SIAM J. SCI. COMPUT.
Vol. 30, No. 6, pp. 3012–3027

ADAPTIVE MESH REFINEMENT FOR NONPARAMETRIC IMAGE REGISTRATION*

ELDAD HABER†, STEFAN HELDMANN†, AND JAN MODERSITZKI‡

Abstract. Three-dimensional (3D) image registration is a computationally intensive problem which is commonly solved in medical imaging. The complexity of the problem stems from its size and nonlinearity. In this paper we present an approach that drastically reduces the problem size by using adaptive mesh refinement. Our approach requires special and careful discretization of the variational form on adaptive quad/octree grids. It further requires an appropriate refinement criterion. We show that in some cases this approach can reduce the computational time by a factor of approximately 10 or so in two dimensions and 5 in three dimensions compared to the nonadaptive approach.

Key words. registration, adaptive, octree

AMS subject classification. 65K10

DOI. 10.1137/070687724

1. Introduction.

Image registration is one of today's challenging image processing problems. Given a so-called reference image R and a so-called template image T, the objective is to find a *reasonable* transformation such that a transformed version of the template image becomes *similar* to the reference image. Image registration has to be applied whenever images resulting from different times, devices, and/or perspectives need to be compared or integrated; see, e.g., [8, 33, 14, 29, 32, 13, 22, 44, 35, 15] and references therein. In medical applications at least one of the images is typically obtained from an individual (patient), and it needs to be matched to another image of the same patient or to an atlas image.

Image registration involves three major challenges. The first challenge is to design an appropriate distance or similarity measure. For images of the same modality, the idea is to find a vector field u such that $T(x+u(x)) \approx R(x)$, and thus the L_2-norm of the difference is a common distance measure. For images of different modalities, specialized measures have been designed; see [38, 39, 25, 20]. The second challenge stems from the inherent ill-posedness of the problem [35]. Hence, regularization is inevitable. Parametric and nonparametric approaches are common. In the parametric approach, the transformation is restricted to a typically low or modest dimensional subspace spanned, for example, by rigid, affine linear, or spline based functions. The task is then to identify optimal expansion coefficients. For the nonparametric approach, an explicit regularizer or penalty for unwanted transformations is introduced; see [35] for an overview. Probably the most commonly used regularizer is the elastic potential; see, e.g., [7, 9, 35]. The idea is that transformations with a large elastic potential are considered to be less likely than those with a small elastic potential. More recent approaches aim to incorporate additional information in terms of constraints. From a modeling point of view, the nonparametric approach is the most powerful one. In

*Received by the editors April 9, 2007; accepted for publication (in revised form) June 16, 2008; published electronically October 13, 2008. This work was funded by NSF grants CCF-0728877, CCF-0427094, and DNS-0724759 and DOE grant DE-FG02-05ER25690.
http://www.siam.org/journals/sisc/30-6/68772.html
†Department of Mathematics and Computer Science, Emory University, Atlanta, GA 30322 (haber@mathcs.emory.edu, heldmann@mathcs.emory.edu).
‡Institute of Mathematics, University of Lübeck, 23560 Lübeck, Germany (modersit@math.uni-luebeck.de).

fact, other approaches might be considered as particular specifications of different regularizations; see [35]. The third challenge in image registration is provided by the computational complexity of the problem. Fast and efficient numerical schemes are crucial. This is especially the case for three-dimensional (3D) images, where tens or hundreds of millions of unknowns need to be evaluated. This problem can particularly cast a challenge in clinical applications when results are needed in minutes rather than hours or days.

In this paper, we address the third challenge. For ease of presentation, we focus on the L_2-norm as a distance measure and the elastic potential as a regularizer. However, it is important to note that the proposed concepts carry over to any differentiable distance measure and regularizer.

Several approaches toward fast implementations have been discussed in the literature: iterative solvers [9, 36], specialized direct solvers [12], fast filter techniques [42, 6], and multigrid [24, 23, 10, 27, 21]. All these techniques are combined with a multilevel strategy. However, they all use the original image grid as the finest grid. Already for moderate sized 3D images this results in a large number of degrees of freedom. For example, for 128^3 images, one has roughly 6 million unknowns. Thus, even a superfast implementation of a multilevel/multigrid method might be too slow in clinical application. In this paper we propose a strategy to reduce the size of the problem by using adaptive multilevel mesh refinement. The idea is hardly new for numerical methods for partial differential equations (PDEs); see [34, 11, 5] and references therein. Nevertheless, the use of adaptive meshing for inverse problems is a relatively new field with very few references; see [2, 3, 4]. To the best of our knowledge, it is completely new in the field of elastic image registration. Some relevant work on octree based image registration is in [40, 41, 28] and our recent work on parametric image registration [18]. In [41, 28] the displacement field was discretized using quadtree splines and in [40] a two-dimensional (2D) surface was embedded in three dimensions and represented using an octree. Other relevant contributions using octrees in image processing have been made in the field of computer graphics [31, 30]. In particular, the work of Losasso et al. on octree discretization demonstrates that images of fine detail, flows, and smoke can be represented efficiently and reliably with this type of data structure.

In this paper we derive a multilevel adaptive mesh refinement method for elastic image registration. We use octrees as a basic structure for the underlying displacement field and discretize the optimization problem on an octree. The goal is to represent a less complex transformation by a smaller number of unknowns. An extreme example is a translation or shift, where the complete transformation can be represented by only three unknowns. Note that the octree structure is used for the transformation, while we use the original high resolution representation for the given images. Further acceleration of the method proposed here can be obtained by using an image-pyramid structure. However, we choose to concentrate on the discretization of the transformation assuming a fixed image size.

The paper is structured as follows. In section 2 we present the adaptive approach taken. In section 3 we describe the discretization of the problem on an octree mesh. In section 4 we discuss how to solve the optimization problem. We explore briefly the limited memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) method [37] for the solution of the problem given a single octree grid. In section 5 we discuss refinement criteria to effectively solve the problem. In section 6 we carry out numerical experiments and demonstrate how an order of magnitude in computational time can be

saved. Finally, in section 7 we summarize the new approach.

2. The adaptive image registration approach. In this section we present the overall idea; details are given in the following sections. In monomodal image registration, the objective is to minimize the functional

$$J(u) = \frac{1}{2}\|T(u) - R\|^2_{L_2(\Omega)} + \frac{\alpha}{2}\|Bu\|^2_{L_2(\Omega)}, \tag{2.1}$$

where Ω is the underlying data domain (for ease of presentation $\Omega =]0,1[^3$), the transformed image is $T(u)(x) := T(x + u(x))$, B is a differential operator related to the regularizer, and $\alpha > 0$ is a regularization parameter; see, e.g., [35]. In general there is no analytic solution for this problem, and we rely on numerical optimization schemes. Here, we first discretize the functional and then optimize using a quasi-Newton method [37].

