

Technical Report

TR-2007-005

Adaptive Mesh Refinement for Non-Parametric Image Registration

by

Eldad Haber, Stefan Heldmann, Jan Modersitzki

MATHEMATICS AND COMPUTER SCIENCE

EMORY UNIVERSITY

Adaptive Mesh Refinement for Non-Parametric Image Registration

Eldad Haber*, Stefan Heldmann* and Jan Modersitzki†

April 9, 2007

Abstract

3D image registration is a computationally extensive problem which is commonly solved in medical imaging. The complexity of the problem stems from its size and non-linearity. 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 octree grids. It further requires an appropriate refinement criteria. We show that this approach can reduce the computational time in a factor of 10 or so compared to the non-adaptive approach.

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, 34, 14, 14, 33, 30, 33, 13, 23, 45, 36, 15] and references therein.

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 ideal 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 [39, 40, 26, 21]. The second challenge stems from the inherent ill-posedness of the problem [36]. Hence, regularization is inevitable. Parametric and non-parametric 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

*Department of Mathematics and Computer Science, Emory University, Atlanta, GA, USA.

†Institute of Mathematics, University of Lübeck, Germany.

non-parametric approach, an explicit regularizer or penalty for unwanted transformations is introduced; see [36] for an overview. Probably the most commonly used regularizer is the elastic potential; see, e.g. [7, 9, 36]. 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 non-parametric approach is the most powerful one. In fact, other approaches might be considered as particular specifications; see [36]. 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 3D images where tens or hundred of millions of unknowns need to be evaluated.

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 towards fast implementations have been discussed in the literature: iterative solvers [9, 37], specialized direct solvers [12], fast filter techniques [43, 6], multigrid [25, 24, 10, 28, 19]. All these techniques are combined with a multilevel strategy. However, they all use the original image grid as a finest grid. Already for moderate sized 3D images this results in large degrees of freedom. For an example, for 128^3 images, one already ends up with roughly 6 million unknowns. Thus, even a super fast 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 [35, 11, 5] and reference therein. Nevertheless, the use of adaptive meshing to inverse problems is a relatively new field with very little references; see [2, 3, 4]. To our best knowledge, it is completely new in the field of elastic image registration. Some relevant work on octree based image registration is in [41, 42, 29] and our recent work on parametric image registration [18]. In [42, 29] the displacement field was discretized using quadtree splines and in [41] a 2D surface was embedded in 3D and represented using an octree. Other relevant contributions using octrees in image processing has been made in the field of computer graphics [32, 31]. 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 L-BFGS method 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 image registration, the objective is to minimize the functional

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

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. [36]. 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 [38].

In standard approaches, J and the displacement field u are discretized on the voxels of the underlying images. Therefore, a standard discretization [22] in 3D 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

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

where $u^h = [u_1^h, \dots, u_3^h]^\top$, 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 [22] for details. Here, we use the elastic potential with $B = (\mu^{1/2} I_3 \otimes \nabla, \lambda^{1/2} \nabla \cdot)$ such that

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

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

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

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 less grid points than Ω_h for the discretization u^h of u in order to reduce the

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
-

number of unknowns and thus the computational cost. Since the image grid does not necessarily coincide with the transformation grid we construct a linear operator Q that maps u^h from the sparse grid, S_h , to the image grid, Ω_h . In this paper for the sake of simplicity, we assume that the image grid does not reduce the information in the image. The new objective function is thus

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

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

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 to 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 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

