Parallel elastic registration using a multigrid preconditioned Gauss-Newton-Krylov solver, grid continuation and octrees

Rahul S. Sampath (rahulss@cc.gatech.edu)

George Biros (gbiros@cc.gatech.edu)

Computational Science and Engineering Division
Georgia Institute of Technology
Atlanta, GA-30332

Abstract

In this article, we present a parallel algorithm for intensity-based elastic image registration. This algorithm integrates several components: parallel octrees, multigrid preconditioning, a Gauss-Newton-Krylov solver, and grid continuation. We use a non-parametric deformation model based on trilinear finite element shape functions defined on octree meshes. Our C++ based implementation uses the Message Passing Interface (MPI) standard and is built on top of the Dendro and PETSc libraries. We demonstrate the performance of our method on synthetic and medical images. We demonstrate the scalability of our implementation on up to 4096 processors on the Sun Constellation Linux Cluster "Ranger" at the Texas Advanced Computing Center (TACC).

1 Introduction

Given two images, $S(x)$ and $T(x)$, *"registration"* is the process of finding a displacement field, $u(x)$, such that $T(x + u(x))$ is "similar" to $S(x)$ in some metric; $S(x)$ is referred to as the fixed image and $T(x)$ is referred to as the moving image. Image registration has the following applications:

- Integrating complementary information contained in images of the same subject obtained using different modalities.
- Aligning temporal sequences of images to compensate for motion of the subject.
- Image guidance during surgery.
- Aligning images from multiple subjects in statistical studies.
- Comparing images taken at different stages of progression of a disease such as in tumor growth.

Overview and challenges. Given a metric, $\mathcal{D}(\cdot, \cdot)$, for measuring the dissimilarity between any given pair of images, the image registration problem can be viewed as a non-convex optimization problem:

$$\min_u \mathcal{D}(S, T(u)) \tag{1.1}$$

In this work, we use the L^2 norm as the similarity metric[1]:

$$\mathcal{D}(S, T(u)) = \int [S - T(u)]^2 \, dx \tag{1.2}$$

Other similarity metrics such as correlation coefficient, correlation ratio and mutual information have also been used in the literature [15, 48]. The choice of the similarity metric mainly depends on whether or not the same imaging modality[2] was used to acquire both the fixed and moving images. Since we focus on registering images acquired using the same modality, L^2 is appropriate here. The main assumption (and limitation in the SSD metric) here is that similar structures share similar greyscale values.

Minimizing Equation 1.2 is an ill-posed and non-convex problem. So, we need to impose a regularization that constrains the displacement field. The choice of regularization is an important feature that can be used to distinguish between different registration algorithms. In this work, we will use the elastic deformation energy as the regularization and hence this procedure is known as "*elastic registration*". This choice is motivated by the fact that many biological materials that are imaged are elastic. The deformation energy used in this work is based on the linear theory of elasticity. The mathematical statement of the elastic registration problem is given below:

$$\begin{aligned} \min_u \mathcal{J}(u) &= \frac{1}{2} \int [S - T(u)]^2 \, dx + \frac{\gamma}{2} \, a(u, u) \\ a(u, v) &= -\int v \cdot (\Delta u + (\lambda + 1) \nabla div \, u) \, dx \end{aligned} \tag{1.3}$$

$a(u, u)$ represents the elastic potential energy due to the deformation and λ is the Lamé parameter. The regularization parameter, γ, controls the trade-off between the two competing terms in the objective function: (a) dissimilarity between the fixed and deformed moving images and (b) the associated deformation energy. The optimality condition for the regularized problem is given by Equation 1.4:

$$g(u) \cdot v = \gamma \, a(u, v) - \int (S - T(u)) \, \nabla T(u) \cdot v \, dx = 0 \; \forall v \tag{1.4}$$

$g(u)$ represents the gradient of the objective function, $\mathcal{J}(u)$. This is the weak form of a nonlinear coupled partial differential equation (PDE). We need to specify appropriate boundary conditions to solve this equation; we chose to enforce homogeneous Dirichlet boundary conditions. We assume that a rigid registration is performed prior to the elastic registration. This means that there is no relative displacement between the frames for the fixed and moving images for the elastic registration and hence homogeneous Dirichlet boundary conditions are appropriate here.

We use a Gauss-Newton algorithm to solve Equation 1.3. Such an approach requires the repeated numerical solution of large linear systems of equations. To reduce the computation time, we considered a combination of various techniques. To reduce the amount of processed data we needed a non-uniform discretization scheme. Such a discretization scheme would also be useful for further extensions of the work that will include adaptive mesh refinement proceedures. Since we

[1] This is also called the sum of squared differences (SSD) similarity metric.

[2] Modality refers to the technique used to acquire the images. Some popular modalities include ultrasound, magnetic resonance imaging and computed tomography.

are dealing with images, it was natural to use octree-based discretizations instead of generic unstructured discretizations. Due to the sheer size of the problem and storage limitations we needed to consider matrix-free schemes and parallel implementations. We considered iterative solvers since direct solvers do not work with matrix-free schemes and moreover they do not scale well. The convergence rates of most iterative solvers deteriorate with the condition number of the matrix to be inverted. The matrices that need to be inverted to solve the registration problem suffer from ill-conditioning and so it is important to address this problem. Multigrid methods are known to be robust for such ill-conditioned systems but the typical multigrid methods for adaptive discretizations are of the algebraic type, which do not work with matrix-free schemes and have high setup costs rendering them unsuitable for large scale adaptive mesh refinement proceedures. Recently, we developed a parallel, matrix-free, geometric multigrid solver for finite element computations on octree meshes [51, 52] and demonstrated its scalability on over 32000 processors. In this work, we used this algorithm to solve the linear system formed in each Gauss-Newton iteration.

Contributions. The main contributions of this work are summarized below:

- We use parallel octree discretizations to reduce the computation time for the registration problem. This implementation will also allow us to register high resolution images that are too large to fit on a single processor.
- We use a multigrid algorithm to solve the linear system that arises in each optimization iteration. In particular, we use a matrix-free geometric multigrid algorithm, which has lower setup costs compared to its algebraic counterpart. This is significant because we need to setup the linear system of equations for the multigrid scheme in every optimization iteration.

Limitations. There are a few limitations in the proposed framework:

- The linear theory of elasticity is only valid for small deformations. Other regularization approaches may be more appropriate for large deformations.
- We do not incorporate any biophysical information to additionally constrain the deformation. It has been suggested that incorporating such information will provide intelligent priors and reduce the ill-posedness of the registration problem [56].
- We do not use adaptive integration in our present implementation. Instead, we fix the order of the Gauss quadrature rule for integration a-priori. The use of adaptive integration can further reduce the computation costs for evaluating the objective function and gradient.

Related work. Image registration has been an active research area for the past two decades. [25, 67, 45] give a comprehensive review of image registration methods. [32, 41, 42] focus on registration techniques commonly used in medical image processing.

Many of the early works on registration focussed on rigid or affine transformations in which u is restricted to be a rigid or affine displacement field. More complex deformation models used in the literature include polynomial models [23, 24], radial basis functions [21, 22], B-splines [5, 39, 40, 54], Thin-plate splines [10, 49] and elastic splines [16, 65]. These registration methods regularize the problem by discretization (i.e., by parameterizing the displacement field) and so cannot allow fine discretizations (too many parameters). Having few degrees of freedom makes these parametric registration methods computationally efficient but it restricts their ability to handle local deformations well. In contrast, the number of degrees of freedom in non-parametric

registration methods is comparable to the number of pixels in the images. This give these methods the flexibility needed to handle local deformations, but they need to be regularized explicitly by adding a penalty term to the objective function. Different choices for regularization give rise to different registration algorithms: diffusive registration [19], elastic registration [4, 31, 33, 63], fluid registration [12, 64] and curvature registration [20]. There are also examples of hybrid approaches that combine parametric and non-parametric registration methods: [53] used elastic registration on affine registered images to improve the accuracy of reconstruction.

Multigrid methods for solving elliptic PDEs have been researched extensively in the past [6, 11, 17, 30, 66] and remain an active research area [1, 2, 7, 8, 26]. The use of multigrid for image registration is fairly recent and some of the relevant works include [14, 29, 31, 33]. [14] used a diffusive regularizer and a steepest-descent type optimization algorithm accelerated using a multigrid solver. [31] used a Full Approximation Scheme (FAS) and d-linear image approximation for elastic registration. [33] presented a multigrid scheme using operator dependent prolongation for elastic image registration. [29] presented a Gauss-Newton algorithm and a multigrid scheme for solving the elastic registration problem using a regular grid discretization.

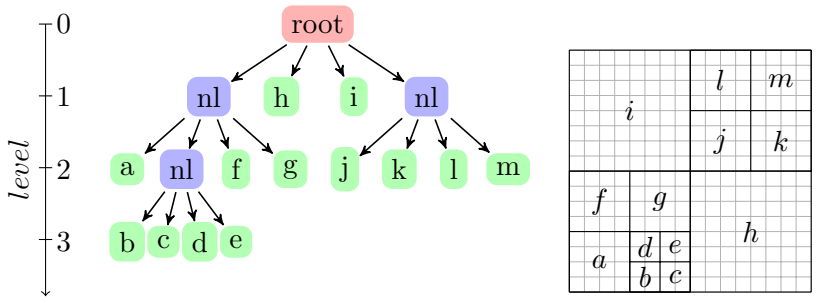
The use of octrees/quadtrees for image registration is also a fairly recent idea [27, 28, 38, 59, 60]. [27] presented a parametric registration algorithm using octree discretization and [28] used octrees for elastic image registration. [38] used quadtrees for affine image registration. [59] and [60] used a family of volumetric tensor product first order (linear) B-splines whose coefficients were defined on an octree and quadtree grid to model the transformation for the registration problem.

There is little work on parallel image registration. [18, 46, 62] focussed on rigid registration using derivative-free optimization algorithms. A steepest-descent approach was used in [35] for parallel rigid registration of 2-D images to 3-D volumes. A parallel non-rigid registration algorithm using a B-spline transformation model was presented in [36]. [55] parallelized the 3-D demon's registration algorithm. [9] used a fixed point iteration combined with parallel FFTs to solve the 2-D elastic registration problem. [43, 44] also used a fixed point iteration to solve the 2-D elastic registration problem and used a parallel Conjugate Gradient (CG) method to solve the linear system within the nonlinear iteration. [37] presented a parallel multigrid scheme to solve the 3-D optical flow problem.

Organization of the paper. The rest of this article is organized as follows. We describe the octree discretization and image interpolation schemes in Sections 2 and 3, respectively. Section 4 describes the linear and nonlinear solvers used to solve the optimization problem. Finally, in Section 5 we present the results from using the proposed algorithm for registering synthetic as well as medical images.

2 Octree discretization

We begin by defining some terms that will be used later in this section. An "*octree*" is a tree data structure that is used for spatial decomposition (Figure 1). The elements of an octree are known as "*octants*". Every octant has a maximum of eight "*children*". An octant with no children is called a "*leaf*" and an octant with one or more children is called an "*interior octant*". "*Complete*" octrees are octrees in which every interior octant has exactly eight children. The only octant with no parent is the "*root*" and all other octants have exactly one parent. Octants that have the same parent are called "*siblings*". The depth of an octant from the root is referred to as its "*level*". We use a "*linear*" octree representation (i.e., we exclude interior octants) using the "*Morton encoding*"

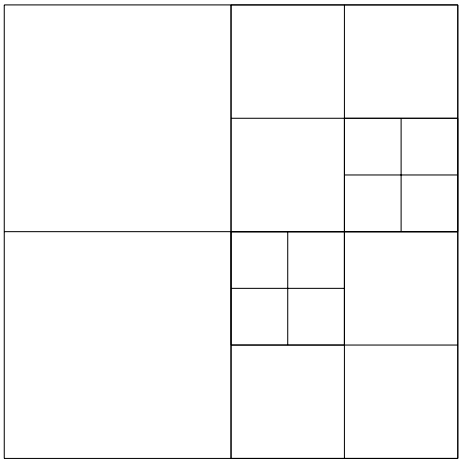


(a) Pointer representation (b) Linear representation

Figure 1: For simplicity, we use quadtrees to illustrate the concepts; quadtrees are 2-D analogues of octrees. (a) Tree representation of a quadtree and (b) decomposition of a square domain using the quadtree, superimposed over a uniform grid. The leaves are labeled in the Morton ordering.

3	3	2	2	1	1	1	1
3	3	2	2	1	1	1	1
2	2	3	3	2	2	50	100
2	2	3	3	2	2	4	10
4	4	3	3	100	500	1	1
4	4	3	3	1	10	1	1
1	1	2	2	100	102	10	11
1	1	2	2	100	101	10	12

(a) Regular Grid



(b) Linear Octree

Figure 2: (a) A regular grid image and (b) the linear octree constructed by coarsening the regular grid image.

scheme [13]. We use "*2:1 balanced octrees*" in this work; these are octrees in which there is no more than one octree-level difference between octants that share a face, edge, or vertex.

Next, we describe how we construct the parallel octree discretization from the high resolution fixed and moving images discretized on an uniform grid. We assume that the number of elements in each dimension of the uniform grid is a power of 2. Note that we only use the octree discretization for the displacements; we continue to use the original uniform grid representation for the given images. To make the distinction between the two discretizations clear, we use the term "octants" to refer to elements in the octree discretization and the term "voxels" to refer to elements in the original regular grid discretization.