In standard approaches, J and the displacement field u are discretized on the voxels of the underlying images. Therefore, a standard discretization [21] in three dimensions on a regular rectangular grid Ω_h with $n = n_1 \times n_2 \times n_3$ cells (voxels) and uniform cell width (voxel size) $h = (h_1, h_2, h_3)$ yields

$$J^h(u^h) = \frac{1}{2}\|T(u^h) - R\|^2_2 + \frac{\alpha}{2}(u^h)^\top A^h u^h, \tag{2.2}$$

where $u^h = [u^h_1, \dots, u^h_3]^\top$ is a vector collecting the displacements for all voxel locations $x^h \in \Omega_h$, R is the vector $R(x^h)$, and $A^h = (B^h)^\top B^h$, where B^h is a discretization of B; see [21] for details. Here, we use the elastic potential with $B = (\mu^{1/2} I_3 \otimes \nabla\ ,\ \lambda^{1/2} \nabla\cdot)$ such that

$$\|Bu\|^2_{L_2(\Omega} = \mu \sum_j \|\nabla u_j\|^2_{L_2(\Omega)} + \lambda \|\nabla \cdot u\|^2_{L_2(\Omega)},$$

and thus A^h is a discrete version of the Navier–Lamé operator,

$$A^h = \mu \vec{\Delta}^h + \lambda \nabla^h \nabla^h \cdot,$$

with Lamé constants λ and μ (here we take the common choice $\mu = 1$, $\lambda = 1$), and $\vec{\Delta}^h$ is the vector Laplacian.

The time consuming part in registration is the solution of the $3n$ Euler–Lagrange equations which arise from the minimization of (2.2). The idea here is to use an adaptive sparse grid S_h with fewer grid points than Ω_h for the discretization u^h of u in order to reduce the number of unknowns and thus the computational cost. Since the image grid does not necessarily coincide with the transformation grid, we construct a linear interpolation operator Q that maps u^h from the sparse grid, S_h, to the image grid, Ω_h. The matrix Q is typically sparse, and nonsquare with only a few nonzero entries per row. In this paper, for the sake of simplicity, we assume that the image grid does not reduce the information in the image. This assumption is realistic even for relatively coarse meshes (see [18]). The new objective function is thus

$$J^h(u^h) = \frac{1}{2}\|T(Qu^h) - R\|^2_2 + \frac{\alpha}{2}(u^h)^\top A^h u^h. \tag{2.3}$$

Note that now $A^h = (B^h)^\top B^h$, where B^h is a discretization of B on the sparse, in general, nonregular grid S_h.

Algorithm 1 Adaptive Image Registration

1. choose initial grid S_h and initial guess u_0^h
2. create Q, A_h for the sparse grid S_h
3. find u_*^h minimizing (2.3) based on the starting value u_0^h
4. if $\|u_0^h - u_*^h\| < \text{tol}$ then stop
5. refine S_h and interpolate u_*^h on the refined grid to obtain a starting guess for the refined grid; goto 2

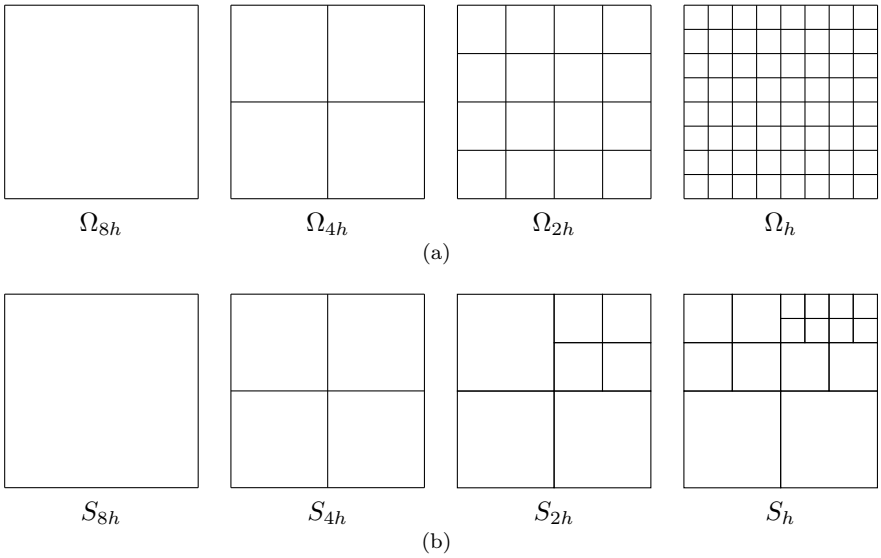

FIG. 1. *2D example for grid refinement:* (a) *regular;* (b) *adaptive.*

Our adaptive scheme is summarized in Algorithm 1; details are given in the following sections.

A naive concretization of Algorithm 1 would be to start on a very coarse grid and refine the grid by just doubling the points in each direction; see Figure 1(a). This is related to the standard multilevel approach. A drawback is that one will finally end up with the fine data grid Ω_h, where a fine grid is used even in regions where the transformation is more or less constant. As a remedy, we use octrees. From our point of view, this choice is quite natural, since an octree grid S_h is nested in the finest regular grid Ω_h and still relates to the pixel structure of discrete images; see Figure 1(b) for a simple example of a sequence of refined sparse grids. A more detailed description of octrees is presented in section 3.

In the next section, we explain the octree data structure and the discretization of the regularizer on a particular octree (step 2 in Algorithm 1). Section 4 explains the optimization technique for a particular discretization and also how to solve the arising linear systems (step 3 in Algorithm 1). Finally, in section 5, we explain how to refine the octree (step 5 in Algorithm 1).

3. Octree data structure and discretization. In this section we discuss octree based discretization of the image registration problem. Following [1], we envision a uniform underlying coarse grid Ω_H with cell width H and a uniform underlying fine grid Ω_h with cell width h; see Figures 1(b) and 1(a). We assume that $H = 2^L h$,

where L denotes the total number of refinement levels. The fine grid is basically the voxel grid of the images, and the coarse grid is inexpensive to work on while still producing a meaningful coarse grid solution that can serve as a starting guess for a refined level.

3.1. Octree data structure. In contrast to the regular grids, the octree grid S_h is composed of square cells of different sizes. Each of these cells can have a width $2^j h$, where $0 \le j \le L$. Thus, S_h is nested in Ω_h. To make the data structure easier to access, we limit the ratio of widths of adjacent cells by two. This results in a tree structure, where each node (cell) has up to eight children in three dimensions and four for the 2D case; see Figure 1(b) for an example.

The grid structure is then stored as a sparse array. The size of each cell is stored in the upper left corner of the array. This allows us to quickly find neighbors, which is a major operation in the computational process. This data structure is closely related to the one suggested in [26]. For example, for the sparse grid S_{2h} presented in Figure 1, the nonzero entries are stored as

$S_{2h} =$

4				2		2	
				2		2	
4				4			

.