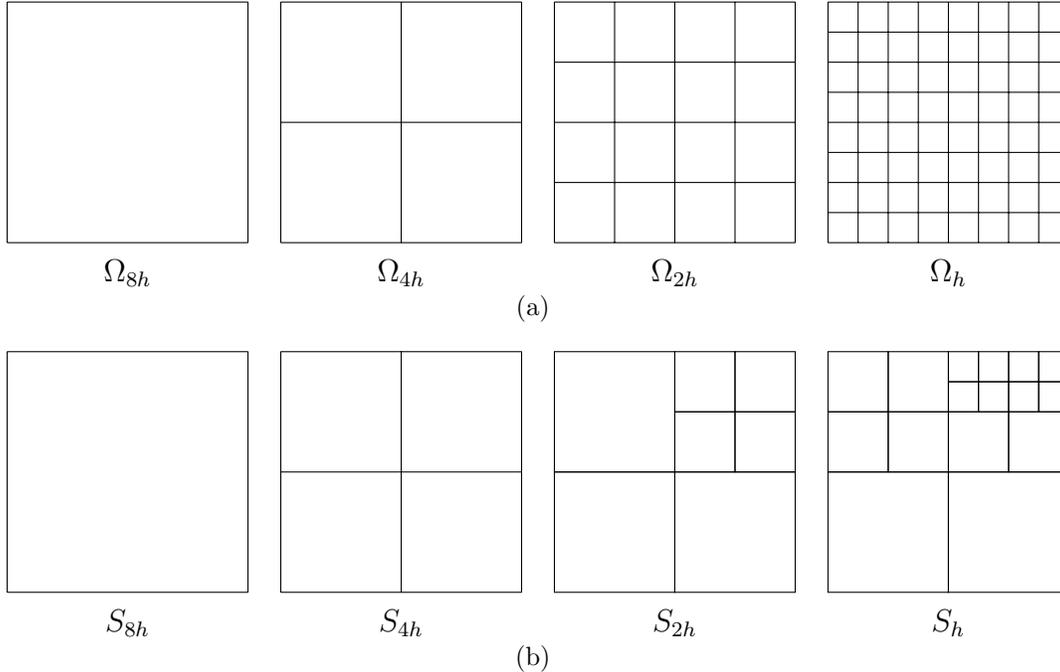


Figure 1: 2D example for grid refinement. (a) regular, refinement (b) adaptive

uniform underlying fine grid Ω_h with cell width h ; see Figure 1(b). 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 size. Each of these cells can have a width $2^j h$ where $0 \leq j \leq L$. 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 3D 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 [27]. For example, for the sparse grid S_{2h} presented in Figure 1 the non-zero entries are stored as

with ∂_j^h the standard central finite differences approximation and $\ell = 1, 2$. Thus, for the quadtree in Figure 2 we obtain the second order approximations

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

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

$$\int_{\text{cell}_j} |\partial_1 u_\ell|^2 dx = \frac{v_j}{2} \left[\partial_1^h u_\ell(x_1, x_2 - h) \right]^2 + \frac{v_j}{2} \left[\partial_1^h u_\ell(x_1, x_2 + h) \right]^2 + \mathcal{O}(h^2) \quad (3.4a)$$

$$\begin{aligned}\int_{\text{cell}_j} |\partial_2 u_\ell|^2 dx &= \frac{v_j}{4} \left[\partial_2^h u_\ell(x_1 + h, x_2 - \frac{h}{2}) \right]^2 + \frac{v_j}{4} \left[\partial_2^h u_\ell(x_1 + h, x_2 + \frac{h}{2}) \right]^2 \\ &\quad + \frac{v_j}{2} \left[\partial_2^h u_\ell(x_1 - h, x_2) \right]^2 + \mathcal{O}(h^2).\end{aligned} \quad (3.4b)$$

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

For ease of presentation, we derive a matrix representation for the discrete gradient operator. Let \odot denote the Hadamard (or element-wise) 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 be an average matrix from edges to cell-center of each cell, we have

$$\begin{aligned}\int_{\Omega} |\nabla u_\ell|^2 dx &= \int_{\Omega} |\partial_1 u_\ell|^2 + |\partial_2 u_\ell|^2 dx \\ &= v^\top A_e^c \left((\nabla^h u_\ell^h) \odot (\nabla^h u_\ell^h) \right) + \mathcal{O}(h^2) \\ &= (u_\ell^h)^\top (\nabla^h)^\top \text{diag}[(A_e^c)^\top v] \nabla^h u_\ell^h + \mathcal{O}(h^2).\end{aligned} \quad (3.5)$$

Note that the regularization is quadratic in u_ℓ^h with a symmetric positive semi-definite matrix $(\nabla^h)^\top \text{diag}[(A_\ell^c)^\top v] \nabla^h$. The diagonal weighting matrix $\text{diag}[(A_\ell^c)^\top v]$ 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 cell-center. For the quadtree presented in Figure 2 we obtain

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

and hence

$$\begin{aligned} \int_{\text{cell}_j} (\nabla \cdot u)^2 dx &= \frac{v_j}{2} \left(\partial_1^h u_1(x_1, x_2 + h) + \partial_1^h u_1(x_1, x_2 - h) + \right. \\ &\quad \left. \partial_2^h u_2(x_1 - h, x_2) + \partial_2^h u_2(x_1 + h, x_2) \right) + \mathcal{O}(h^2). \end{aligned} \quad (3.6)$$

