm	Data size	Parameters estimated	Iterations	PDE iterations	Total time
2D elastic	64 x 64	50	160	—	25 sec
2D fluid	256 x 256	2.6 x 10^5	150	250	110 sec
3D elastic	64 x 64 x 50	192	100	—	7.5 min
3D fluid	128 x 128 x 100	9.8 x 10^6	250	100	1.8 hours

Table 2. Typical 3D fluid execution time for a 128 x 128 x 100 voxel data set, 100 fluid PDE iterations, and 250 time steps. The three middle columns correspond to the calculation of Equations 5, 6, and 7, used to generate the fluid transformation. Execution times for the MIPS 150-MHz R4400 and 75-MHz R8000 sequential processors were generated using a Silicon Graphics Indigo^2 and a Power Challenge computer, respectively, and were extrapolated from 32 x 32 x 25 data sets.

Architecture	Body-force calculation	Fluid PDE calculation	Update displacement field	Total time
150-MHz MIPS R4400	20 hours	6.4 days	4.2 hours	7.4 days
75-MHz MIPS R8000	6.4 hours	1.1 days	1.0 hours	1.4 days
128 x 128 MasPar (MP2)	1.3 hours	28 min	24 sec	1.8 hours

suited for transforming the template in parallel. The evaluation of the template and its derivatives in the transformed coordinate system is computed on each iteration of the deformation algorithms. The problem of transforming the template differs from that of solving PDEs because it requires a nonregular pattern of communication between the PEs (see Figure 8). Long-distance communication is necessary because of large template deformations; the nonregular communication pattern is due to local template deformations.

Normally, a nonregular pattern of communication (random access) is very inefficient on a mesh-connected computer because it is not a parallel operation using nearest neighbor data transfer. Two MasPar features—the global router (Figure 8) and local PE indirect memory addressing (Figure 9)—reduce this problem. Together, they provide random access to the required data. First, each PE calculates the memory address of the data that it needs to compute its value of the deformed template. This address is converted into a target PE address and its local memory address. Using the target PE addresses, the global router sends the local memory addresses from the requesting PEs to the target PEs in parallel. Next, all target PEs use indirect addressing to fetch the requested data from their local memory in parallel. Finally, the target PEs use the global router to return the data to the requesting PEs in parallel. Without the global router and PE indirect addressing, the PEs would be very inefficient at the random access required to transform the template.

The PEs on the MasPar are divided into 4 × 4 subgroupings called clusters. Only one PE per cluster can send data, and only one PE can receive data during a router data transfer. Therefore, address contention occurs when two or more PEs request to send or receive from the same cluster. To handle this contention, the router hardware uses several router data transfers until all data is transferred. This reduces parallelism but is faster than nearest neighbor PE communication for long-distance, random-address data transfers.

Timing information

Table 1 gives typical program execution times for the linear elastic[4,5] and fluid[6] algorithms. These entries reflect typical data sizes and parameters used for 2D and 3D problems. In theory, the 2D and 3D linear elastic transformation times could be four times faster, because they were computed using only a 64 × 64-PE array, one fourth of the MasPar's PEs. The column labeled "PDE iterations" corresponds to the number of iterations used to solve the fluid PDE (Equation 6) at each iteration of the fluid algorithm. Notice that the execution times of the 2D and 3D algorithms are not linear with respect to the third dimension. This is due to the increased number of PDE cross-derivatives in 3D as opposed to 2D (see Equations 4 and 6) and to the increased computations required for trilinear interpolation of the template and its derivatives over bilinear interpolation.

We compared execution times of the 3D fluid algorithm on MasPar's 128 × 128 MP2 and on MIPS Technologies' 150-MHz R4400 and 75-MHz R8000 sequential processors (see Table 2). The execution time of the fluid PDE procedure scales linearly with the number of fluid PDE iterations, and the total execution time scales linearly with the number of iterations and the dimensions of the data set. The largest portion of the execution time for the R4400 and R8000 is the calculation of the fluid PDE, Equation 6, which is 330 and 57 times slower, respectively, than the same calculation on the MasPar. The MasPar's speed advantage is expected there, since it calculates the fluid PDE in parallel, but its speed advantage is much less for the body-force calculation. That's because the body-force calculation requires random data access to evaluate the deformed template and its derivatives. Sequential processors are designed for random memory access and therefore are much better suited for calculating the body force than the global router and parallel indirect addressing of the MasPar.