3.2. Discretization of the regularization operator. Given a particular octree grid one has to decide where to discretize the different variables. In our previous work [21] we have used staggered grids in order to discretize $u = (u_1, u_2, u_3)$. In the context of octree discretization and due to the discretization of derivatives in the tangential directions, a second order staggered grid discretization is possible but difficult to obtain; see [17]. In this work, we have therefore chosen a nodal grid based discretization which implies that all variables are discretized at the nodes. While this discretization is not optimal from a multigrid perspective, it is substantially simpler to work with and implement, and second order accuracy can be easily obtained even on octrees. For ease of presentation, we derive our discretizations in two dimensions; the 3D extension is straightforward.

3.2.1. Discretizing the gradient. We focus on an arbitrary component u_ℓ of the displacement. Consider the quadtree (2D octree) cell depicted in Figure 2 with cell-center $\bullet$ at position (x_1, x_2) and cell-width $2h$.

In the nodal discretization all the components of u are discretized on the nodes. The partial derivatives are thus naturally discretized to second order accuracy along the centers of the edges of each cell, i.e.,

$$\partial_1 u_\ell(x_1, x_2) = \partial_1^h u_\ell(x_1, x_2) + \mathcal{O}(h^2) \quad \text{and} \quad \partial_2 u_\ell(x_1, x_2) = \partial_2^h u_\ell(x_1, x_2) + \mathcal{O}(h^2),$$

with ∂_j^h the standard central finite difference approximation and $\ell = 1, 2$. Thus, for

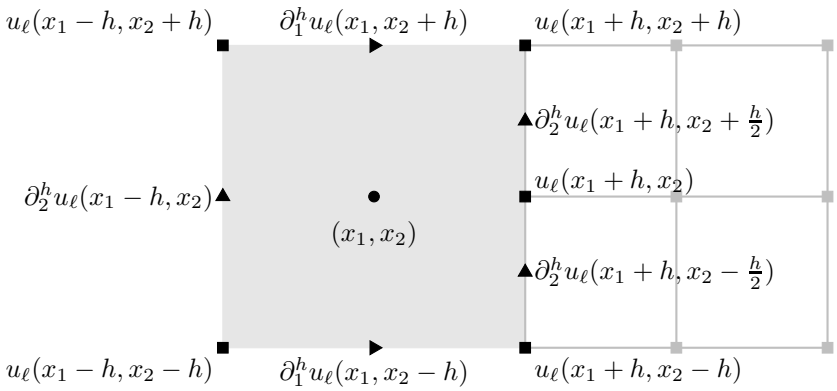

FIG. 2. *Discretization of* ∇u_ℓ.

the quadtree in Figure 2 we obtain the second order approximations

$$\begin{aligned}
\partial_1^h u_\ell(x_1,x_2+h) &= \frac{u_\ell(x_1+h,x_2+h)-u_\ell(x_1-h,x_2+h)}{2h},\\
\partial_1^h u_\ell(x_1,x_2-h) &= \frac{u_\ell(x_1+h,x_2-h)-u_\ell(x_1-h,x_2-h)}{2h},\\
\partial_2^h u_\ell(x_1-h,x_2) &= \frac{u_\ell(x_1-h,x_2+h)-u_\ell(x_1-h,x_2-h)}{2h},\\
\partial_2^h u_\ell(x_1+h,x_2-h/2) &= \frac{u_\ell(x_1-h,x_2)-u_\ell(x_1-h,x_2-h)}{h},\\
\partial_2^h u_\ell(x_1+h,x_2+h/2) &= \frac{u_\ell(x_1-h,x_2+h)-u_\ell(x_1-h,x_2)}{h}.
\end{aligned}$$

Using this second order difference scheme, we can discretize the gradient of u_ℓ on the quadtree edges.

We now show how to use this approximation in order to discretize the regularization operator. To this end we write

$$\int_\Omega |\nabla u_\ell|^2\,dx = \sum_j \int_{\text{cell}_j} |\nabla u_\ell|^2\,dx = \sum_j \int_{\text{cell}_j} |\partial_1 u_\ell|^2\,dx + \sum_j \int_{\text{cell}_j} |\partial_2 u_\ell|^2\,dx.$$

Using the midpoint quadrature rule we approximate the integral over each cell, which yields a second order approximation to the integral. In the case of the above 2D example, with $v_j = \text{volume}(\text{cell}_j) = (2h)^2$ the cell's volume, we obtain

$$\int_{\text{cell}_j} |\partial_1 u_\ell|^2\,dx = \frac{v_j}{2}\Big[\partial_1^h u_\ell(x_1,x_2-h)\Big]^2 + \frac{v_j}{2}\Big[\partial_1^h u_\ell(x_1,x_2+h)\Big]^2 + \mathcal{O}(h^2), \tag{3.1a}$$

$$\begin{aligned}
\int_{\text{cell}_j} |\partial_2 u_\ell|^2\,dx &= \frac{v_j}{4}\Big[\partial_2^h u_\ell(x_1+h,x_2-\tfrac{h}{2})\Big]^2 + \frac{v_j}{4}\Big[\partial_2^h u_\ell(x_1+h,x_2+\tfrac{h}{2})\Big]^2\\
&\quad + \frac{v_j}{2}\Big[\partial_2^h u_\ell(x_1-h,x_2)\Big]^2 + \mathcal{O}(h^2).
\end{aligned} \tag{3.1b}$$

Summing over all of the cells, we hence obtain an $\mathcal{O}(h^2)$ approximation to the integral $\int_\Omega |\operatorname{grad} u_\ell|^2\,dx$ and therefore to $\int_\Omega |\operatorname{grad} u|^2\,dx$.

For ease of presentation, we derive a matrix representation for the discrete gradient operator. Let $\odot$ denote the Hadamard (or elementwise) product, $\nabla^h = [D_1^h, D_2^h]^\top$, with $D_j^h u_\ell^h$ the collection of $\partial_j^h u_\ell(x_1, x_2)$ for all discretization points, v the vector collecting all cell volumes, and A_e^c an average matrix from edges to cell-center of each cell; we have

$$\begin{aligned}\int_\Omega |\nabla u_\ell|^2 \, dx &= \int_\Omega |\partial_1 u_\ell|^2 + |\partial_2 u_\ell|^2 dx \\ &= v^\top A_e^c \Big((\nabla^h u_\ell^h) \odot (\nabla^h u_\ell^h) \Big) + \mathcal{O}(h^2) \\ &= (u_\ell^h)^\top (\nabla^h)^\top \operatorname{diag}[(A_e^c)^\top v] \nabla^h u_\ell^h + \mathcal{O}(h^2). \end{aligned} \tag{3.2}$$

Note that the regularization is quadratic in u_ℓ^h with a symmetric positive semidefinite matrix $(\nabla^h)^\top \operatorname{diag}[(A_e^c)^\top v] \nabla^h$ whose size is the number of nodes in the quad/octree mesh. The diagonal weighting matrix $\operatorname{diag}[(A_e^c)^\top v]$ handles the different cell volumes as well as the averaging from edges to cell-centers.