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

$$\begin{aligned} \int_{\Omega} (\nabla \cdot u)^2 dx &= v^\top (\nabla^h \cdot u^h) \odot (\nabla^h \cdot u^h) + \mathcal{O}(h^2) \\ &= (u^h)^\top (\nabla^h \cdot)^\top \text{diag}(v) (\nabla^h \cdot) u^h + \mathcal{O}(h^2). \end{aligned} \quad (3.7)$$

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

$$A^h = \mu I_d \otimes \left[(\nabla^h)^\top \text{diag}[(A_\ell^c)^\top v] \nabla^h \right] + \lambda (\nabla^h \cdot)^\top \text{diag}(v) (\nabla^h \cdot). \quad (3.8)$$

4 Solving the optimization problem

Since we use standard optimization techniques with implementation details similar to previous work [22], we only briefly summarize the strategy.

We implemented the L-BFGS method as suggested in [38]. 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

$$J(u) = \frac{1}{2} \|T(Qu) - R\|_2^2 + \frac{\alpha}{2} u^\top Au. \quad (4.9)$$

Any gradient descent direction requires the computation of the gradient of the objective. 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 [20, 22]. Though the Jacobian is a sparse matrix, the nonzero entries can vary in order of magnitudes.

For the L-BFGS method we build an approximation to the inverse of the Hessian by using the most recent L directions $\{u^{(k-L)}, \dots, u^{(k)}\}$ and the gradients $\{\nabla J^{(k-L)}, \dots, \nabla J^{(k)}\}$ and an initial approximation to the Hessian; see [38, 16] for implementation details. As discussed in [16], it is crucial to initialize the approximation to the Hessian with the regularization operator. 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 required to be h-elliptic (see [44] 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 successfully applied to nodal discretization of such systems [44].

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., [44].

We next develop a refinement criteria. 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,

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

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

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

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

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

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 on 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 $|\nabla \rho|$ 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 we already have to compute 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 $|\nabla^h \rho_j^h|$ 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 $|\nabla^h \rho_j^h|$ is large we use a parameter τ and refine every cell that satisfies $|\nabla^h \rho_j^h| > \tau$. The refinement process is terminated when $|\nabla^h \rho_j^h| \leq \tau$ for all cells.

6 Numerical experiments

To demonstrate our method, we present two 128^2 2D and a 128^3 3D examples. 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. As 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 was chosen between 0 (refine everywhere) to 10. In all experiments we started our multi-level method on a coarsest mesh consisting of a single cell yielding 2×4 and 3×8 unknowns in 2D and 3D, respectively. The stopping criteria 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.