We first construct two parallel complete linear octrees; one for the fixed image and another for the moving image. The uniform grids for the input images can be viewed as octrees in which all the octants are at the same octree-level. We coarsen these octrees by replacing every set of eight siblings by their parent as long as the maximum difference between the image values of the siblings

is less than an user-specified threshold (δ). After coarsening, we set the image value of the parent to be the average of the values of its children and repeat the process until further coarsening is not possible. This is illustrated in Figure 2. Next, we merge the two octrees and linearize the result using the algorithm described in [58].[3] We then use the algorithm described in [58] and construct a 2:1 balanced octree. We construct an octree mesh from the 2:1 balanced octree using the algorithm described in [57]. We discretize the optimization problem given in Equation 1.3 using trilinear finite elements on this octree mesh. A characteristic feature of such octree meshes is that they contain "hanging" vertices; these are vertices of octants that coincide with the centers of faces or mid-points of edges of other octants. The vertices of the former type are called "face-hanging" vertices and those of the latter type are called "edge-hanging" vertices. The 2:1 balance constraint ensures that there is at most one hanging vertex on any edge or face. The hanging vertices do not represent independent degrees of freedom in a FEM solution. We discussed a way to tackle these hanging vertices and build conforming finite element spaces in [52, 51, 57]; we follow the same approach in this work too.

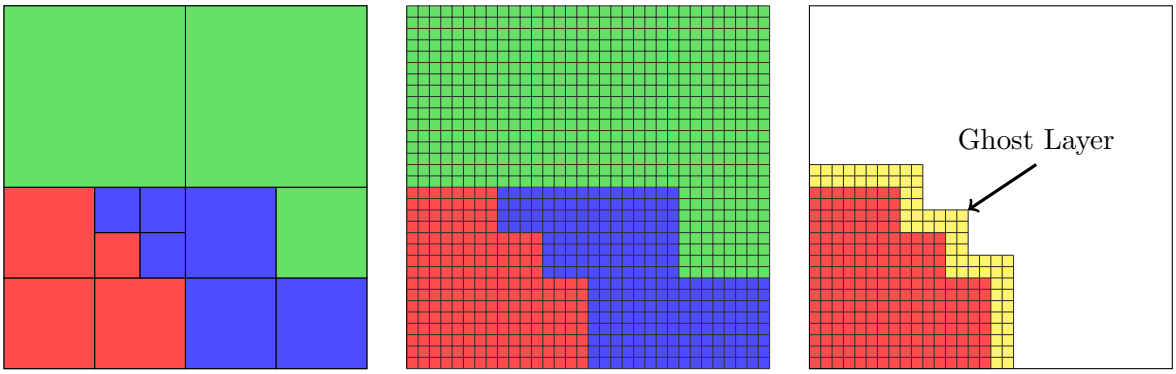
3 Interpolation

In this section, we describe how we construct continuous approximations to the images that can be used to evaluate their intensity values at arbitrary locations within the domain. Such evaluations are necessary for computing the integrals in Equation 1.3 and Equation 1.4 numerically. We use a gradient-based optimization algorithm and to compute the gradient, $g(u)$, (Equation 1.4) we need derivatives of the approximation of the moving image, $T(x)$. Although computationally efficient, piecewise trilinear image approximations are not suitable for our purposes because they are not continuously differentiable and we require image derivatives. For this reason, we use piecewise tricubic Hermite polynomials. Since the Hermite polynomial approximations interpolate the function and its derivatives, we need both T and ∇T to construct the image approximations. To compute ∇T, we use the second-order accurate central difference scheme. Once we have ∇T, then the approximations to the images within each voxel are given by Equation 3.1.

$$
\begin{aligned}
T(x,y,z) &= \sum_{i,j,k=0}^{1} \sum_{\substack{l,m,n=0 \\ l+m+n\leq 1}}^{1} \widehat{T}_{ijk}^{lmn} N_{ijk}^{lmn}(\xi,\eta,\zeta) \\
\widehat{T}_{ijk}^{lmn} &= \left(\frac{h}{2}\right)^{l+m+n} \left.\frac{\partial^{(l+m+n)} T}{\partial x^l \partial y^m \partial z^n}\right|_{\substack{x=x0+ih \\ y=y0+jh \\ z=z0+kh}} \\
\xi = -1 + \frac{2(x-x0)}{h} \;;\; \eta &= -1 + \frac{2(y-y0)}{h} \;;\; \zeta = -1 + \frac{2(z-z0)}{h} \\
(x,y,z) &\in [x0, x0+h) \times [y0, y0+h) \times [z0, z0+h)
\end{aligned}
\tag{3.1}
$$

where, N_{ijk}^{lmn} is given by Equation 3.2 and P_i^l is given by Equation 3.3. For each pair (i,l), exactly one of $P_i^l(-1)$, $P_i^l(1)$, $\frac{dP_i^l(-1)}{d\xi}$ and $\frac{dP_i^l(1)}{d\xi}$ is 1 and the rest are 0.

[3] Our implementation is capable of simultaneously registering multimodal images. For simplicity, we only explain the monomodal case. For the multimodal case, an octree is constructed for each modality and all the octrees are merged in the end.



(a) (b) (c)

Figure 3: Illustration of image partition: (a) An octree distributed on 3 processors, (b) the input image aligned with the octree and (c) the part of the input image owned by the first processor. In this example, the ghost layer received from other processors is 2 voxels thick.

$$\begin{aligned} N_{ijk}^{lmn}(\xi,\eta,\zeta) &= P_i^l(\xi)P_j^m(\eta)P_k^n(\zeta) \\ i,j,k,l,m,n \in \{0,1\} \quad &; \quad l+m+n \leq 1 \end{aligned} \tag{3.2}$$

$$\begin{aligned} P_0^0(\xi) = \frac{2-3\xi+\xi^3}{4} \;;\; P_0^1(\xi) = \frac{1-\xi-\xi^2+\xi^3}{4} \\ P_1^0(\xi) = \frac{2+3\xi-\xi^3}{4} \;;\; P_1^1(\xi) = \frac{-1-\xi+\xi^2+\xi^3}{4} \end{aligned} \tag{3.3}$$

3.1 Image partition

In our parallel implementation, we partition the images across the processors. The displacements computed during the solve may be such that some processors may need to access portions of the images owned by other processors in order to perform interpolation. To reduce the communication costs associated with this operation we impose the partition for the finest octree onto the images as well; hence, the portion of the images owned by each processor is aligned with the part of the finest octree owned by that processor. We then expand this initial partition of the images to include a layer of image voxels that is owned by other processors; the thickness of this layer is a few voxels and it is a parameter that can be controlled. This is illustrated in Figure 3. For the case of small deformations, most of the points, where the images and their gradients need to be evaluated, generated by each processor will lie within the portion of the images owned by that processor. We communicate the remaining points to the processors that own the respective image voxels, the receiving processor will evaluate the image and gradients at these points and communicate the results back to the processor that generated these points.

4 Solvers

In this section, we describe three techniques used to accelerate the solution of the elastic registration problem. First, we describe a Gauss-Newton method that generates good search directions for the optimization and succeeds in keeping the number of optimization iterations small. Second, we describe a multigrid scheme to efficiently solve the linear system of equations that arise in each optimization iteration. Third, we describe a grid continuation scheme that solves a sequence of optimization problems on increasingly finer grids to reduce the overall computational cost and to improve the convergence of the algorithm.

4.1 Gauss-Newton approximation

In this section, we describe how we solve Equation 1.4 using a Newton-type method. In the Newton's algorithm for optimization problems, the search directions, p, are generated by inverting the Hessian, H, in Equation 4.1:

$$\begin{aligned} H(u_{n-1})p_{n-1} &= -g(u_{n-1}) \\ u_n &= u_{n-1} + \alpha_{n-1}p_{n-1} \end{aligned} \tag{4.1}$$

where, α_{n-1} satisifies the Armijo backtracking condition [47].

$$w \cdot H(u)p = \gamma\, a(w,p) + \int w \cdot \left[\nabla T(u) \otimes \nabla T(u) - (S - T(u))\, \nabla^2 T(u)\right] p\, dx \;\; \forall p, w \tag{4.2}$$

The true Hessian for the registration problem, given by Equation 4.2, can be indefinite when $\gamma = 0$ due to the $\nabla^2 T(u)$ term and so the Newton step might not be a descent direction. Moreover, second derivatives of the image are unreliable for non-smooth images. Instead, we use a Gauss-Newton approximation (Equation 4.3), which is positive definite. This approximation is derived by dropping the $\nabla^2 T(u)$ term in Equation 4.2.

$$w \cdot H(u)p = \gamma\, a(w,p) + \int w \cdot (\nabla T(u) \otimes \nabla T(u))\, p\, dx \;\; \forall p, w \tag{4.3}$$

The discretized Hessian requires computation of the integrals in Equation 4.3. The $a(w,p)$ term is computed exactly using precomputed stencils. We use numerical integration to approximate the integral involving the $\nabla T(u) \otimes \nabla T(u)$ term. For this numerical integration, we used an integration rule in which the points of integrand evaluation coincided with the mesh points. Thus, the $\nabla T(u) \otimes \nabla T(u)$ part of the Hessian matrix[4] has a block diagonal form. It is cheaper to construct the matrix or apply the Matvec for this block diagonal approximation to the Hessian as compared to using a Hessian built using a more accurate integration rule with more points of integrand evaluation within each element.

[4] In our implementation, we do not build the Hessian matrix explicitly; instead, we implement a Hessian Matvec, i.e. a function that accepts a vector as input and returns the product of the Hessian matrix and the vector as the output.

Algorithm 1 GAUSS-NEWTON ALGORITHM

Repeat until convergence:

1. Given u, use interpolation to compute $T(u)$ and $\nabla T(u)$.
2. Evaluate the objective function and the gradient of the objective function.
3. Compute the Newton step, p, in Equation 4.2 using the multigrid solver.
4. Use line search to scale the step:
   - Let w be the Newton step.
   - Set $\alpha = 1$
   - While $J(u + \alpha w) > (J(u) + c_1 \alpha \nabla J(u)^T w)$
     - set $\alpha = c_2 \alpha$. $(c_2 \in (0, 1))$

   A trust region approach could be used instead of line search.
5. Update u as shown in Equation 4.2.
6. Declare convergence if one of the following holds:
   - The step-length is smaller than a given tolerance, which is typically some fraction of the voxel size.
   - Reduction in gradient is sufficient.

4.2 Multigrid preconditioner

The Gauss-Newton Hessian, described in the previous section, can be highly ill-conditioned. We tackle the problem of ill-conditioning using the multigrid method, which is known to be robust for ill-conditioned systems. We solve Equation 4.1 using a Conjugate Gradient algorithm preconditioned using one multigrid V-cycle. We use a 3×3 block-Jacobi smoother at each level in the V-cycle and use a direct solver (LU) at the coarsest grid. We use the restriction and prolongation operators described in [51, 52]. To form the coarse grid operator, we first transfer the values of $\nabla T(u)$ from the fine grid vertices to the coarse grid vertices by injection. Note that we restrict $\nabla T(u)$ and not $T(u)$. We then assemble the coarse grid operators by using the coarse grid discretization for the elasticity part and nodal integration for the image part as decribed in the previous section.

4.3 Grid continuation

We use a multiscale optimization algorithm to reduce the overall computational cost and to improve the convergence of the algorithm. This idea has been used in other works as well [3, 29, 31]. In this approach, we solve a sequence of optimization problems on increasingly finer grids. The solution of each optimization problem is used as an initial guess for the subsequent optimization on the next finer level. The coarser grid iterations are cheaper than the fine grid iterations and they help escape local minima by aligning the coarse-level details. We also use the initial guess to construct an octree corresponding to the deformed moving image instead of the original moving image. The multiscale optimization algorithm is listed in Algorithm 2.

Algorithm 2 GRID CONTINUATION

1. Construct a sequence of coarse images from the given image by averaging.
2. Solve the optimization problem on the coarsest grid using zero guess.
3. Interpolate the solution to the next finer grid and use it as an initial guess for the optimization problem on that grid.
4. Repeat the last step until the finest grid is reached.

5 Results

In this section, we present the results from using the multiscale Gauss-Newton algorithm to register synthetic as well as medical images. We performed four sets of experiments:

- We tested the performance of the algorithm on two synthetic examples.
- We tested the performance of the algorithm on medical MR images of brains.
- We tested the effect of the thresholding parameter (δ) on the registration accuracy and the corresponding reduction in problem size. This parameter controls the number of elements in the octree; For a given image, using higher values of δ would result in coarser octrees compared to those constructed using lower values of δ.
- We tested the fixed-size and isogranular scalability of our MPI-based implementation of the proposed algorithms on the Sun Constellation Linux Cluster "Ranger" at the Texas Advanced Computing Center (TACC).