SYNTHESIZING INDIVIDUALIZED ATLASES that accurately accommodate local anatomical shape variability has many exciting potential applications. For example, these techniques can be used to help locate areas in the brain where different functions take place. This would be accomplished by transforming functional data such as PET and functional MRI collected from different individuals to the atlas brain. Once registered with the atlas, the data can be interpreted as if all experiments were performed on the same individual. Neuromorphometrics, the study of the shape of neuroanatomical structures, is another important application. Segmentations of anatomical structures are used to characterize normal and abnormal anatomical shapes and their shape variation in diseases such schizophrenia, Alzheimer's, Huntington's, and Parkinson's disease. Finally, individualized atlases will improve patient care by giving the physician more accurate information to diagnose and correct problems. Abnormalities will be flagged in the individualized atlas for further review along with a list of possible causes and possible treatments. Individualized atlases will also let physicians plan and practice surgical operations using the patient's own anatomy on the computer before actually making the first cut in the operating room. **I**

Acknowledgments

This research was supported in part by the Craniofacial Imaging Laboratory, St. Louis Children's Hospital, Washington University Medical Center, NIH-NCRR-RR01380, ARO-DAAL03-92-G-0015, ARPA-MDA972-93-1-0012, the Reconstruction and Representation of Cerebral Cortex grant 5-R01-MH/DA52158, and a grant from Digital Equipment Corp.

References

1. U. Grenander, *General Pattern Theory*, Oxford University Press, 1993.
2. U. Grenander and M.I. Miller, "Representations of Knowledge in Complex Systems," *J. Royal Statistical Soc.*, Vol. 56, No. 4, 1992, pp. 72-78.
3. K.H. Höhne et al., "A 3D Anatomical Atlas Based on a Volume Model," *IEEE Computer Graphics and Applications*, Vol. 12, No. 4, July 1992, pp. 72-78.
4. M.I. Miller et al., "Mathematical Textbook of Deformable Neuroanatomies," *Proc. Nat'l Academy of Science*, Vol. 90, No. 24, Dec. 1993, pp. 11,944-11,948.
5. G.E. Christensen, R.D. Rabbitt, and M.I. Miller, "3D Brain Mapping Using a Deformable Neuroanatomy," *Physics in Medicine and Biology*, Vol. 39, 1994, pp. 609-618.
6. G.E. Christensen, R.D. Rabbitt, and M.I. Miller, "Deformable Templates Using Large Deformation Kinematics," accepted for 1996 publication in *IEEE Trans. Image Processing*.
7. G.E. Christensen et al., "Topological Properties of Smooth Anatomic Maps," in *Information Processing in Medical Imaging*, Y. Bizais, C. Brailot, and R. Di Paloa, eds., Kluwer Academic Publishers, Boston, 1987, Vol. 3, pp. 101-112.
8. L.A. Segel, *Mathematics Applied to Continuum Mechanics*, Dover Publications, New York, 1987.
9. H.L. Van Trees, *Detection, Estimation and Modulation Theory*, Part 1, John Wiley and Sons, New York, 1968.
10. J.L. Potter, ed., *Massively Parallel Processor*, Scientific Computation Series, MIT Press, Cambridge, Mass., 1985.
11. J.C. Strikwerda, *Finite Difference Schemes and Partial Differential Equations*, Wadsworth and Brooks/Cole, Pacific Grove, Calif., 1989.

Gary E. Christensen *is a research assistant professor in the Department of Surgery at Washington University School of Medicine. He holds a joint appointment as a research assistant professor in the Mallinckrodt Institute of Radiology and is the director of the Craniofacial Imaging Laboratory, St. Louis Children's Hospital, Washington University Medical Center. He received BS degrees in electrical engineering and computer science in 1988, graduating magna cum laude, and MS and DSc degrees in electrical engineering in 1989 and 1994, all from Washington University, St. Louis.*

Michael I. Miller *joined Johns Hopkins University's EE department in 1983. He is a professor of electrical engineering and holds the Newton R. and Sarah L. Wilson Professorship in Biomedical Engineering. He is also a full professor in the Mallinckrodt Institute of Radiology and the Institute for Biomedical Computing and director of the Army Center for Imaging Science, a consortium of universities including Washington, Harvard, MIT, and Texas. He received a BSEE degree from SUNY, Stony Brook, in 1976 and an MSEE and a PhD in biomedical engineering from Johns Hopkins University in 1978 and 1983.*

Michael W. Vannier *is a professor at the Mallinckrodt Institute of Radiology, Washington University School of Medicine, St. Louis, Missouri. He has authored more than 250 scientific publications and serves as editor-in-chief of the* IEEE Transactions on Medical Imaging. *He graduated from the University of Kentucky School of Medicine and completed a diagnostic radiology residency. He also holds engineering degrees from the University of Kentucky and Colorado State University. In 1994, he was inducted into the US Space Foundation Hall of Fame for work on digital medical imaging.*

Ulf Grenander *has been a professor in the Division of Applied Mathematics at Brown University since 1966 and is the L. Herbert Ballou University Professor Emeritus. His research interests include probability, statistics, and pattern theory. He is the author of several books and has received numerous awards. His recent honors include being named an honorary fellow, Royal Statistical Society, London, 1989; honorary Dr. Sci., University of Chicago, 1994; and member, American Academy of Arts and Sciences, 1995. He received Fil. Lic. and Fil. Dr. degrees from the University of Stockholm in 1948 and 1950.*

Readers can contact Christensen at the Mallinckrodt Institute of Radiology, Washington University School of Medicine, 510 S. Kingshighway, St. Louis, MO 63110. His e-mail address is gec@ee.wustl.edu.