3.2.2. Discretizing the divergence. To discretize $\nabla \cdot u$ at cell-centered points, we again average the second order discretization of the derivatives to the cell-center. For the quadtree presented in Figure 2 we obtain

$$\begin{aligned}\partial_1 u_1(x_1, x_2) &= \tfrac{1}{2}\partial_1^h u_1(x_1, x_2 + h) + \tfrac{1}{2}\partial_1^h u_1(x_1, x_2 - h) + \mathcal{O}(h^2), \\ \partial_2 u_2(x_1, x_2) &= \tfrac{1}{4}\partial_2^{h/2} u_2(x_1 + h, x_2 - h/2) + \tfrac{1}{4}\partial_2^{h/2} u_2(x_1 + h, x_2 + h/2) \\ &\quad + \tfrac{1}{2}\partial_2^h u_2(x_1 - h, x_2) + \mathcal{O}(h^2) \\ &= \tfrac{1}{2}\partial_2^h u_2(x_1 - h, x_2) + \tfrac{1}{2}\partial_2^h u_2(x_1 + h, x_2) + \mathcal{O}(h^2),\end{aligned}$$

and hence

$$\begin{aligned}\int_{\text{cell}_j} (\nabla \cdot u)^2 \, dx = \frac{v_j}{2} \Big(&\partial_1^h u_1(x_1, x_2 + h) + \partial_1^h u_1(x_1, x_2 - h) \\ &+ \partial_2^h u_2(x_1 - h, x_2) + \partial_2^h u_2(x_1 + h, x_2) \Big) + \mathcal{O}(h^2).\end{aligned} \tag{3.3}$$

Using the notation $(\nabla^h \cdot)$ for the discretized divergence, we end up with the following approximation:

$$\begin{aligned}\int_\Omega (\nabla \cdot u)^2 \, dx &= v^\top (\nabla^h \cdot u^h) \odot (\nabla^h \cdot u^h) + \mathcal{O}(h^2) \\ &= (u^h)^\top (\nabla^h \cdot)^\top \operatorname{diag}(v) (\nabla^h \cdot) u^h + \mathcal{O}(h^2).\end{aligned} \tag{3.4}$$

3.3. The discrete regularizer. Summarizing the previous subsections, the discretized regularizer is

$$\frac{1}{2}\|Bu\|_{L_2(\Omega}^2 = \frac{\mu}{2} \sum_j \|\nabla u_j\|_{L_2(\Omega)}^2 + \frac{\lambda}{2} \|\nabla \cdot u\|_{L_2(\Omega}^2 = \frac{1}{2}(u^h)^\top A^h u^h + \mathcal{O}(h^2),$$

where A^h is

$$A^h = \mu I_d \otimes \Big[(\nabla^h)^\top \operatorname{diag}[(A_e^c)^\top v] \nabla^h \Big] + \lambda (\nabla^h \cdot)^\top \operatorname{diag}(v) (\nabla^h \cdot). \tag{3.5}$$

4. Solving the optimization problem. Since we use standard optimization techniques with implementation details similar to those in previous work [21], we only briefly summarize the strategy.

We implemented the L-BFGS method as suggested in [37]. Since the optimization scheme is applied for a fixed refinement level, for ease of presentation, we drop the subscript h in this section. Our goal is to minimize the discrete objective function

$$J(u) = \frac{1}{2}\|T(Qu) - R\|_2^2 + \frac{\alpha}{2}u^\top Au. \tag{4.1}$$

Any gradient descent direction requires the computation of the gradient of the objective function. Differentiating the different components with respect to u yields the Euler–Lagrange equation

$$\nabla J = Q^\top T_u^\top (T - R) + \alpha Au = 0,$$

where T_u is the Jacobian of T with respect to u; see [19, 21]. Though the Jacobian is a sparse matrix, the nonzero entries can vary by order of magnitudes.

For the L-BFGS method we build an approximation to the inverse of the Hessian of the objective function by using the most recent L directions $\{u^{(k-L)}, \dots, u^{(k)}\}$, the gradients $\{\operatorname{grad} J^{(k-L)}, \dots, \operatorname{grad} J^{(k)}\}$, and an initial approximation to the Hessian; see [37, 16] for implementation details. As discussed in [16], it is crucial to initialize the approximation to the Hessian with the Hessian of the regularizer. Thus, each iteration requires solving a linear system with the matrix A.

For an efficient solution of the linear system, iterative methods are required. In particular, a multigrid method can be applied (see [17]). To this end, the discretization is required to be h-elliptic (see [43] for details). While the analysis of multigrid for the octree discretization of the Navier–Lamé operator is beyond the scope of this paper, we note that multigrid methods have been successfully applied to nodal discretization of such systems [43].

5. Adaptive mesh refinement. The cost of the optimization process is directly impacted by the size of the problem and the initial guess for the solution. Adaptive multilevel refinement methods are targeted to achieve a low-cost good starting guess by using coarse grids, and to reduce the size of the discrete fine grid problem by using adaptive nested grids that refine only in areas where the error in the solution is large. Unfortunately, finding a unique refinement criterion that works for different problems is rather difficult; see, e.g., [43].

We next develop a refinement criterion. The basic idea is bounding the discretization error of the underlying continuous optimization problem and the objective functional J, respectively. Let S_h be a given octree discretization with cells $\Omega_1, \dots, \Omega_n$. Then J can be written as a sum over the octree cells, i.e.,

$$J(u) = \frac{1}{2}\sum_j \int_{\Omega_j} \Big(T(u) - R\Big)^2 + \alpha|Bu|^2\, dx.$$

In our discretization we approximate the integrals over the octree cells Ω_j by the midpoint rule. For the derivation of an error estimate let

$$\rho(x) := \Big(T(u(x)) - R\Big)^2 + \alpha|Bu(x)|^2$$

such that $J(u) = \frac{1}{2}\sum_j \int_{\Omega_j} \rho \, dx$, and let $\Omega_j = \{x : \|x - x_j\|_\infty < h_j\} \subset \mathbb{R}^d$ be an octree cell with cell-center x_j and width h_j. Using a first order Taylor expansion of ρ, we obtain

$$\int_{\Omega_j} \rho(x)\, dx = v_j \rho(x_j) + \int_{\Omega_j} \operatorname{grad} \rho(\xi(x))^\top (x - x_j)\ dx,$$

where $\xi(x)$ is a point in Ω_j and $v_j := h_j^d$ is the volume of the cell in d dimensions. Thus, the discretization error is bounded by

$$\left| \int_{\Omega_j} \rho \ dx - v_j \rho(x_j) \right| \leq v_j \sup_{\xi \in \Omega_j} |\nabla \rho(\xi)| \sup_{x \in \Omega_j} |x - x_j| = \frac{\sqrt{d}}{2} h_j v_j \sup_{\xi \in \Omega_j} |\nabla \rho(\xi)|.$$

Therefore, if $|\operatorname{grad} \rho|$ is large compared to the cell-width h_j, the approximation is inaccurate. Since the solution of the optimization problem depends on an accurate discretization of the integral, we want to refine in areas where the error is large.