The Newton iterations in all these experiments were terminated when the maximum step length was less than $0.1 \times h$, where h is the regular grid spacing. In these experiments, we measured the performance of registration for different regularization parameters (γ) using the following metrics: (A) the reduction in absolute value of the mismatch between the registered and fixed images, (B) the number of optimization iterations, (C) the relative reductions in objective function, and (D) the relative reduction in the 2-norm of the gradient. We also computed the determinants of the Jacobians of the recovered deformations at a small number (7) of points within each element and report the maximum and minimum values across all elements; values closer to 1 indicate small deformations, values greater than 1 indicate expansion, values lesser than 1 indicate compression and negative values indicate non-physical deformations such as those involving intersecting or flipped elements.

5.1 Synthetic examples

We first present two synthetic example problems of resolution $256 \times 256 \times 256$ in Figures 4 and 7. In the former example, the moving image was chosen to be $255 \sin^2(2\pi x) \sin^2(2\pi y) \sin^2(2\pi z)$ and the fixed image was generated by applying 3 successive synthetic diffeomorphic displacement field to this image. We found the maximum and minimum values for the determinants of the Jacobians of the first deformation to be 1.12 and 0.84, respectively; the corresponding values for the second deformation were 1.28 and 0.64, respectively and the corresponding values for the third deformation were 1.70 and 0.08, respectively. In the latter example, the moving image is a sphere

Table 1: Performance of the optimizer for the synthetic images shown in Figure 4. J is the objective function and g is the 2-norm of the gradient. γ is the regularization parameter. Approximately 10^4 elements were used in the finest grid. The Newton iterations were terminated when the maximum step-length was less than $0.1 \times h$, where h is the regular grid spacing for that level. The maximum and minimum values of the determinants of the Jacobian of the recovered deformation for $\gamma = 10^2$ was found to be 1.89 and -0.14, respectively. The maximum and minimum values of the determinants of the Jacobian of the recovered deformation for $\gamma = 10^3$ was found to be 1.56 and 0.35, respectively. The maximum and minimum values of the determinants of the Jacobian of the recovered deformation for $\gamma = 10^4$ was found to be 1.21 and 0.85, respectively.

γ	Level 1 (Coarsest)			Level 2			Level 3			Level 4			Level 5 (Finest)		
	Its.	$\frac{J}{J_0}$	$\frac{g}{g_0}$	Its.	$\frac{J}{J_0}$	$\frac{g}{g_0}$	Its.	$\frac{J}{J_0}$	$\frac{g}{g_0}$	Its.	$\frac{J}{J_0}$	$\frac{g}{g_0}$	Its.	$\frac{J}{J_0}$	$\frac{g}{g_0}$
10^2	16	0.07	0.06	8	0.67	0.03	12	0.92	0.01	15	0.98	0.01	9	0.99	0.01
10^3	4	0.30	0.07	3	0.92	0.04	2	0.99	0.18	3	0.99	0.06	3	0.99	0.05
10^4	2	0.78	0.14	2	0.98	0.03	2	0.99	0.05	2	0.99	0.04	2	0.99	0.04

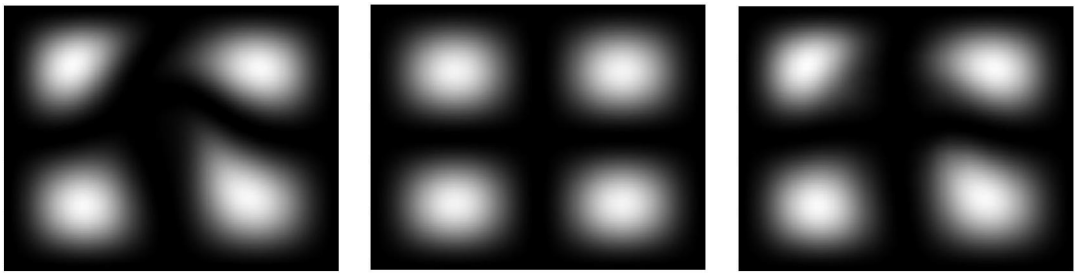
and the fixed image is a partial torus. In the figures, we only show a few selected slices in which the differences between the fixed and moving images are easily noticeable. We report the results from solving these registration problems using different regularization parameters in Figures 5 and 8, respectively. We show the corresponding recovered deformations for these two examples in Figures 6 and 9, respectively. We also report the corresponding number of Gauss-Newton iterations and relative reductions in objective function and gradient for each level of the multiscale algorithm in Tables 1 and 2, respectively.

5.2 Medical imaging examples

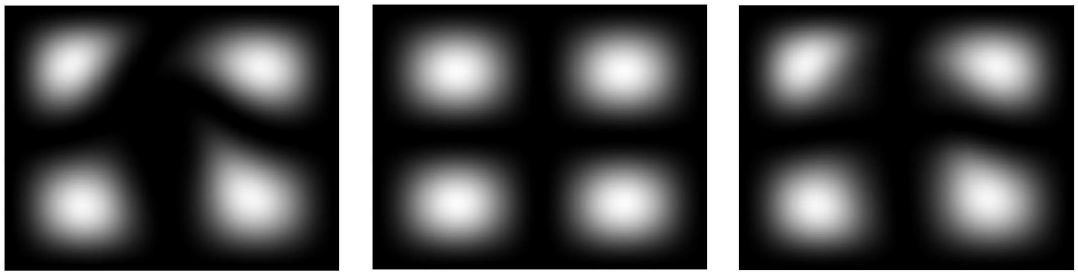
Next, we tested the performance of the algorithm on medical MR images of resolution $256 \times 256 \times 256$. Figure 10 shows a few z-plane crosssection of the fixed, moving and registered MR images of the brain of the same subject taken at different times. We show the result of registration for different choices of the regularization parameter in Figure 11. The corresponding number of Gauss-Newton iterations and relative reductions in objective function and gradient for each level of the multiscale algorithm is reported in Table 3. Another medical imaging example is shown in Figure 12; this is an example of inter-subject registration. The mismatch between the fixed and moving images before and after registration is shown in Figure 13. This experiment was repeated for different registration parameters and for each case the respective performance metrics like the number of optimization iterations and relative reductions in objective function and gradient are reported in Table 4. We also computed the determinants of the Jacobians of the deformation at the centers of each voxel and this is shown in Figure 14.

5.3 Effect of the thresholding parameter

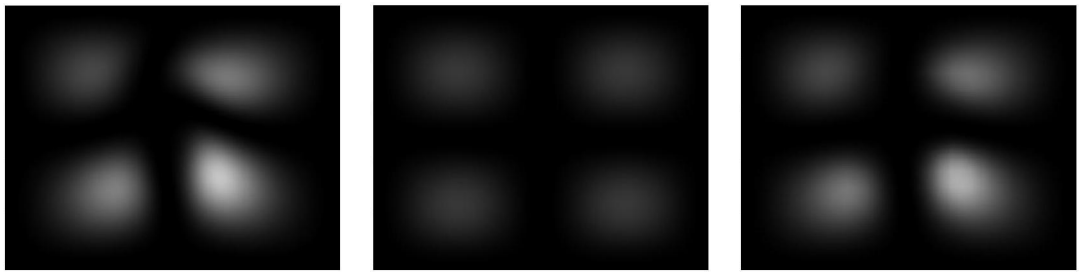
In Figure 15, we show the effect of the thresholding parameter, δ, on the registration accuracy for the example shown in Figure 12. We also report the relative reduction in mismatch between the registered and fixed images and the corresponding number of octants in the finest octree in Table 5. The thresholding parameter has little effect on the registration accuracy but has a significant



(a) Fixed (Slice A) (b) Moving (c) Registered

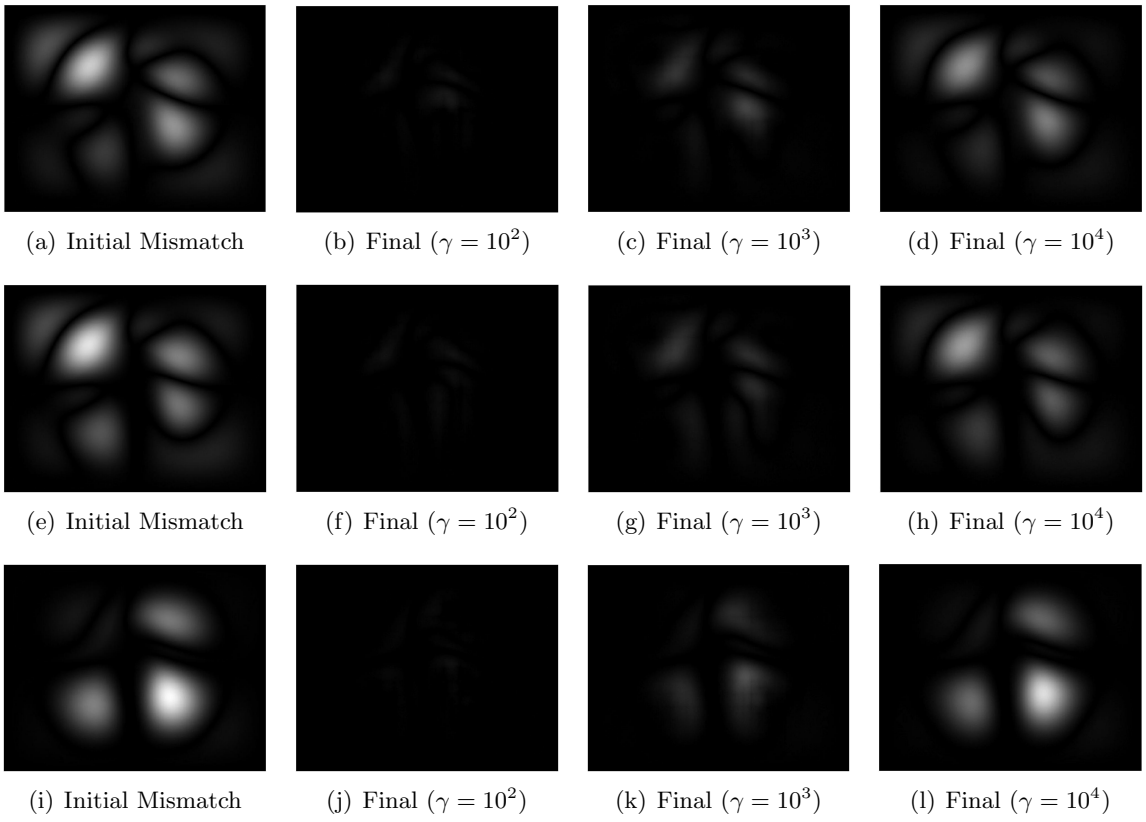


(d) Fixed (Slice B) (e) Moving (f) Registered



(g) Fixed (Slice C) (h) Moving (i) Registered

Figure 4: First synthetic example. The moving image was chosen to be $255\sin^2(2\pi x)\sin^2(2\pi y)\sin^2(2\pi z)$ and the fixed image was generated by applying 3 successive synthetic diffeomorphic displacement field to this image. The resolution of each image was $256 \times 256 \times 256$. Each row shows a z-plane crosssection of the fixed, moving and corresponding registered (deformed moving) images using the regularization parameter: $\gamma = 1000$.



(a) Initial Mismatch (b) Final ($\gamma = 10^2$) (c) Final ($\gamma = 10^3$) (d) Final ($\gamma = 10^4$)

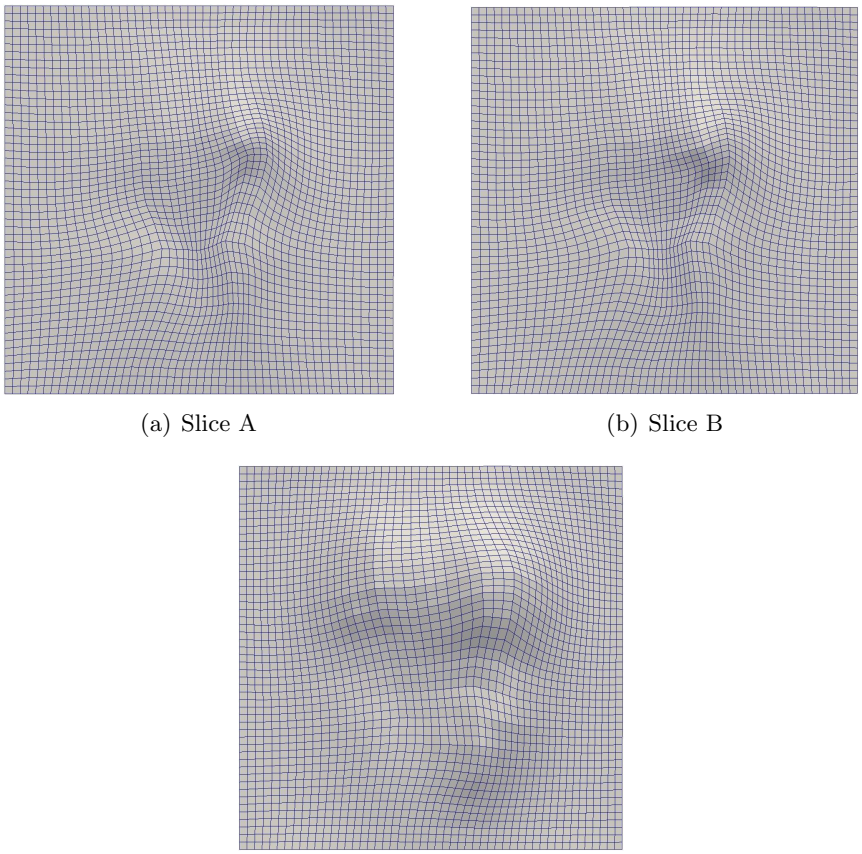
(e) Initial Mismatch (f) Final ($\gamma = 10^2$) (g) Final ($\gamma = 10^3$) (h) Final ($\gamma = 10^4$)

(i) Initial Mismatch (j) Final ($\gamma = 10^2$) (k) Final ($\gamma = 10^3$) (l) Final ($\gamma = 10^4$)

Figure 5: Results from using the proposed methology on the images shown in Figure 4. Each row shows a z-plane crosssection of the initial mismatch between the fixed and moving images and the final mismatch between the registered and fixed images for different regularization parameters (γ).