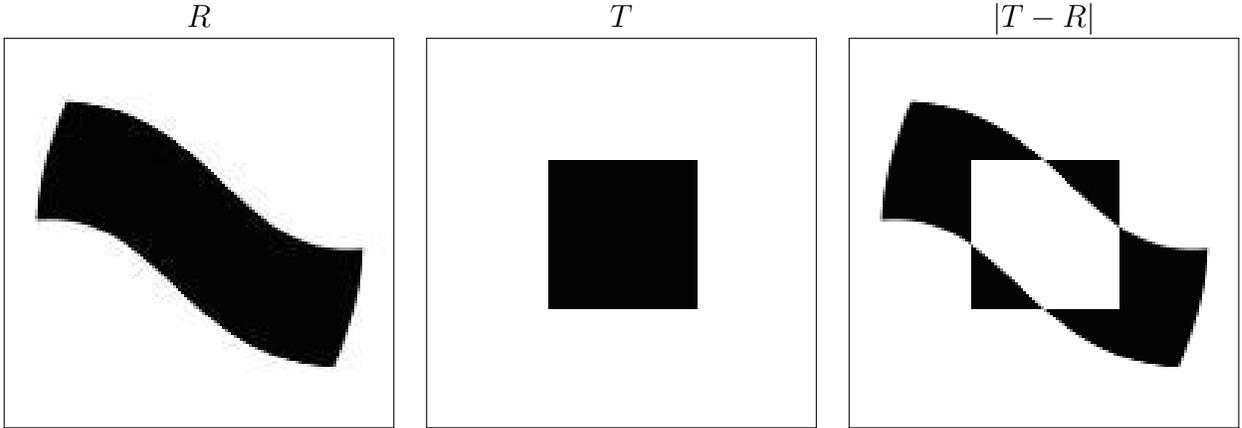


Figure 3: Images for the 2D "Flag" example

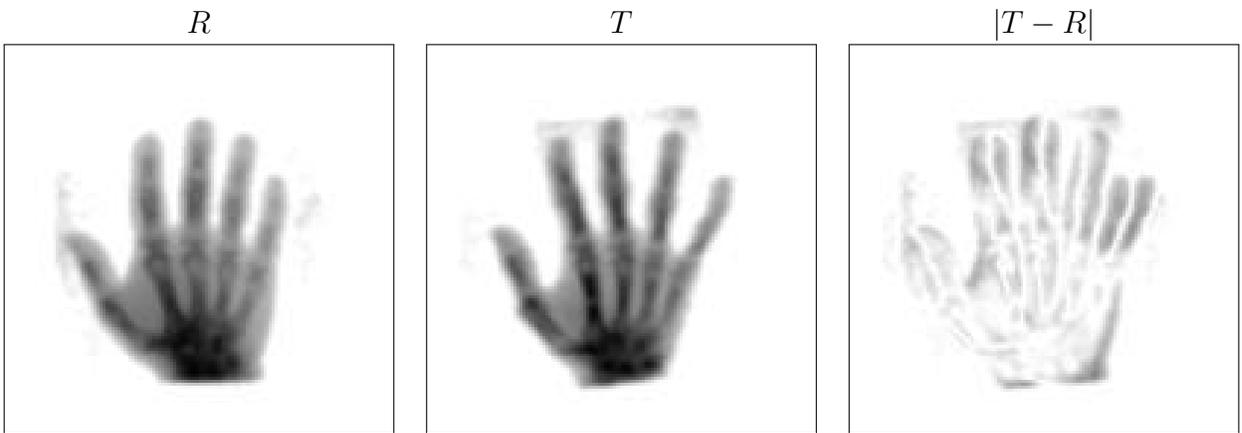


Figure 4: Images for the 2D "Hand" example

In 2D, 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. All images have a size of 128×128 pixels.

In 3D, we registered two CT data sets of the chest under maximal inspiration and expiration respectively, cf. Figure 3. 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., $\|T(u_{\text{opt}}) - R\|/\|T - R\|$ 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 $\tau = 0$.

Our results show, that the proposed method scales well with the grid tolerance and in general we found an acceleration of factor circa 6–8 in the 2D and speedup of approximately 10 in 3D. Furthermore, we observed, that the overhead for the adaptive grid refinement was in general less than 0.1% of the execution time.

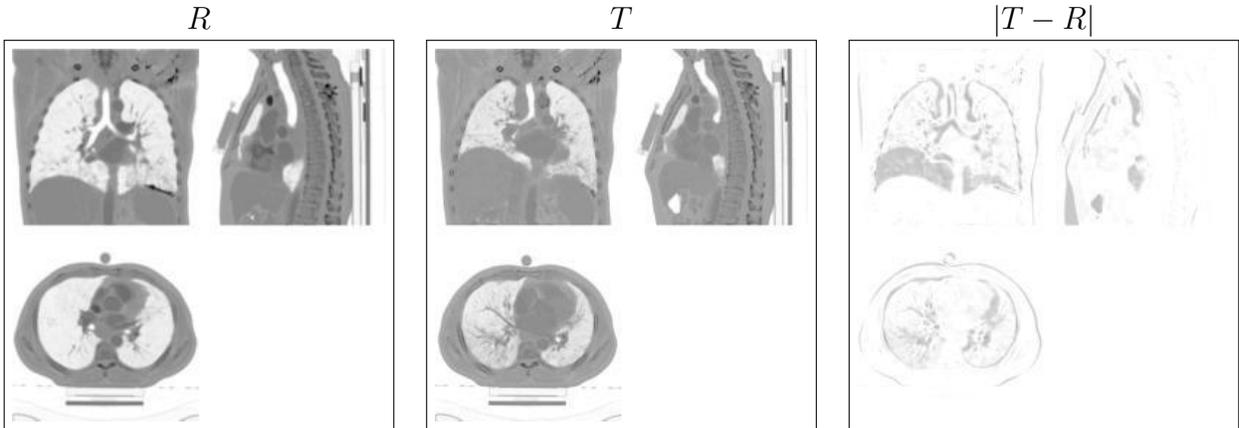


Figure 5: Images for the 3D CT example (volumes orthogonal sliced)

| τ | #cells | misfit | elapsed time | | | | speedup |
|--------|--------|--------|--------------|---------|---------|---------|---------|
| | | | total | interp. | solving | meshing | |
| 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 1: Results for the 2D “Flag” example

To give a visual impression for the quality of the results we show present few of the obtained images and grids in Figures 6, 7, and 8.