Clearly, we cannot evaluate the supremum exactly. To this end we use the quantities for which we already have an approximation $\rho_j^h \approx \rho(x_j)$ and subsequently an approximation $\nabla^h \rho_j^h$ to the gradient $\nabla \rho(x_j)$ using finite differences. Then, in areas where $|\operatorname{grad}^h \rho_j^h|$ is large, we may assume that the grid should be finer, while in areas where the approximation to ρ is relatively flat, no further refinement is needed.

In order to decide if $|\operatorname{grad}^h \rho_j^h|$ is large, we use a parameter τ and refine every cell that satisfies $|\operatorname{grad}^h \rho_j^h| > \tau$. The refinement process is terminated when $|\operatorname{grad}^h \rho_j^h| \leq \tau$ for all cells. The choice of τ is, in general, difficult. Nevertheless, a simple strategy can help to determine an appropriate value. If τ is set to be large first, then a coarse grid is obtained. One can then observe the misfit and decide if τ should be decreased. If τ is decreased but without a significant improvement in the registration quality, then we conclude that this τ is sufficiently small.

6. Numerical experiments. To demonstrate our method, we present two 128^2 2D examples and a 128^3 3D example. The goal of the experiments is to investigate different aspects of our algorithms and compare octree to standard multilevel methods.

The general setup for each test case is as follows. We performed registrations for various tolerances τ in our refinement criteria. For the underlying image domain we considered the unit cube $[0,1]^d$ and the images were scaled to a gray-value range of $[0,1]$. The tolerances τ were chosen between 0 (refine everywhere) to 10. In all experiments we started our multilevel method on a coarsest mesh consisting of a single cell yielding 2×4 and 3×8 unknowns in two dimensions and three dimensions, respectively. The stopping criterion for the optimization on a single level was when the maximum difference of consecutive iterates was below 0.1 voxel/pixel width. The linear systems were solved to a precision of 10^{-5} using a preconditioned conjugate gradient method.

In two dimensions, we considered an academic and a real data example. In the academic example we registered a square to a "flag"; cf. Figure 3. The real data example is about the registration of two X-ray images of hands; cf. Figure 4. Both images have a size of 128×128 pixels.

In three dimensions, we registered two CT data sets of the chest under maximal inspiration and expiration, respectively; cf. Figure 5. Both images have 128^3 voxels.

The results are summarized in Tables 1, 2, and 3. For each experiment we give the number of cells on the finest level, the misfit of the images, i.e., $\|T(u_{\text{opt}}) - R\| / \|T - R\|$,

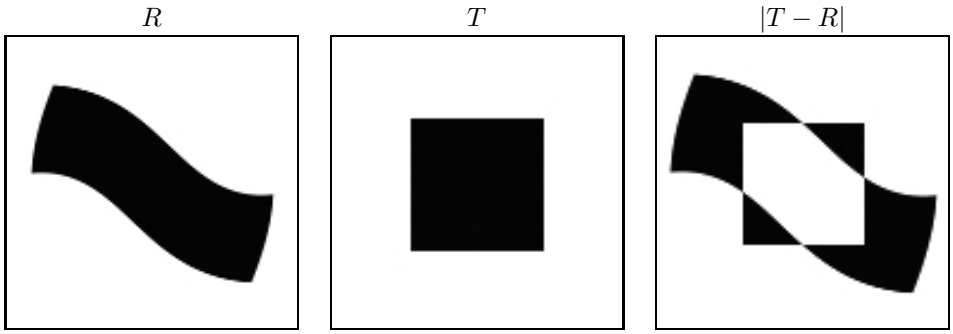

FIG. 3. *Images for the 2D flag example.*

FIG. 4. *Images for the 2D hand example.*

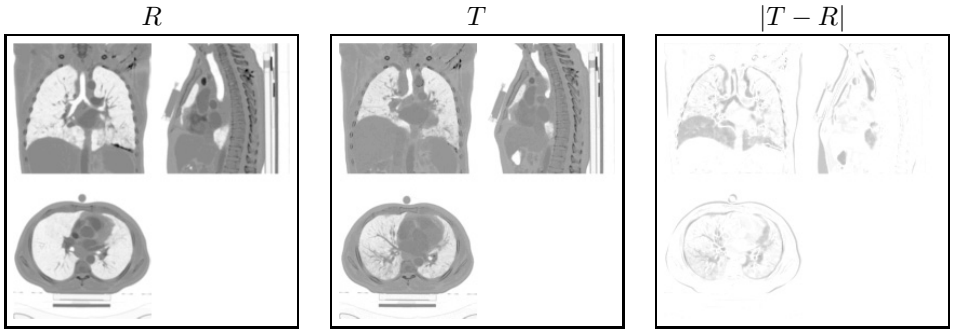

FIG. 5. *Images for the 3D CT example (volumes orthogonally sliced).*

and the speedup to a regular refinement ($\tau = 0$). To provide a fair measurement of the speedup, we compared the execution times to the execution time for the regular refinement without meshing; i.e., we compared the total time to the time spent only for the optimization (Algorithm 1, step 3) in the experiment with $\tau = 0$.

As observed in Tables 1 and 2, major computational savings can be obtained as τ increases. These computational savings do not come without cost; coarser meshes yield worse matching between the images. For example, for the flag example, using 103 cells we obtain a misfit of 15%, while using the full grid we obtain a misfit of 7%. Thus, the adaptive mesh adds regularization to the process. Although the misfit for $\tau = 10$ is larger than the misfit for $\tau = 0$ (full grid), the results of the registration

TABLE 1

Results for the 2D flag example. We observe a major speedup with only a slight increase of the misfit.