Table 2: Performance of the optimizer for the synthetic images shown in Figure 7. J is the objective function and g is the 2-norm of the gradient. γ is the regularization parameter. Approximately 2.9×10^5 elements were used in the finest grid. The Newton iterations were terminated when the maximum step-length was less than $0.1 \times h$, where h is the regular grid spacing for that level or after 30 iterations. The maximum and minimum values of the determinants of the Jacobian of the recovered deformation for $\gamma = 10^3$ was found to be 84.82 and -11.94, respectively. The maximum and minimum values of the determinants of the Jacobian of the recovered deformation for $\gamma = 10^4$ was found to be 3.33 and -0.42, respectively. The maximum and minimum values of the determinants of the Jacobian of the recovered deformation for $\gamma = 10^5$ was found to be 1.07 and 0.97, respectively.

γ	Level 1 (Coarsest)			Level 2			Level 3			Level 4			Level 5 (Finest)		
	Its.	$\frac{J}{J_0}$	$\frac{g}{g_0}$	Its.	$\frac{J}{J_0}$	$\frac{g}{g_0}$	Its.	$\frac{J}{J_0}$	$\frac{g}{g_0}$	Its.	$\frac{J}{J_0}$	$\frac{g}{g_0}$	Its.	$\frac{J}{J_0}$	$\frac{g}{g_0}$
10^3	17	0.21	0.06	30	0.55	0.12	21	0.81	0.16	30	0.88	0.18	30	0.91	0.19
10^4	5	0.75	0.17	8	0.93	0.08	11	0.97	0.09	30	0.98	0.06	30	0.99	0.16
10^5	2	0.97	0.15	1	0.99	-	2	0.99	0.09	2	0.99	0.07	2	0.99	0.07



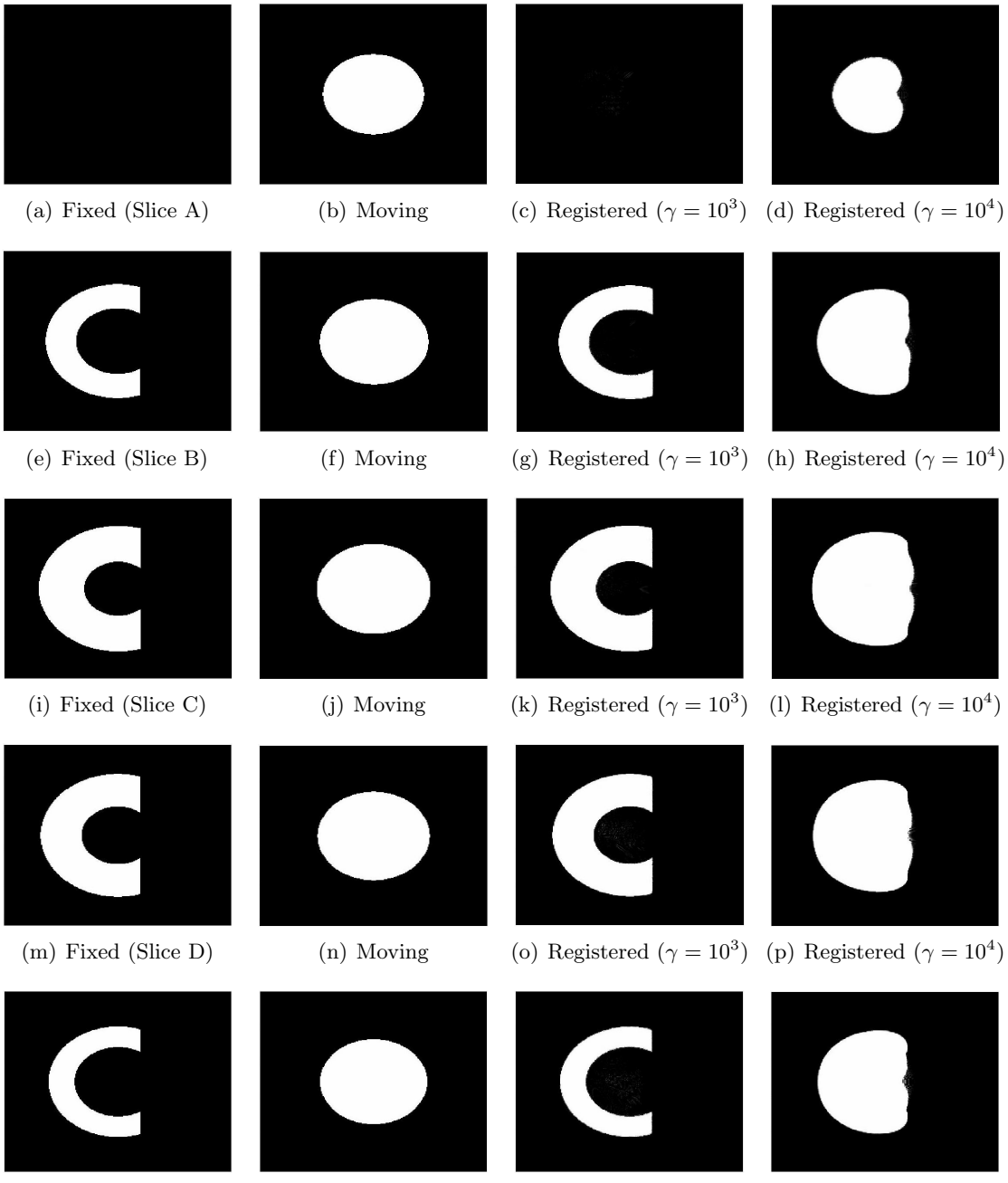
(a) Slice A (b) Slice B

(c) Slice C

Figure 6: The z-plane crosssection of the reconstructed deformation for the example shown in Figure 4 using $\gamma = 100$.

Table 3: Performance of the optimizer for the MR images shown in Figure 10. J is the objective function and g is the 2-norm of the gradient. γ is the regularization parameter. Approximately 1.28×10^6 elements were used in the finest grid. The Newton iterations were terminated when the maximum step-length was less than $0.1 \times h$, where h is the regular grid spacing for that level. The maximum and minimum values of the determinants of the Jacobian of the recovered deformation for $\gamma = 200$ was found to be 2.79 and 0.087, respectively. The maximum and minimum values of the determinants of the Jacobian of the recovered deformation for $\gamma = 500$ was found to be 1.65 and 0.49, respectively. The maximum and minimum values of the determinants of the Jacobian of the recovered deformation for $\gamma = 1000$ was found to be 1.35 and 0.67, respectively.

γ	Level 1 (Coarsest)			Level 2			Level 3			Level 4			Level 5 (Finest)		
	Its.	$\frac{J}{J_0}$	$\frac{g}{g_0}$	Its.	$\frac{J}{J_0}$	$\frac{g}{g_0}$	Its.	$\frac{J}{J_0}$	$\frac{g}{g_0}$	Its.	$\frac{J}{J_0}$	$\frac{g}{g_0}$	Its.	$\frac{J}{J_0}$	$\frac{g}{g_0}$
200	4	0.67	0.34	11	0.69	0.16	11	0.77	13.1	11	0.89	1.6e-2	12	0.97	8.9e-3
500	6	0.73	0.29	4	0.82	0.36	4	0.85	0.195	4	0.94	0.055	4	0.98	0.038
1000	3	0.78	0.21	4	0.87	0.16	4	0.92	0.106	27	0.97	3.1e-3	15	0.99	6.3e-3



(a) Fixed (Slice A) (b) Moving (c) Registered ($\gamma = 10^3$) (d) Registered ($\gamma = 10^4$)

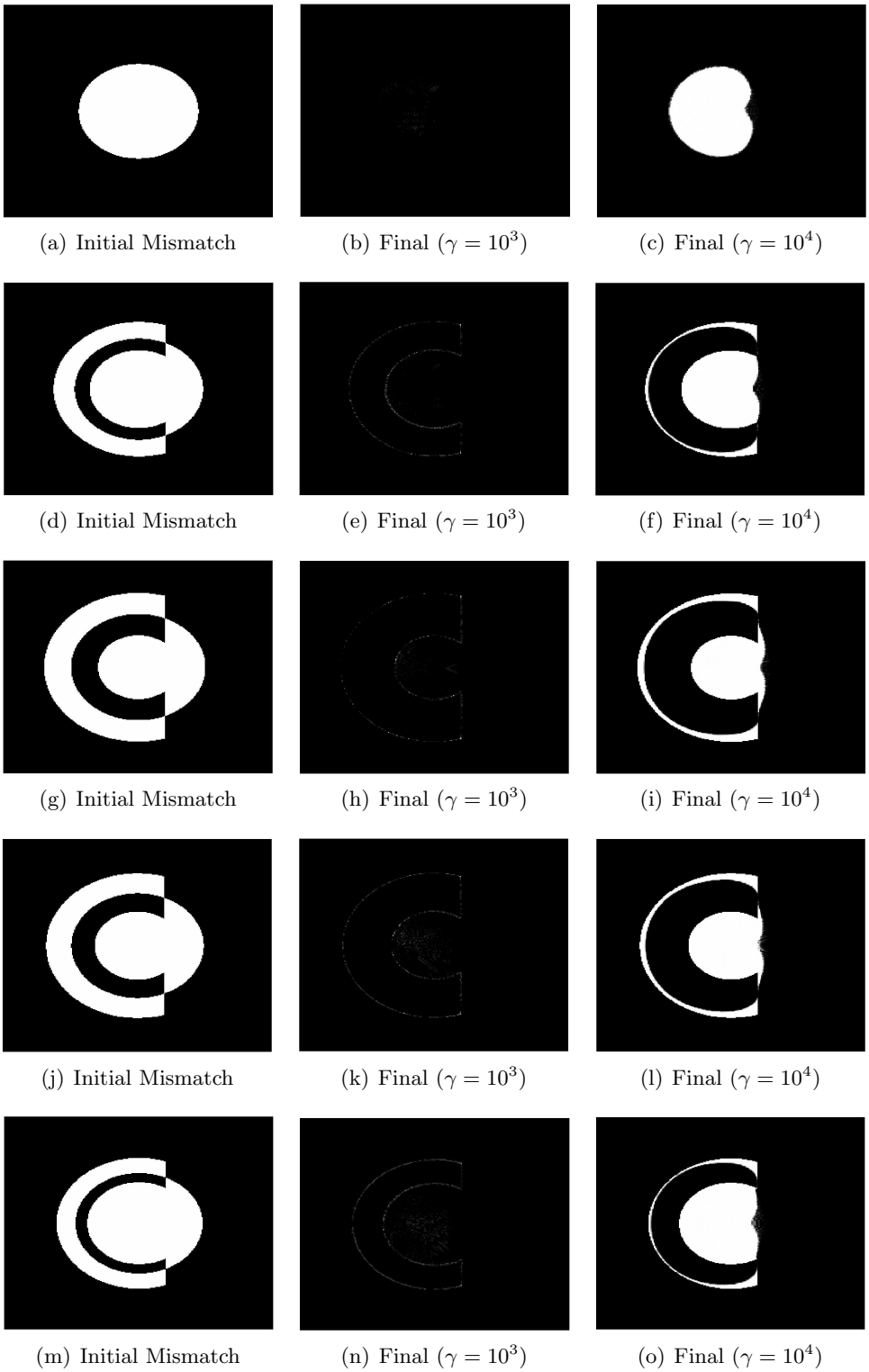
(e) Fixed (Slice B) (f) Moving (g) Registered ($\gamma = 10^3$) (h) Registered ($\gamma = 10^4$)

(i) Fixed (Slice C) (j) Moving (k) Registered ($\gamma = 10^3$) (l) Registered ($\gamma = 10^4$)

(m) Fixed (Slice D) (n) Moving (o) Registered ($\gamma = 10^3$) (p) Registered ($\gamma = 10^4$)

(q) Fixed (Slice E) (r) Moving (s) Registered ($\gamma = 10^3$) (t) Registered ($\gamma = 10^4$)

Figure 7: Second synthetic example. The moving image is a sphere and the fixed image is a partial torus. The resolution of each image was $256 \times 256 \times 256$. Each row shows a z-plane crosssection of the fixed, moving and corresponding registered (deformed moving) images for the regularization parameters: $\gamma = 10^3$ and $\gamma = 10^4$.



(a) Initial Mismatch (b) Final ($\gamma = 10^3$) (c) Final ($\gamma = 10^4$)

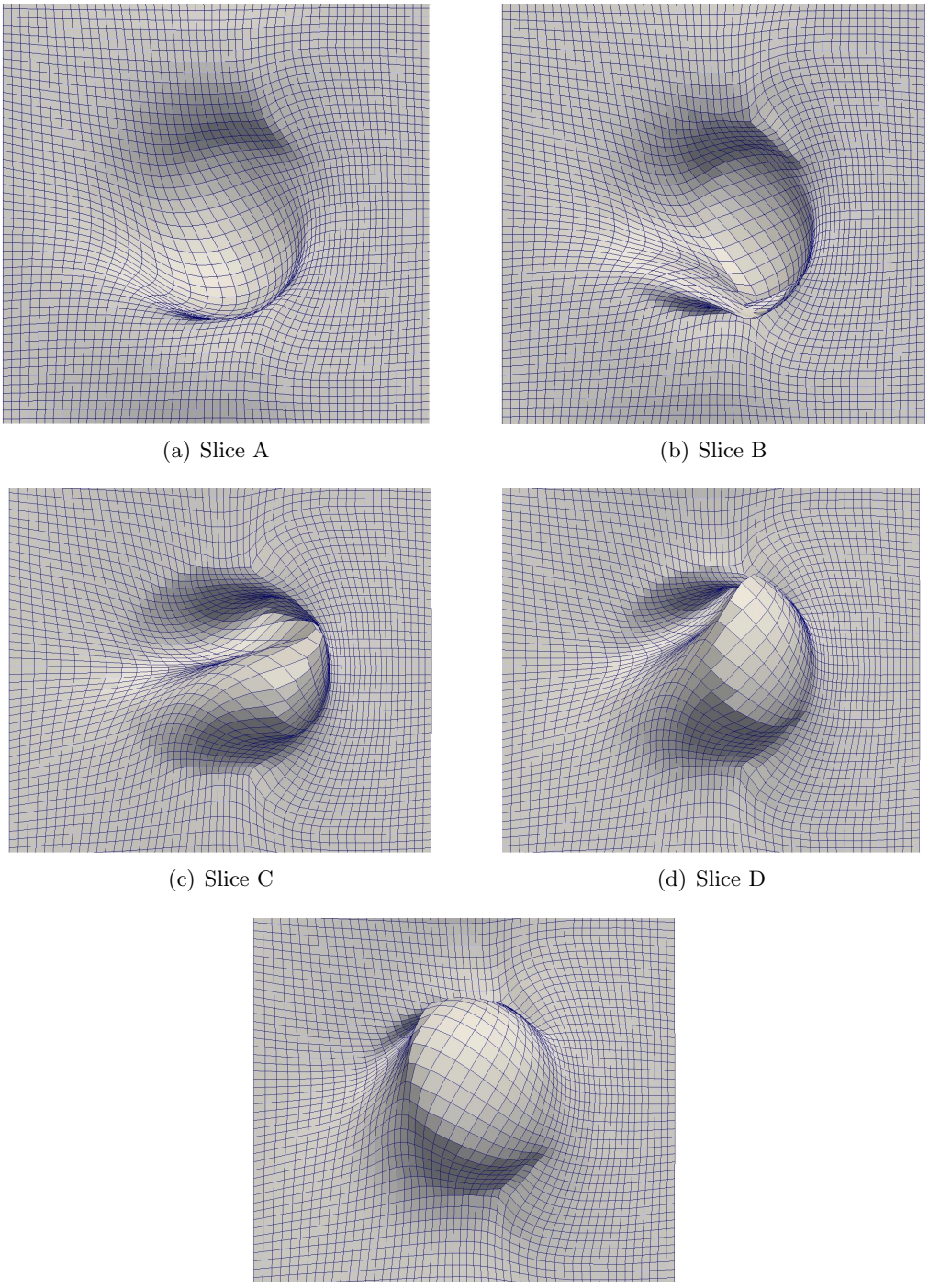
(d) Initial Mismatch (e) Final ($\gamma = 10^3$) (f) Final ($\gamma = 10^4$)

(g) Initial Mismatch (h) Final ($\gamma = 10^3$) (i) Final ($\gamma = 10^4$)

(j) Initial Mismatch (k) Final ($\gamma = 10^3$) (l) Final ($\gamma = 10^4$)

(m) Initial Mismatch (n) Final ($\gamma = 10^3$) (o) Final ($\gamma = 10^4$)

Figure 8: Results from using the proposed methology on the images shown in Figure 7. Each row shows a z-plane crosssection of the initial mismatch between the fixed and moving images and the final mismatch between the registered and fixed images for different regularization parameters (γ).



(a) Slice A

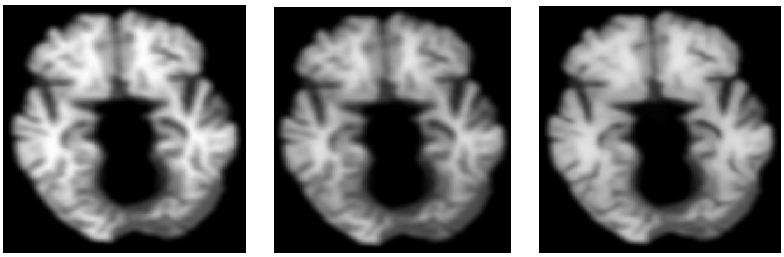
(b) Slice B

(c) Slice C

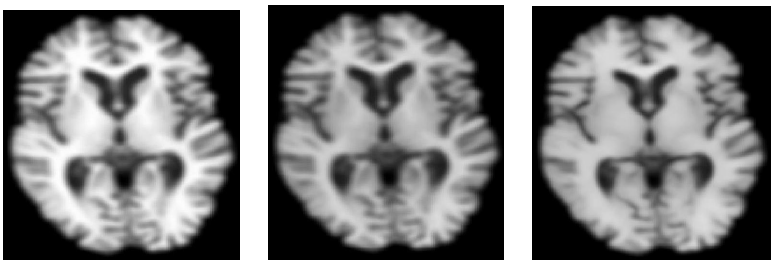
(d) Slice D

(e) Slice E

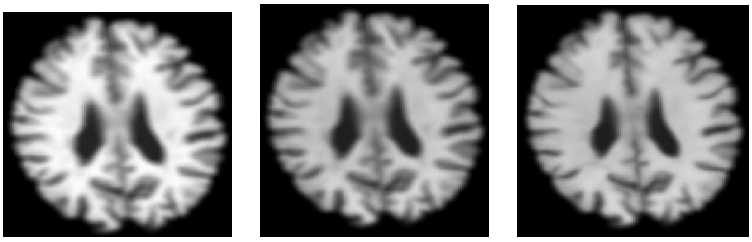
Figure 9: The z-plane crosssection of the reconstructed deformation for the example shown in Figure 7 using $\gamma = 1000$.



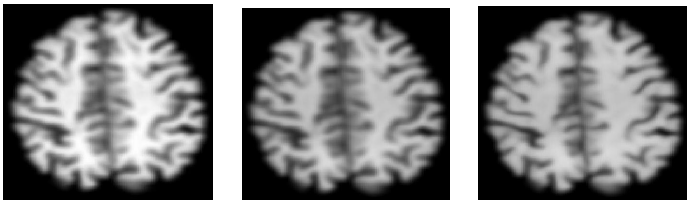
(a) Fixed (Slice A) (b) Moving (c) Registered



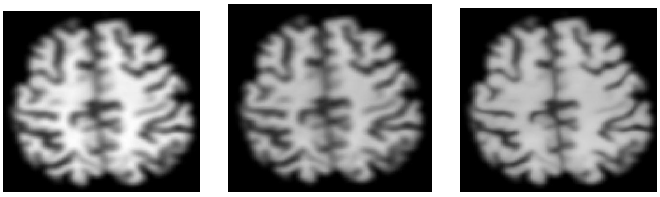
(d) Fixed (Slice B) (e) Moving (f) Registered



(g) Fixed (Slice C) (h) Moving (i) Registered

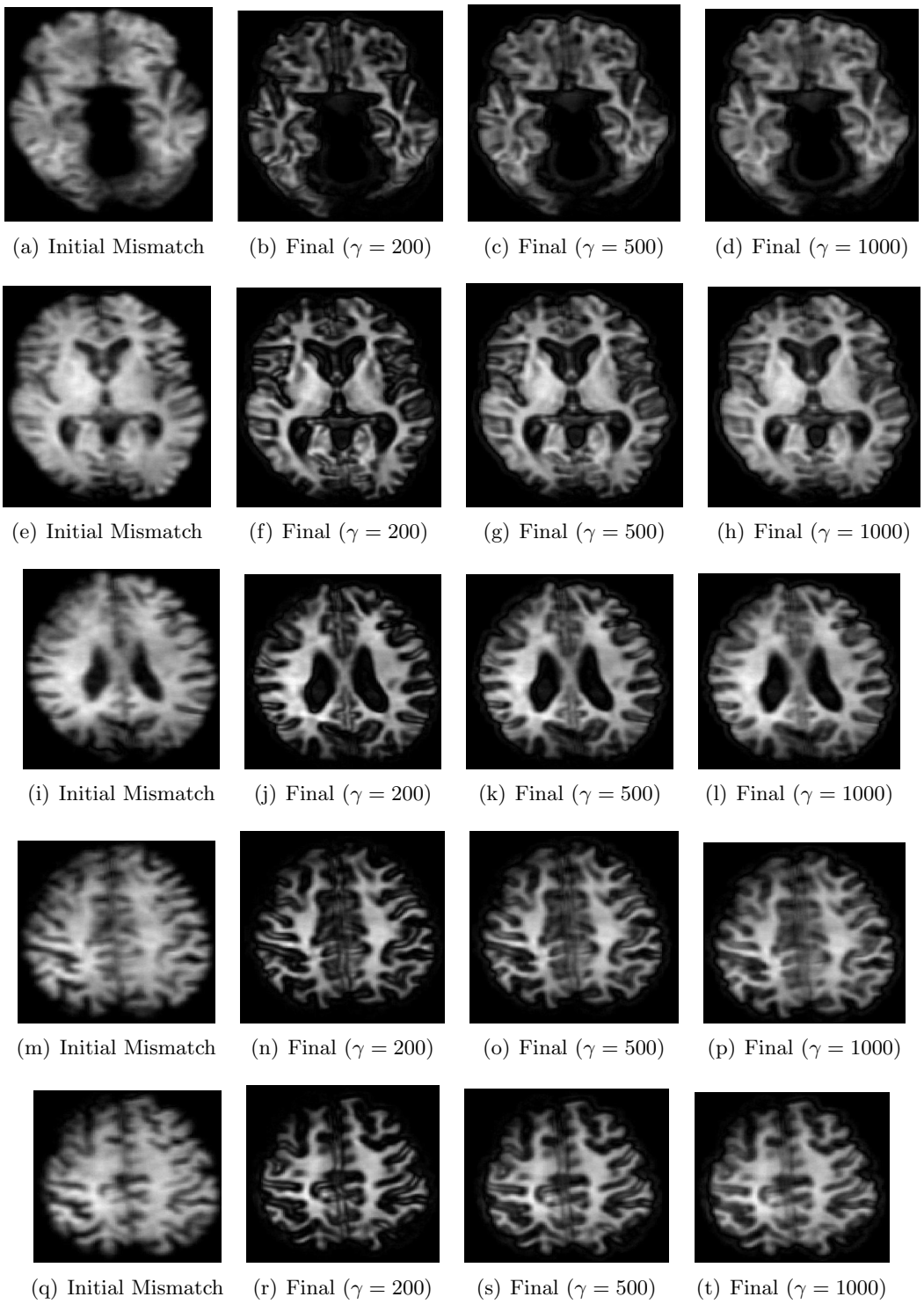


(j) Fixed (Slice D) (k) Moving (l) Registered



(m) Fixed (Slice E) (n) Moving (o) Registered

Figure 10: Skull stripped MR images of the brain of the same subject taken at different times and the corresponding registered (deformed moving) image for $\gamma = 200$. Each row shows a z-plane crosssection of the fixed, moving and registered images.