			Elapsed time				
τ	#cells	misfit	total	interp.	solving	meshing	speedup
0.00	16384	0.18%	78.1s	4.4s	67.5s	0.9s	1.0
0.10	1084	0.16%	19.3s	4.6s	10.5s	0.2s	3.9
0.20	772	0.23%	13.6s	4.4s	5.3s	0.2s	5.5
0.30	769	0.16%	15.3s	4.6s	6.8s	0.3s	4.9
0.40	556	0.14%	14.1s	4.8s	5.2s	0.3s	5.3
0.50	505	0.19%	12.9s	4.6s	4.3s	0.3s	5.8
1.00	376	0.21%	11.7s	4.4s	3.4s	0.2s	6.4
5.00	169	0.21%	9.2s	4.0s	1.9s	0.2s	8.2
10.00	139	0.39%	8.6s	4.0s	1.4s	0.2s	8.7

TABLE 2

Results for the 2D hand example. A factor of 12.5 *speedup is observed. The misfit is increased, but visually the images are still very similar.*

			Elapsed time				
τ	#cells	misfit	total	interp.	solving	meshing	speedup
0.00	16384	6.99%	70.3s	4.0s	59.6s	0.9s	1.0
0.10	1282	7.60%	11.5s	4.0s	3.4s	0.3s	5.8
0.20	886	8.45%	10.1s	3.8s	2.2s	0.3s	6.7
0.30	700	8.65%	9.5s	4.0s	2.0s	0.2s	7.0
0.40	613	8.92%	8.8s	3.9s	1.4s	0.2s	7.6
0.50	574	9.31%	8.3s	3.7s	1.2s	0.2s	8.0
1.00	409	10.10%	7.7s	3.3s	0.9s	0.2s	8.8
5.00	148	13.24%	6.6s	3.3s	0.5s	0.2s	10.2
10.00	103	15.45%	5.4s	2.4s	0.4s	0.1s	12.5

TABLE 3

Results for the 3D CT example. A factor of 5 *speedup is observed with an increase in the misfit. Visually, we could not see the difference between the* 6.96% *and the* 12.06%.

			Elapsed time				
τ	#cells	misfit	total	interp.	solving	meshing	speedup
0.00	2097152	6.96%	17933.9s	2637.3s	12904.1s	27.9s	0.9
0.50	83140	6.43%	10461.4s	2824.2s	6005.6s	10.2s	1.6
1.00	47216	7.89%	5355.3s	2650.3s	1270.1s	8.1s	3.1
5.00	17165	7.65%	5000.4s	2624.2s	975.5s	4.6s	3.4
10.00	16038	8.47%	4443.2s	2506.1s	582.5s	4.0s	3.8
50.00	6385	10.01%	4144.1s	2513.8s	268.7s	2.1s	4.1
100.00	3935	10.55%	3976.4s	2450.1s	179.6s	1.7s	4.2
200.00	2794	11.33%	3642.0s	2290.5s	99.2s	1.4s	4.6
500.00	2017	12.08%	3559.6s	2254.5s	63.2s	1.3s	4.7

look almost identical. This is demonstrated in Figures 6, 7, and 8, where we present the final deformed images for different values of τ.

From a computational point of view, our results show that the proposed method scales well with the grid tolerance, and in general we found an acceleration factor of 6–10 in two dimensions and a speedup of approximately 5 in three dimensions. Furthermore, we observed that the overhead for the adaptive grid refinement (meshing time in the table) was in general less than 0.1% of the execution time in two dimensions and 0.2–0.3% in three dimensions.

7. Summary. In this paper we have developed an adaptive multilevel refinement (AMR) method for nonparametric image registration. We have used the elastic

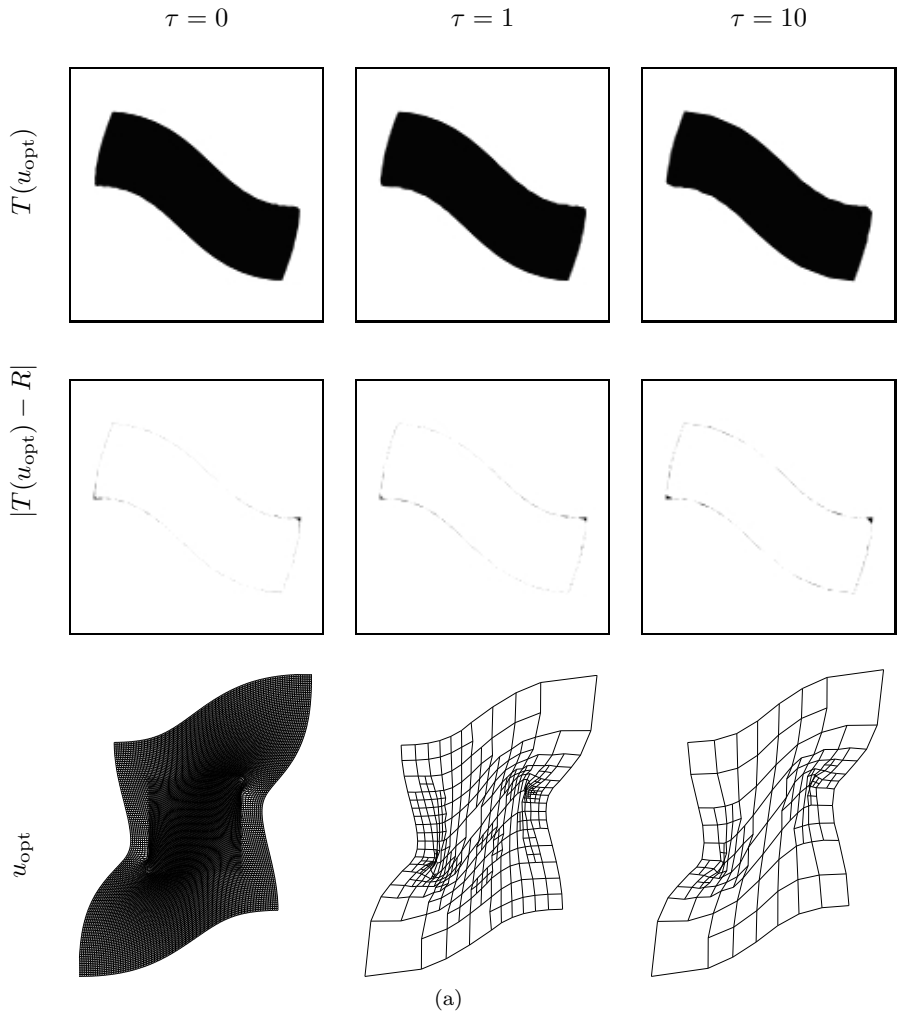

FIG. 6. *Results for the flag example. The recovered images (top), the difference between T and R (middle), and the quadtree mesh (bottom) for different values of τ.*

potential as a regularizer and discussed how to effectively and accurately discretize and solve the problem on octree grids. We have used the L-BFGS method for optimization which requires the solution of a linear system which evolves from the Hessian of the regularizer. We develop a refinement criterion based on the accurate evaluation of the variational form. Numerical experiments demonstrate that we can obtain a substantial speedup and reduction in problem size by using our method.