(a) Initial Mismatch (b) Final ($\gamma = 200$) (c) Final ($\gamma = 500$) (d) Final ($\gamma = 1000$)

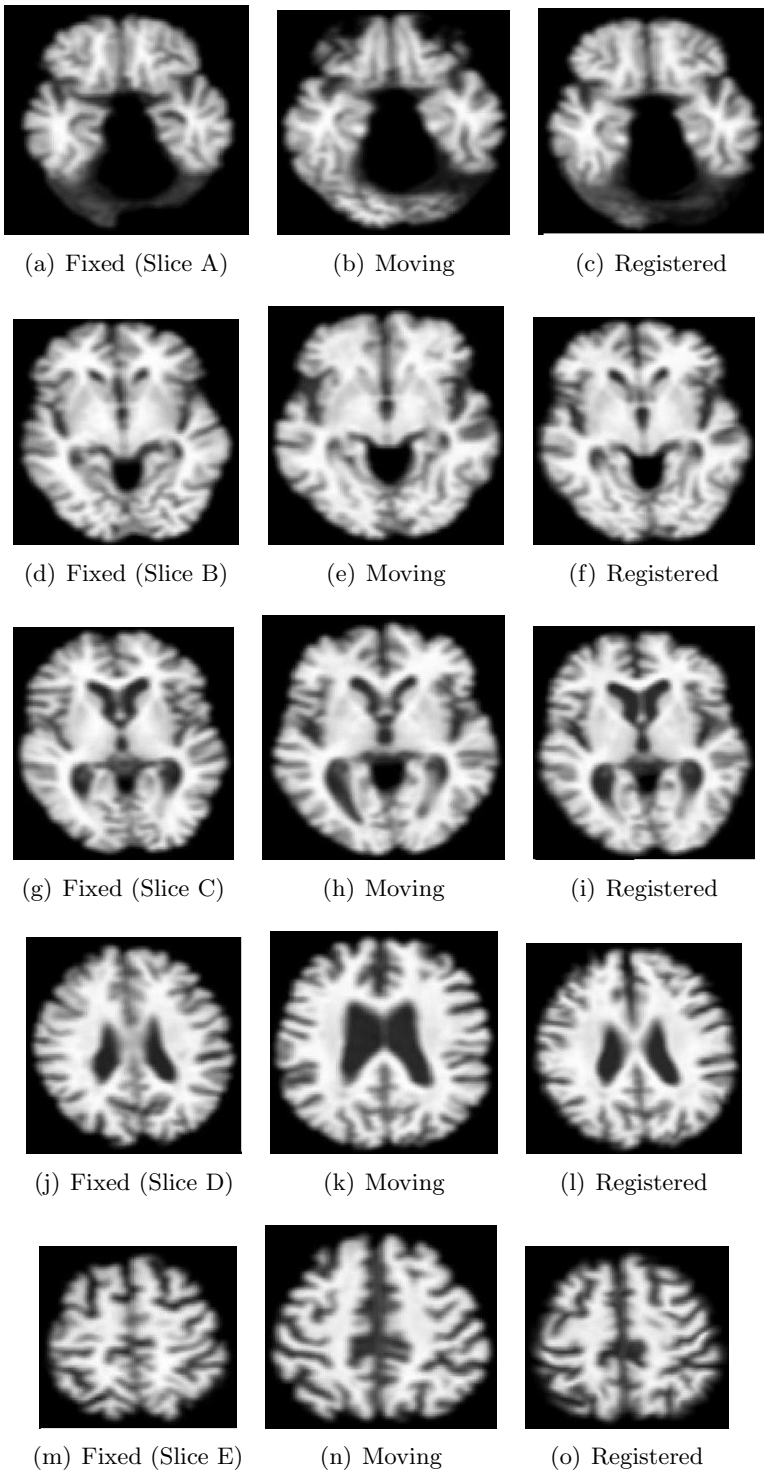
(e) Initial Mismatch (f) Final ($\gamma = 200$) (g) Final ($\gamma = 500$) (h) Final ($\gamma = 1000$)

(i) Initial Mismatch (j) Final ($\gamma = 200$) (k) Final ($\gamma = 500$) (l) Final ($\gamma = 1000$)

(m) Initial Mismatch (n) Final ($\gamma = 200$) (o) Final ($\gamma = 500$) (p) Final ($\gamma = 1000$)

(q) Initial Mismatch (r) Final ($\gamma = 200$) (s) Final ($\gamma = 500$) (t) Final ($\gamma = 1000$)

Figure 11: Results from using the proposed methology for registering the example shown in Figure 10. Each row shows a z-plane crosssection of the initial mismatch between the fixed and moving images and the final mismatch between the registered and fixed images for different regularization parameters (γ).



(a) Fixed (Slice A) (b) Moving (c) Registered

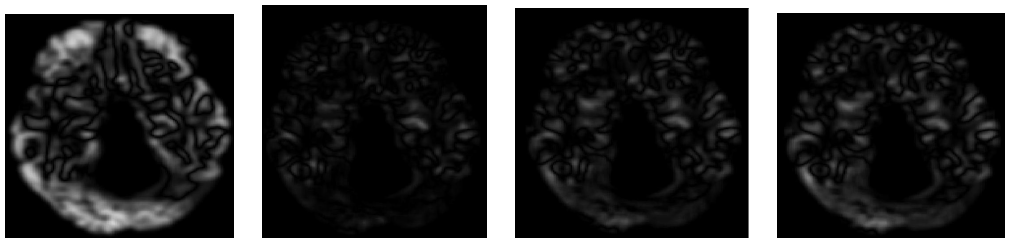
(d) Fixed (Slice B) (e) Moving (f) Registered

(g) Fixed (Slice C) (h) Moving (i) Registered

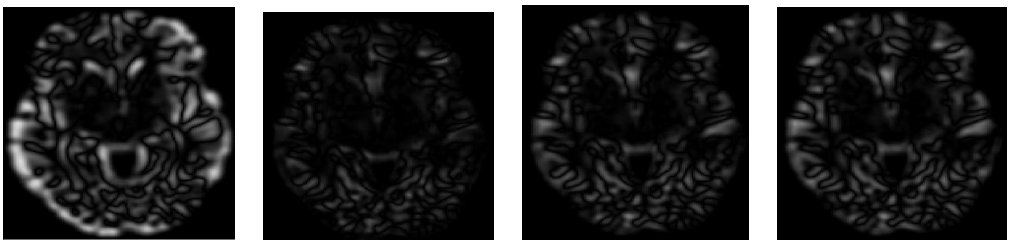
(j) Fixed (Slice D) (k) Moving (l) Registered

(m) Fixed (Slice E) (n) Moving (o) Registered

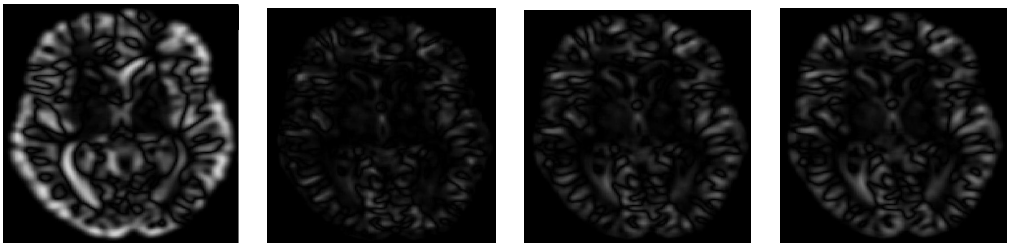
Figure 12: Skull stripped MR images of the brains of two different subjects and the corresponding registered (deformed moving) image for $\gamma = 3000$. Each row shows a z-plane crosssection of the fixed, moving and registered images.



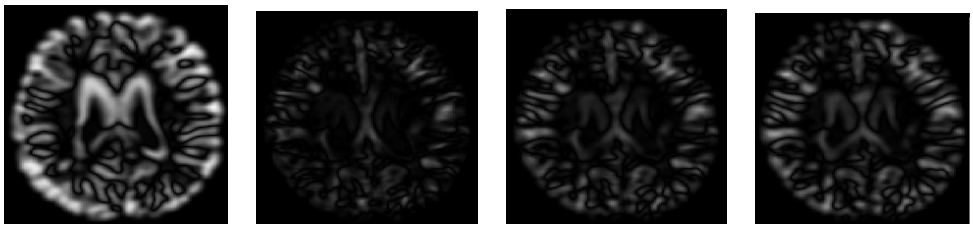
(a) Initial Mismatch (b) Final ($\gamma = 1000$) (c) Final ($\gamma = 3000$) (d) Final ($\gamma = 5000$)



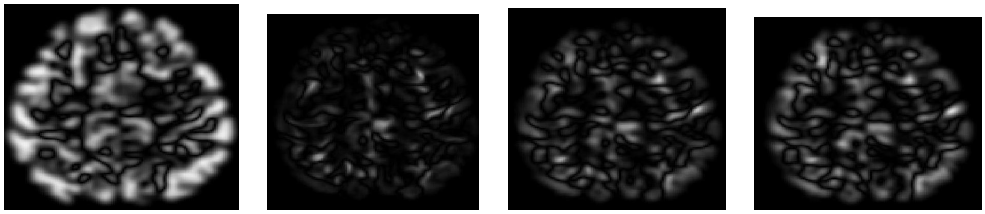
(e) Initial Mismatch (f) Final ($\gamma = 1000$) (g) Final ($\gamma = 3000$) (h) Final ($\gamma = 5000$)



(i) Initial Mismatch (j) Final ($\gamma = 1000$) (k) Final ($\gamma = 3000$) (l) Final ($\gamma = 5000$)

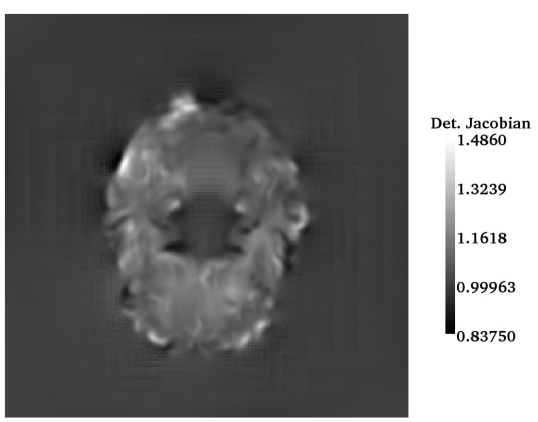


(m) Initial Mismatch (n) Final ($\gamma = 1000$) (o) Final ($\gamma = 5000$) (p) Final ($\gamma = 5000$)

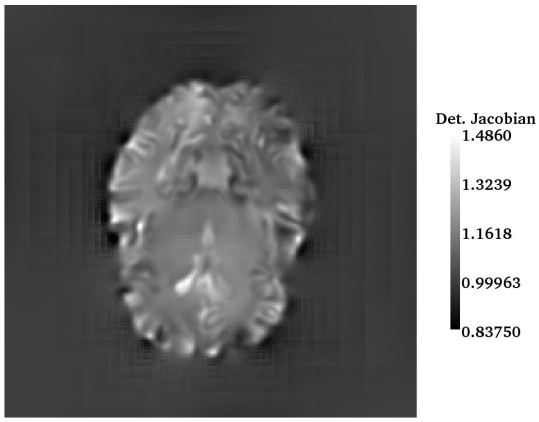


(q) Initial Mismatch (r) Final ($\gamma = 1000$) (s) Final ($\gamma = 3000$) (t) Final ($\gamma = 5000$)

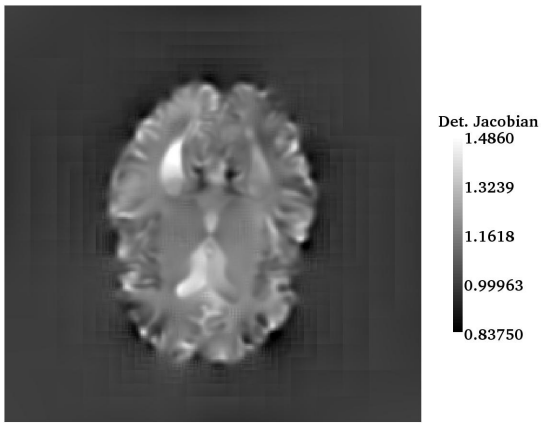
Figure 13: Results from using the proposed methology for registering the example shown in Figure 12. Each row shows a z-plane crosssection of the initial mismatch between the fixed and moving images and the final mismatch between the registered and fixed images for different regularization parameters (γ).



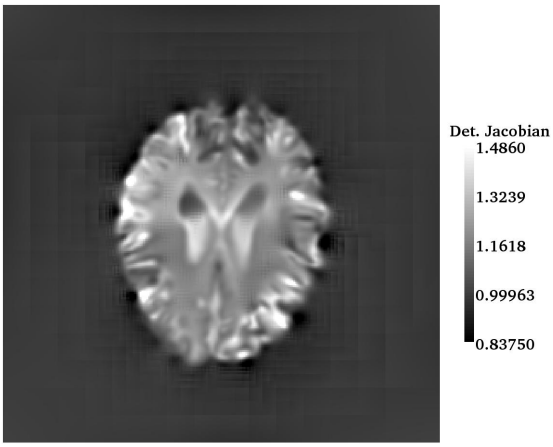
(a) Slice A



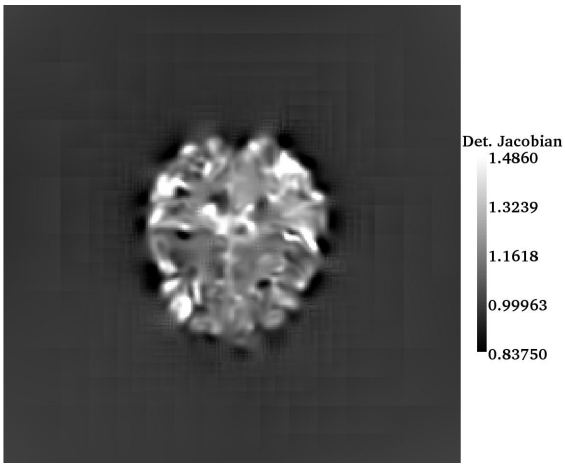
(b) Slice B



(c) Slice C



(d) Slice D



(e) Slice E

Figure 14: Determinants of the Jacobians at the centers of voxels for the example shown in Figure 12.

Table 4: Performance of the optimizer for registering the MR images shown in Figure 12. J is the objective function and g is the 2-norm of the gradient. γ is the regularization parameter. The threshold parameter, δ, was set equal to 10 in this experiment. Approximately 1.4×10^6 elements were used in the finest grid. The Newton iterations were terminated when the maximum step-length was less than $0.1 \times h$, where h is the regular grid spacing for that level. We computed the determinant of the Jacobians of the deformation at 7 points within each voxel. The maximum and minimum values of the determinants of the Jacobian of the recovered deformation for $\gamma = 1000$ was found to be 11.15 and -1.34, respectively. The maximum and minimum values of the determinants of the Jacobian of the recovered deformation for $\gamma = 3000$ was found to be 3.83 and 0.53, respectively. The maximum and minimum values of the determinants of the Jacobian of the recovered deformation for $\gamma = 5000$ was found to be 2.19 and 0.74, respectively.

γ	Level 1 (Coarsest)			Level 2			Level 3			Level 4			Level 5 (Finest)		
	Its.	$\frac{J}{J_0}$	$\frac{g}{g_0}$	Its.	$\frac{J}{J_0}$	$\frac{g}{g_0}$	Its.	$\frac{J}{J_0}$	$\frac{g}{g_0}$	Its.	$\frac{J}{J_0}$	$\frac{g}{g_0}$	Its.	$\frac{J}{J_0}$	$\frac{g}{g_0}$
1e3	8	0.26	0.06	9	0.69	0.08	20	0.70	0.01	43	0.84	4e-3	14	0.95	6e-3
3e3	3	0.44	0.13	3	0.82	0.20	7	0.84	0.04	24	0.92	0.01	33	0.98	3e-3
5e3	2	0.54	0.25	2	0.86	0.33	9	0.88	0.02	9	0.95	0.01	12	0.98	5e-3

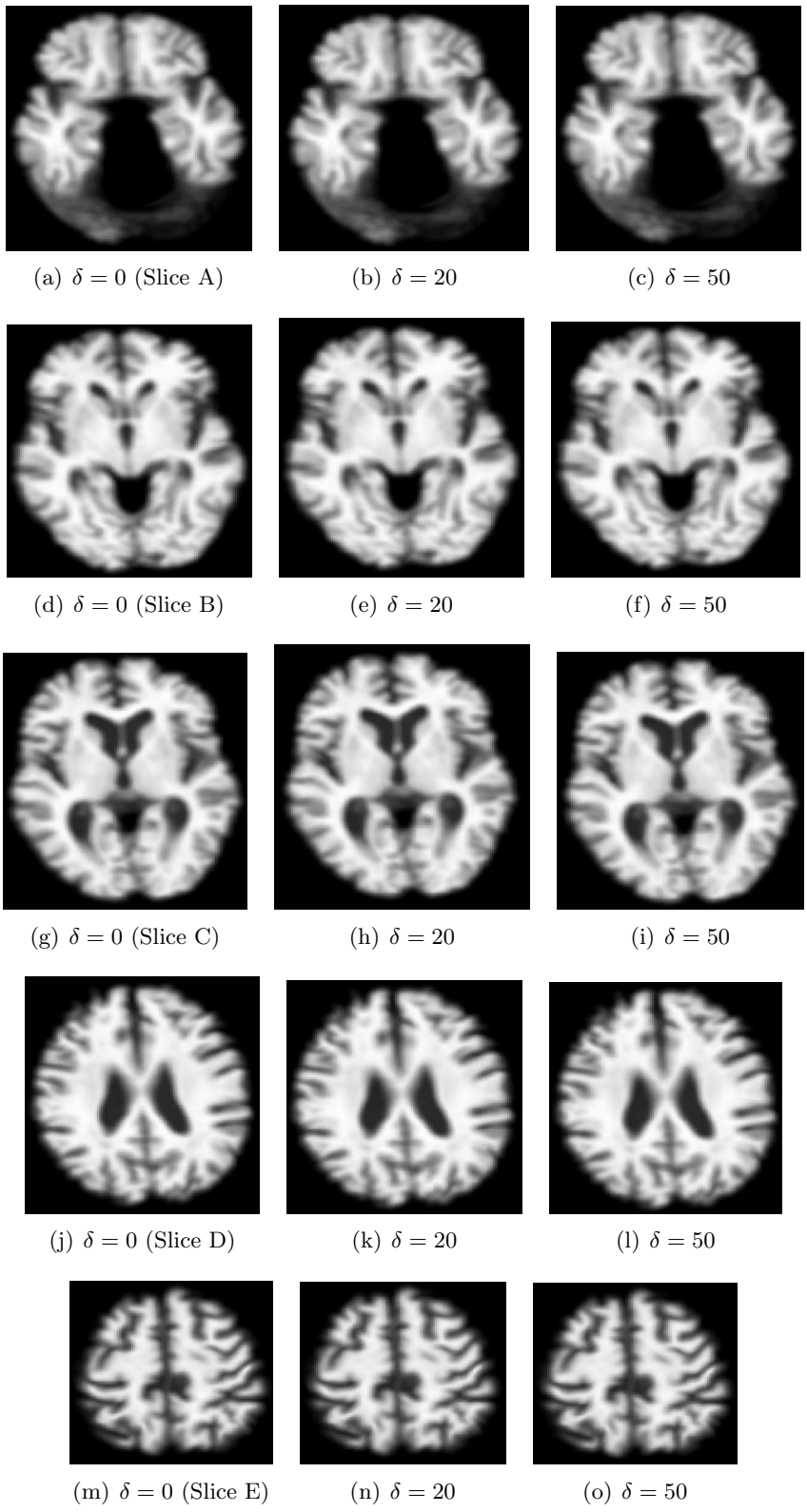
Table 5: Effect of the thresholding parameter (δ) used for octree construction for the example shown in Figure 12. The regularization parameter, γ, was set equal to 3000 in this experiment. We report the sum of the square of the mismatch between the fixed and registered images normalized by the sum of the square of the mismatch between the fixed and moving images. We also report the number of octants in the finest octree for each case.

δ	Relative SSD	Number of Elements
0	0.151	1.678×10^7
10	0.152	1.37×10^6
20	0.152	1.18×10^6
30	0.153	9.4×10^5
40	0.154	6.7×10^5
50	0.155	4.3×10^5

effect on the number of octants. This suggests that we can significantly reduce the computation time without sacrificing registration accuracy.

5.4 Parallel scalability

Finally, we tested the parallel scalability of our implementation on the Sun Constellation Linux Cluster "Ranger" at the Texas Advanced Computing Center (TACC). We report the results from the fixed-size scalability and isogranular scalability experiments in Tables 6 and 7, respectively. In the fixed-size scalability experiment, we track the execution time to solve a fixed problem on different processor counts. In the isogranular scalability experiment, we track the execution time while increasing the problem size and the number of processors proportionately.



(a) $\delta = 0$ (Slice A) (b) $\delta = 20$ (c) $\delta = 50$

(d) $\delta = 0$ (Slice B) (e) $\delta = 20$ (f) $\delta = 50$

(g) $\delta = 0$ (Slice C) (h) $\delta = 20$ (i) $\delta = 50$

(j) $\delta = 0$ (Slice D) (k) $\delta = 20$ (l) $\delta = 50$

(m) $\delta = 0$ (Slice E) (n) $\delta = 20$ (o) $\delta = 50$

Figure 15: Effect of the thresholding parameter (δ) used for octree construction for the example shown in Figure 12. Each row shows a z-plane crosssection of the registered (deformed moving) images for $\gamma = 3000$.

Table 6: Fixed-size scalability for the example shown in Figure 12 using $\gamma = 3000$ and $\delta = 50$. Approximately 4.3×10^5 elements were used in the finest grid. The time spent in evaluating the objective function is reported in the row labelled "Objective". The time spent in evaluating the gradient is reported in the row labelled "Gradient". Interpolation at the Gauss points were required to evaluate the objective function and the gradient. Interpolation at nodes were required to build the approximate Hessian using under-integration. The total time spent in the optimization function is reported in the row labelled "Gauss-Newton". The total time spent in the linear solves in each Newton iteration is reported in the row labelled "KSP solve". The time spent in the Matvecs for the elasticity operator is reported in the row labelled "Elas-Matvec"; this is only used to evaluate the objective and the gradient and not for the Hessian. The time spent in the Hessian Matvecs is reported in the row labelled "Hess-Matvec". The time required to update the Hessian is reported in the row labelled "Update-Hess"; this includes the time spent in interpolations. The time spent in building the image patches on each processor and gathering the image values from the Petsc DA ordering to the local ordering is reported in the row labelled "Build Patches". The time spent in setting up the multigrid solver is reported in the row labelled "MG-Setup"; this includes the time spent in coarsening and balancing all the coarser octrees for the multigrid and meshing for all the multigrid levels. The total runtime is the sum of the times spent in setting up the Multigrid, building the image patches and the Gauss-Newton iterations. These rows are printed in boldface. All timings are reported in seconds. This experiment was performed on the Teragrid system "Ranger" [61].

Processors	16	32	64	128	256
Objective	35.28	153.04	42.68	35.81	16.22
Gradient	28.37	20.12	10.27	8.66	6.85
Interpolation at Gauss points	861.60	606.03	273.44	157.15	83.38
Interpolation at Nodes	5.41	3.84	1.68	1.04	0.57
Gauss-Newton	**2.37e3**	**1.65e3**	**912.70**	**620.77**	**474.32**
KSP Solve	1.46e3	1.01e3	622.79	443.01	379.48
Elas-Matvec	43.2	62.4	42.17	28.75	16.45
Hess-Matvec	1.30e3	892.8	523.52	345.91	303.73
Update-Hess	654.30	462.77	212.53	126.68	70.31
Build Patches	**92.95**	**92.08**	**87.02**	**80.54**	**78.76**
MG-Setup	**3.02**	**3.77**	**3.53**	**2.92**	**5.03**

Table 7: Isogranular scalability for registering synthetic examples. The fixed and moving images for the first case ($N = 64$) were down-sampled versions of the images used in Figure 7. The higher resolution images were generated by repeating the lower resolution images. For example, the images for the $N = 128$ case were generated by joining 8 copies of the images for the $N = 64$ case. We set $\gamma = 10^4$ for all the cases. For this choice, the maximum and minimum determinants of jacobians of the deformations for each case was approximately 2.17 and 0.34, respectively. Reducing γ resulted in deformations with more extreme (maximums were higher and minimums were lower) determinants of jacobians. The row labelled "N" gives the number of voxels in each dimension of the images and the corresponding number of octants at the finest octree is reported in the row labelled "Octants". The time spent in evaluating the objective function is reported in the row labelled "Objective". The time spent in evaluating the gradient is reported in the row labelled "Gradient". Interpolation at the Gauss points were required to evaluate the objective function and the gradient. Interpolation at nodes were required to build the approximate Hessian using under-integration. The total time spent in the optimization function is reported in the row labelled "Gauss-Newton". The total time spent in the linear solves in each Newton iteration is reported in the row labelled "KSP solve". The time spent in the Matvecs for the elasticity operator is reported in the row labelled "Elas-Matvec"; this is only used to evaluate the objective and the gradient and not for the Hessian. The time spent in the Hessian Matvecs is reported in the row labelled "Hess-Matvec". The time required to update the Hessian is reported in the row labelled "Update-Hess"; this includes the time spent in interpolations. The time spent in building the image patches on each processor and gathering the image values from the Petsc DA ordering to the local ordering is reported in the row labelled "Build Patches". The time spent in setting up the multigrid solver is reported in the row labelled "MG-Setup"; this includes the time spent in coarsening and balancing all the coarser octrees for the multigrid and meshing for all the multigrid levels. The total runtime is the sum of the times spent in setting up the Multigrid, building the image patches and the Gauss-Newton iterations. These rows are printed in boldface. All timings are reported in seconds. This experiment was performed on the Teragrid system "Ranger" [61].

Processors	1	8	64	512	4096
N	64	128	256	512	1024
Octants	1.52×10^4	1.2×10^5	9.35×10^5	7.42×10^6	5.95×10^7
Objective	2.23	7.93	17.89	93.05	57.98
Gradient	6.18	12.11	9.51	9.97	5.48
Interpolation at Gauss points	143.34	288.12	274.46	356.45	193.31
Interpolation at Nodes	0.981	1.88	1.66	2.17	1.44
Gauss Newton	**340.0**	**896.19**	**860.51**	**1075.8**	**667.53**
KSP Solve	187.95	589.68	572.61	699.70	449.63
Elas-Matvec	3.43	7.90	15.80	50.41	29.64
Hess-Matvec	180.56	544.70	513.23	625.68	376.99
Update-Hess	108.78	216.74	205.88	269.62	149.09
Build Patches	**4.73**	**14.22**	**19.26**	**31.23**	**45.59**
MG-Setup	**0.375**	**1.07**	**2.09**	**10.24**	**25.64**

6 Summary

We presented a parallel algorithm for solving the elastic image registration problem using octrees. We used a Gauss-Newton optimization algorithm and solved the linear system that arises in each optimization iteration using a multigrid preconditioned Conjugate Gradient algorithm. We used standard restriction and prolongation operators in our multigrid scheme. We also used a multiscale optimization approach to speed up the computation and to escape local minima. We demonstrated the performance of the algorithm on synthetic as well as medical images.

References

[1] M. F. Adams, H.H. Bayraktar, T.M. Keaveny, and P. Papadopoulos. Ultrascalable implicit finite element analyses in solid mechanics with over a half a billion degrees of freedom. In *Proceedings of SC2004*, The SCxy Conference series in high performance networking and computing, Pittsburgh, Pennsylvania, 2004. ACM/IEEE.