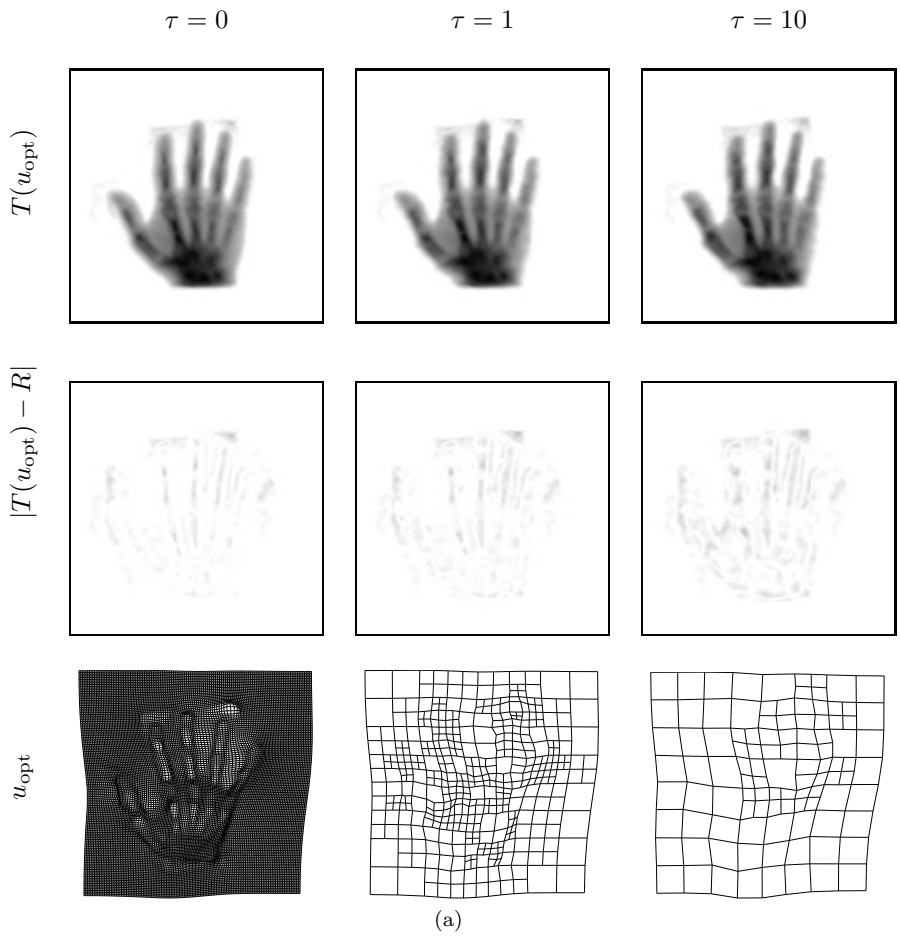

FIG. 7. *Results for the hand example. The recovered images (top), the difference between T and R (middle), and the quadtree mesh (bottom) for different values of τ.*

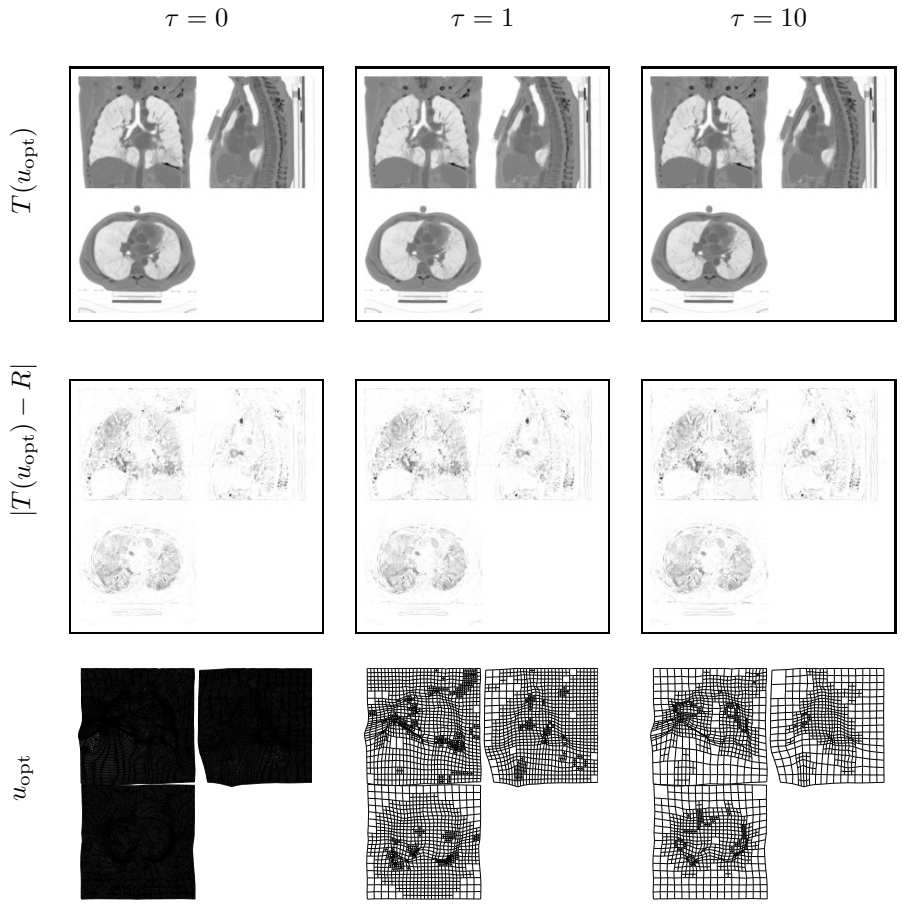

FIG. 8. *Results for the 3D CT example. A middle slice from the recovered images (top), the difference between T and R (middle), and the quadtree mesh (bottom) for different values of τ.*

REFERENCES

[1] U. Ascher and E. Haber, *Grid refinement and scaling for distributed parameter estimation problems*, Inverse Problems, 17 (2001), pp. 571–590.

[2] W. Bangerth, *Adaptive Finite Element Methods for the Identification of Distributed Coefficients in Partial Differential Equations*, Ph.D. thesis, University of Heidelberg, Heidelberg, Germany, 2002.

[3] R. Becker, *Adaptive Finite Element Methods for Optimal Control Problems*, Habilitation Thesis, University of Heidelberg, Heidelberg, Germany, 2001.

[4] R. Becker, H. Kapp, and R. Rannacher, *Adaptive finite element methods for optimal control of partial differential equations*, SIAM J. Control Optim., 39 (2000), pp. 113–132.

[5] M. Berger, *Adaptive Mesh Refinement for Time-Dependent Partial Differential Equations*, Ph.D. dissertation, Computer Science Report STAN-CS-82-924, Stanford University, Stanford, CA, 1982.

[6] M. Bro-Nielsen and C. Gramkow, *Fast fluid registration of medical images*, in Proceedings of the 4th International Conference on Visualization in Biochemical Computing, Lecture Notes in Comput. Sci. 1131, Springer-Verlag, London, 1996, pp. 267–276.