[2] Mark Adams and James W. Demmel. Parallel multigrid solver for 3D unstructured finite element problems. In *Supercomputing '99: Proceedings of the 1999 ACM/IEEE conference on Supercomputing*, New York, NY, USA, 1999. ACM Press.

[3] Volkan Akcelik, George Biros, and Omar Ghattas. Parallel multiscale Gauss-Newton-Krylov methods for inverse wave propagation. In *SC '02: Proceedings of the 2002 IEEE/ACM conference on Supercomputing*. IEEE, 2002.

[4] Daniel Alexander, James Gee, and Ruzena Bajcsy. Elastic matching of diffusion tensor mris. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 244 – 249, 1999.

[5] Ignacio Arganda-Carreras, Carlos O.S. Sorzano, Roberto Marabini, Jose Maria Carazo, Carlos Ortiz de Solorzano, and Jan Kybic. Consistent and elastic registration of histological sections using vector-spline regularization. In *Computer Vision Approaches to Medical Image Analysis (CVAMIA)*, volume 4241 of *LNCS*, pages 85 – 95, 2006.

[6] Randolph E. Bank and Todd Dupont. An optimal order process for solving finite element equations. *Mathematics of Computation*, 36(153):35 – 51, 1981.

[7] R. Becker and M. Braack. Multigrid techniques for finite elements on locally refined meshes. *Numerical Linear Algebra with applications*, 7:363 – 379, 2000.

[8] B Bergen, F. Hulsemann, and U. Rude. Is 1.7×10^{10} unknowns the largest finite element system that can be solved today? In *SC '05: Proceedings of the 2005 ACM/IEEE conference on Supercomputing*, Washington, DC, USA, 2005. IEEE Computer Society.

[9] Martin Bohme, Rainer Hagenau, Jan Modersitzki, and Bodo Siebert. Non-linear image registration on PC-clusters using parallel FFT techniques. Technical report, University of Lubeck, 2002.

[10] Fred L. Bookstein. Principal warps: thin plate splines and the decomposition of deformations. *IEEE transactions on pattern analysis and machine intelligence*, 11(6):567 – 585, 1989.

[11] D. Braess and W. Hackbusch. A new convergence proof for the multigrid method including the V-cycle. *SIAM Journal on Numerical Analysis*, 20(5):967 – 975, 1983.

[12] Morten Bro-Nielsen and Claus Gramkow. Fast fluid registration of medical images. In *Visualization in Biomedical Computing (VBC'96)*, volume 1131 of *LNCS*, pages 267 – 276, 1996.

[13] Paul M. Campbell, Karen D. Devine, Joseph E. Flaherty, Luis G. Gervasio, and James D. Teresco. Dynamic octree load balancing using space-filling curves. Technical Report CS-03-01, Williams College Department of Computer Science, 2003.

[14] U. Clarenz, M. Droske, and M. Rumpf. Towards fast nonrigid registration. In *Inverse Problems, Image Analysis and Medical Imaging, AMS Special Session Interaction of Inverse Problems and Image Analysis*, pages 67 – 84. AMS, 2002.

[15] W.R. Crum, T. Hartkens, and D.L.G. Hill. Non-rigid image registration: theory and practice. *The British Journal of Radiology*, 77:S140 – S153, 2004.

[16] Malcolm H. Davis, Alireza Khotanzad, Duane P. Flamig, and Steven E. Harms. Coordinate transformation in 3d image matching by a physics based method-elastic body splines. In *International Symposium on Computer Vision*, pages 218 – 222, 1995.

[17] J. E. Dendy. Black box multigrid. *Journal of Computational Physics*, 48:366 – 386, 1982.

[18] Tarek A. El-Ghazawi, Prachya Chalermwat, and Jacqueline Le Moigne. Wavelet-based image registration on parallel computers. In *ACM/IEEE SC97 Conference*, 1997.

[19] Bernd Fischer and Jan Modersitzki. Fast diffusion registration. In *AMS Contemporary Mathematics, Inverse Problems, Image Analysis, and Medical Imaging*, volume 313, pages 117 – 129, 2002.

[20] Bernd Fischer and Jan Modersitzki. Curvature based image registration. *Journal of Mathematical Imaging and Vision*, 18:81 – 85, 2003.

[21] M. Fornefett, K. Rohr, and H.S. Stiehl. Radial basis functions with compact support for elastic registration of medical images. *Image and Vision Computing*, 19:87 – 96, 2001.

[22] Mike Fornefett, Karl Rohr, and H. Siegfried Stiehl. Elastic registration of medical images using radial basis functions with compact support. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 402 – 407, 1999.

[23] Ardeshir Goshtasby. Piecewise cubic mapping functions for image registration. *Pattern Recognition*, 20:525 – 533, 1987.

[24] Ardeshir Goshtasby. Image registration by local appoximation methods. *Image and Vision Computing*, 6(4):255 – 261, 1988.

[25] Lisa gottesfeld brown. A survey of image registration techniques. *ACM computing surveys*, 24(4):325 – 376, 1992.

[26] Eldad Haber and Stefan Heldmann. An octree multigrid method for quasi-static Maxwell's equations with highly discontinuous coefficients. *Journal of Computational Physics*, 223(2):783 – 796, 2007.

[27] Eldad Haber, Stefan Heldmann, and Jan Modersitzki. An octree method for parametric image registration. *SIAM journal on scientific computing*, 29(5):2008 – 2023, 2007.

[28] Eldad Haber, Stefan Heldmann, and Jan Modersitzki. Adaptive mesh refinement for nonparametric image registration. *SIAM journal on scientific computing*, 30(6):3012 – 3027, 2008.

[29] Eldad Haber and Jan Modersitzki. A multilevel method for image registration. *SIAM journal on scientific computing*, 27(5):1594 – 1607, 2006.

[30] Wolfgang Hackbusch. *Multigrid methods and applications*, volume 4 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, 1985.

[31] Stefan Henn and Kristian Witsch. Iterative multigrid regularization techniques for image matching. *SIAM journal on scientific computing*, 23(4):1077 – 1093, 2001.

[32] Derek L G Hill, Philipp G Batchelor, Mark Holden, and David J Hawkes. Medical image registration. *Physics in medicine and biology*, 46:R1 – R45, 2001.

[33] Lars Homke. A multigrid method for anisotropic pdes in elastic image registration. *Numerical Linear Algebra with Applications*, 13:215 – 229, 2006.

[34] Luis Ibanez, Will Schroeder, Lydia Ng, and Josh Cates. *The ITK Software Guide*. Kitware, Inc., second edition, 2005.

[35] Fumihiko Ino, Yasuhiro Kawasaki, Takahito Tashiro, Yoshikazu Nakajima, Yoshinobu Sato, Shinichi Tamura, and Kenichi Hagihara. A parallel implementation of 2-d/3-d image registration for computer-assisted surgery. In *11th International Conference on Parallel and Distributed Systems (ICPADS'05)*, 2005.

[36] Fumihiko Ino, Kanrou Ooyama, and Kenichi Hagihara. A data distributed parallel algorithm for nonrigid image registration. *Parallel Computing*, 31:19 – 43, 2005.

[37] El Mostafa Kalmoun, Harald Kostler, and Ulrich Rude. 3-d optical flow computation using a parallel variational multigrid scheme with application to cardiac c-arm ct motion. *Image and Vision Computing*, 25:1482 – 1494, 2007.

[38] Stefan Kruger and Andrew Calway. Image registration using multiresolution frequency domain correlation. In *British Machine Vision Conference*, pages 316 – 325, 1998.

[39] Jan Kybic and Michael Unser. Multiresolution spline warping for epi registration. In *in Proceedings of SPIE*, pages 571 – 579, 1999.

[40] Jan Kybic and Michael Unser. Fast parametric elastic image registration. *IEEE Transactions on Image Processing*, 12(11):1427 – 1442, 2003.

[41] Hava Lester and Simon R. Arridge. A survey of hierarchical non-linear medical image registration. *Pattern Recognition*, 32:129 – 149, 1999.

[42] J. B. Antoine Maintz and Max A. Viergever. A survey of medical image registration. *Medical Image Analysis*, 2(1):1 – 36, 1998.

[43] J. Modersitzki, G. Lustig, O. Schmitt, and W. Obeloer. Elastic registration of brain images on large pc-clusters. *Future generation computer systems*, 18:115 – 125, 2001.

[44] J. Modersitzki, W. ObelSer, O. Schmitt, and G. Lustig. Elastic matching of very large digital images on high performance clusters. In *Proceedings of the 7th International Conference on High-Performance Computing and Networking*, volume 1593 of *LNCS*, pages 141 – 149, 1999.

[45] Jan Modersitzki. *Numerical Methods for Image Registration*. Numerical Mathematics and Scientific Computation. Oxford Univ Press, 2004.

[46] Jacqueline Le Moigne, William J. Campbell, and Robert F. Cromp. An automated parallel image registration technique based on the correlation of wavelet features. *IEEE transactions on geoscience and remote sensing*, 40(8):1849 – 1864, 2002.

[47] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer series in operations research. Springer, 2006.

[48] Alexis Roche, Gregoire Malandain, Nicholas Ayache, and Sylvain Prima. Towards a better comprehension of similarity measures used in medical image registration. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, volume 1679 of *LNCS*, pages 555 – 567, 1999.

[49] K. Rohr, H.S. Stiehl, R. Sprengel, W. Beil, T.M. Buzug, J. Weese, and M.H. Kuhn. Point-based elastic registration of medical image data using approximating thin-plate splines. *Visualization in biomedical computing*, pages 297 – 306, 1996.

[50] Karl Rohr. Spline-based elastic image registration. *Proceedings in Applied Mathematics and Mechanics*, 3:36 – 39, 2003.

[51] Rahul S. Sampath, Santi S. Adavani, Hari Sundar, Ilya Lashuk, and George Biros. Dendro: Parallel algorithms for multigrid and AMR methods on 2:1 balanced octrees. In *SC '08: Proceedings of the 2008 IEEE/ACM conference on Supercomputing*. IEEE, 2008.

[52] Rahul S. Sampath and George Biros. A parallel geometric multigrid method for finite elements on octree meshes. Technical report, Georgia Institute of Technology, 2008. Submitted for publication.

[53] Oliver Schmitt, Jan Modersitzki, Stefan Heldmann, Stefan Wirtz, and Bernd Fischer. Image registration of sectioned brains. *International Journal of Computer Vision*, 73(1):5 – 39, 2007.

[54] Carlos O. S. Sorzano, Philippe Thevenaz, and Michael Unser. Elastic registration of biological images using vector-spline regularization. *IEEE transactions on biomedical engineering*, 52(4):652 – 663, 2005.

[55] Radu Stefanescu, Xavier Pennec, and Nicholas Ayache. Grid-enabled non-rigid registration of medical images. *Parallel processing letters*, 14(2):197 – 216, 2004.

[56] Hari Sundar, Christos Davatzikos, and George Biros. Biomechanically-constrained 4D estimation of myocardial motion. Technical report, University of Pennsylvania, 2008.

[57] Hari Sundar, Rahul S. Sampath, Santi S. Adavani, Christos Davatzikos, and George Biros. Low-constant parallel algorithms for finite element simulations using linear octrees. In *SC '07: Proceedings of the 2007 ACM/IEEE conference on Supercomputing*, New York, NY, USA, 2007. ACM Press.

[58] Hari Sundar, Rahul S. Sampath, and George Biros. Bottom-up construction and 2:1 balance refinement of linear octrees in parallel. *SIAM Journal on Scientific Computing*, 30(5):2675 – 2708, 2008.

[59] Richard Szeliski and Stephane Lavallee. Matching 3-d anatomical surfaces with nonrigid deformations using octree-splines. *International Journal of Computer Vision*, 18(2):171 – 186, 1996.

[60] Richard Szeliski and Heung-Yeung Shum. Motion estimation with quadtree splines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(12):1199 – 1210, 1996.

[61] TACC. Ranger's system architecture. http://www.tacc.utexas.edu.

[62] Mark P. Wachowiak and Terry M. Peters. Parallel optimization approaches for medical image registration. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, volume 3216 of *LNCS*, pages 781 – 788, Berlin Heidelberg, 2004. Springer-Verlag.

[63] Stefan Wirtz, Bernd Fischer, Jan Modersitzki, and Oliver Schmitt. Super-fast elastic registration of histologic images of a whole rat brain for three-dimensional reconstruction. *Medical Imaging*, 5370:328 – 334, 2004.

[64] Gert Wollny and Frithjof Kruggel. Computional cost of nonrigid registration algorithms based on fluid dynamics. *IEEE Transactions on Medical Imaging*, 21(8):946 – 952, 2002.

[65] Stefan Worz and Karl Rohr. Physics-based elastic registration using non-radial basis functions and including landmark localization uncertainties. *Computer Vision and Image Understanding*, 111:263 – 274, 2008.

[66] Harry Yserentant. On the convergence of multi-level methods for strongly nonuniform families of grids and any number of smoothing steps per level. *Computing*, 30:305 – 313, 1983.

[67] Barbara Zitova and Jan Flusser. Image registration methods: A survey. *Image and Vision Computing*, 21:977 – 1000, 2003.