[7] C. Broit, *Optimal registration of deformed images*, Ph.D. thesis, Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA, 1981.

[8] L. G. Brown, *A survey of image registration techniques*, Surveys, 24 (1992), pp. 325–376.

[9] G. E. Christensen, *Deformable shape models for anatomy*, Ph.D. thesis, Sever Institute of Technology, Washington University in St. Louis, St. Louis, MO, 1994.

[10] U. Clarenz, M. Droske, and M. Rumpf, *Towards fast non–rigid registration*, in Inverse Problems, Image Analysis and Medical Imaging, AMS Special Session Interaction of Inverse Problems and Image Analysis, Volume 313, AMS, Providence, RI, pp. 67–84.

[11] K. Eriksson, D. Estep, P. Hansbo, and C. Johnson, *Introduction to adaptive methods for differential equations*, in Acta Numerica, 1995, Acta Numer., Cambridge University Press, Cambridge, UK, 1995, pp. 105–158.

[12] B. Fischer and J. Modersitzki, *Fast inversion of matrices arising in image processing*, Numer. Algorithms, 22 (1999), pp. 1–11.

[13] J. M. Fitzpatrick, D. L. G. Hill, and C. R. Maurer, Jr., *Image registration, handbook of medical imaging*, in Medical Image Processing and Analysis, Volume 2, SPIE, Bellingham, WA, 2000, pp. 447–513.

[14] C. Glasbey, *A review of image warping methods*, J. Appl. Statist., 25 (1998), pp. 155–171.

[15] A. Ardeshir Goshtasby, 2-*D and* 3-*D Image Registration*, Wiley, New York, 2005.

[16] E. Haber, *Quasi-Newton methods for large-scale electromagnetic inverse problems*, Inverse Problems, 21 (2005), pp. 305–333.

[17] E. Haber and S. Heldmann, *An octree multigrid method for quasi-static Maxwell's equations with highly discontinuous coefficients*, J. Comput. Phys., 223 (2007), pp. 783–796.

[18] E. Haber, S. Heldmann, and J. Modersitzki, *An octree method for parametric image registration*, SIAM J. Sci. Comput., 29 (2007), pp. 2008–2023.

[19] E. Haber and J. Modersitzki, *Numerical methods for volume preserving image registration*, Inverse Problems, 20 (2004), pp. 1621–1638.

[20] E. Haber and J. Modersitzki, *Beyond mutual information: A simple and robust alternative*, in Bildverarbeitung für die Medizin 2005, H. P. Meinzer, H. Handels, A. Horsch, and T. Tolxdorff, eds., Springer-Verlag, New York, 2005, pp. 350–354.

[21] E. Haber and J. Modersitzki, *A multilevel method for image registration*, SIAM J. Sci. Comput., 27 (2006), pp. 1594–1607.

[22] J. Hajnal, D. Hawkes, and D. Hill, *Medical Image Registration*, CRC Press, Boca Raton, FL, 2001.

[23] S. Henn and K. Witsch, *Multimodal image registration using a variational approach*, SIAM J. Sci. Comput., 25 (2003), pp. 1429–1447.

[24] S. Henn and K. Witsch, *Iterative multigrid regularization techniques for image matching*, SIAM J. Sci. Comput., 23 (2001), pp. 1077–1093.

[25] G. Hermosillo, *Variational Methods for Multimodal Image Matching*, Ph.D. thesis, Université de Nice, Nice, France, 2002.

[26] G. R. Hjaltason and H. Samet, *Speeding up construction of quadtrees for spatial indexing*, The VLDB J., 11 (2002), pp. 109–137.

[27] E. M. Kalmoun and U. Rüde, *A Variational Multigrid for Computing the Optical Flow*, Technical report, Department of Computer Science, Friedrich-Alexander University Erlangen-Nuremberg, Erlangen, Germany, 2003.

[28] S. Kruger and A. Calway, *Image registration using multiresolution frequency domain correla-*

tion, in Proceedings of the British Machine Vision Conference, University of Southhampton, Highfield, Southhampton, UK, 1998, pp. 316–325.

[29] H. Lester and S. R. Arridge, *A survey of hierarchical non-linear medical image registration*, Pattern Recognition, 32 (1999), pp. 129–149.

[30] F. Losasso, R. Fedkiw, and S. Osher, *Spatially adaptive techniques for level set methods and incompressible flow*, Comput. & Fluids, 35 (2006), pp. 457–462.

[31] F. Losasso, F. Gibou, and R. Fedkiw, *Simulating water and smoke with an octree data structure*, in Proceedings of the International Conference on Computer Graphics and Interactive Techniques, ACM, New York, 2004, pp. 457–462.

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

[33] C. R. Maurer and J. M. Fitzpatrick, *A review of medical image registration*, in Interactive Image-Guided Neurosurgery, American Association of Neurological Surgeons, Park Ridge, IL, 1993, pp. 17–44.

[34] S. F. McCormick, *Multilevel Adaptive Methods for Partial Differential Equations*, Frontiers Appl. Math. 6, SIAM, Philadelphia, 1989.

[35] J. Modersitzki, *Numerical Methods for Image Registration*, Oxford University Press, Oxford, UK, 2004.

[36] J. Modersitzki, G. Lustig, O. Schmitt, and W. Obelöer, *Elastic registration of brain images on large PC-clusters*, Future Generation Computer Systems, 18 (2001), pp. 115–125.

[37] J. Nocedal and S. Wright, *Numerical Optimization*, Springer-Verlag, New York, 1999.

[38] J. P. W. Pluim, J. B. Antoine Maintz, and Max A. Viergever, *Image registration by maximization of combined mutual information and gradient information*, IEEE TMI, 19 (2000), pp. 809–814.

[39] A. Roche, *Recalage d'images médicales par inférence statistique*, Ph.D. thesis, Université de Nice, Sophia-Antipolis, France, 2001.

[40] R. Szeliski and S. Lavallée, *Matching* 3*-D anatomical surfaces with non-rigid deformations using octree-splines*, Internat. J. Comput. Vision, 18 (1996), pp. 171–186.

[41] R. Szeliski and H. Y. Shum, *Motion estimation with quadtree splines*, IEEE Trans. Pattern Analysis and Machine Intelligence, 18 (1996), pp. 1199–1210.

[42] J.-P. Thirion, *Non-rigid matching using demons*, in Proceedings of the 1996 Conference on Computer Vision and Pattern Recognition, IEEE Computer Society, Washington, DC, 1996, p. 245.

[43] U. Trottenberg, C. Oosterlee, and A. Schuller, *Multigrid*, Academic Press, New York, 2001.

[44] B. Zitová and J. Flusser, *Image registration methods: A survey*, Image and Vision Computing, 21 (2003), pp. 977–1000.