7 Summary

In this paper we have developed an adaptive multilevel refinement method for non-parametric image registration. We have used the elastic potential as a regularizer and demonstrated that it can be effectively and accurately discretized on octree grids. We have used the L-BFGS method for optimization which requires the solution of a linear system involved the Hessian of the regularizer. We develop a refinement criteria based on the accurate evaluation of the variational form. Numerical experiments demonstrate that an order reduction in problem size and computational time is obtained by using our method.

| τ | #cells | misfit | elapsed time | | | | speedup |
|--------|--------|--------|--------------|---------|---------|---------|---------|
| | | | total | interp. | solving | meshing | |
| 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 2: Results for the 2D "Hand" example

| τ | #cells | misfit | elapsed time | | | | speedup |
|--------|---------|--------|--------------|---------|----------|---------|---------|
| | | | total | interp. | solving | meshing | |
| 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 |

Table 3: Results for the 3D "CT" example

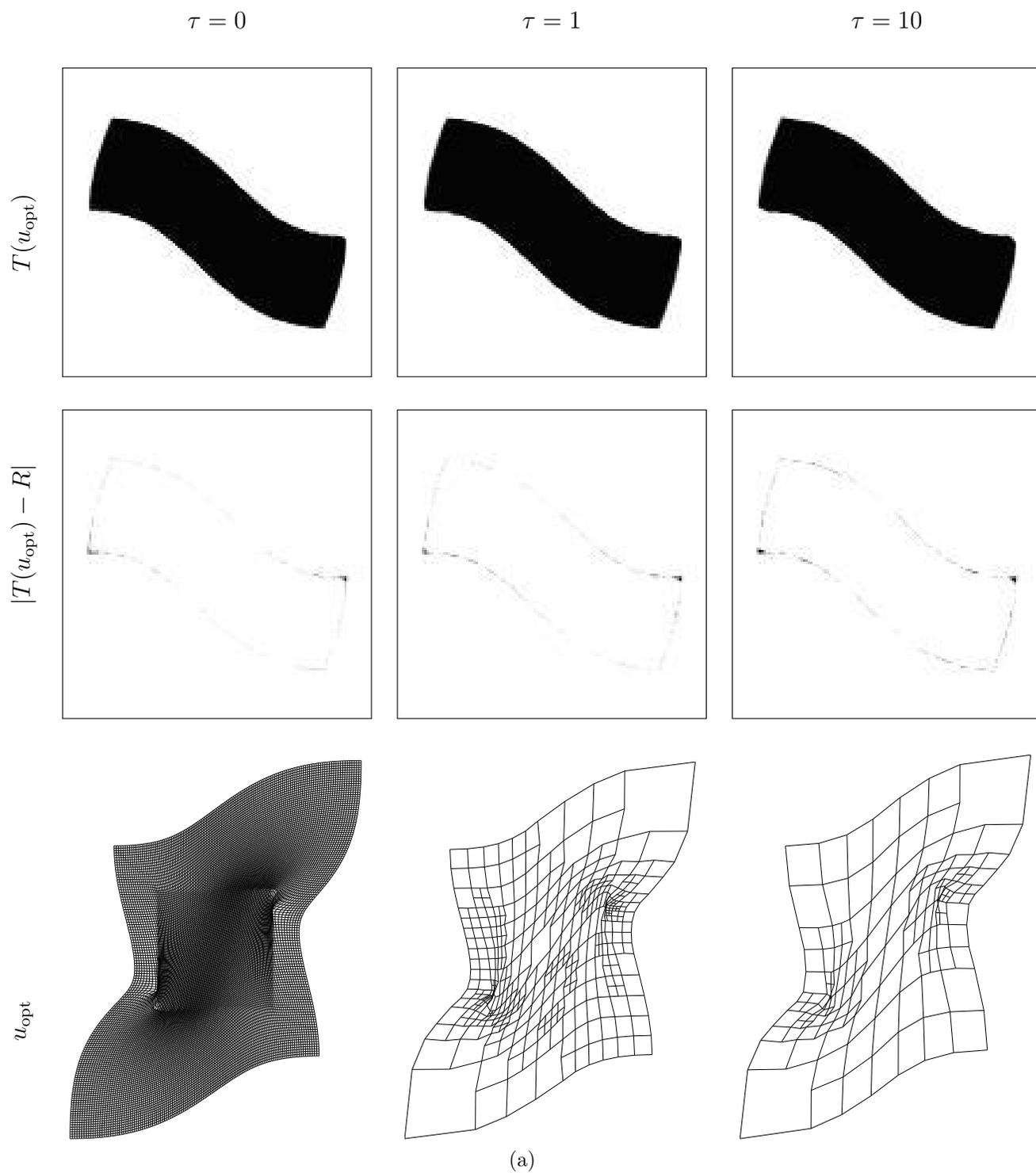


Figure 6: Some results for the "flag" example

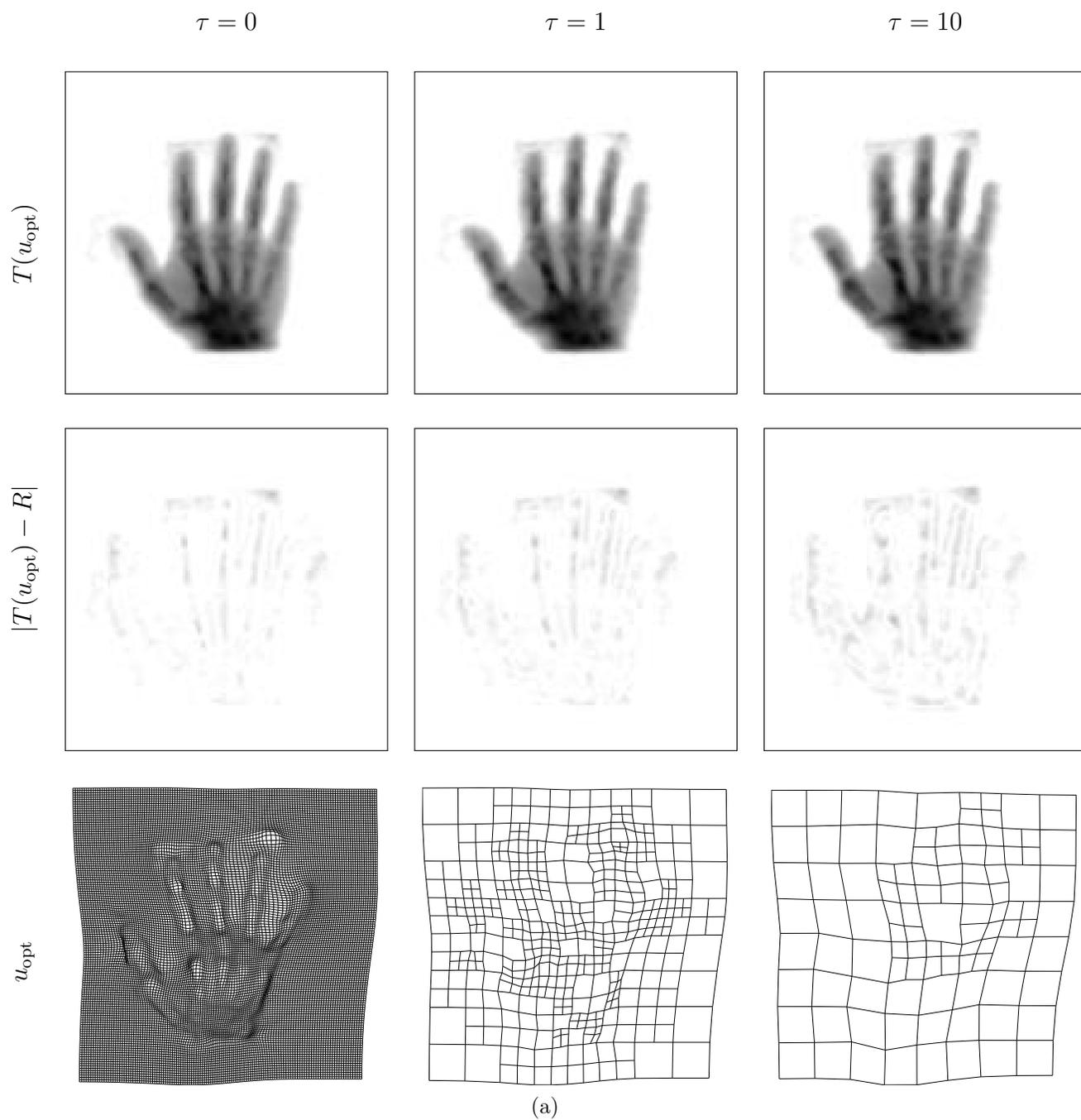


Figure 7: Some results for the "hand" example

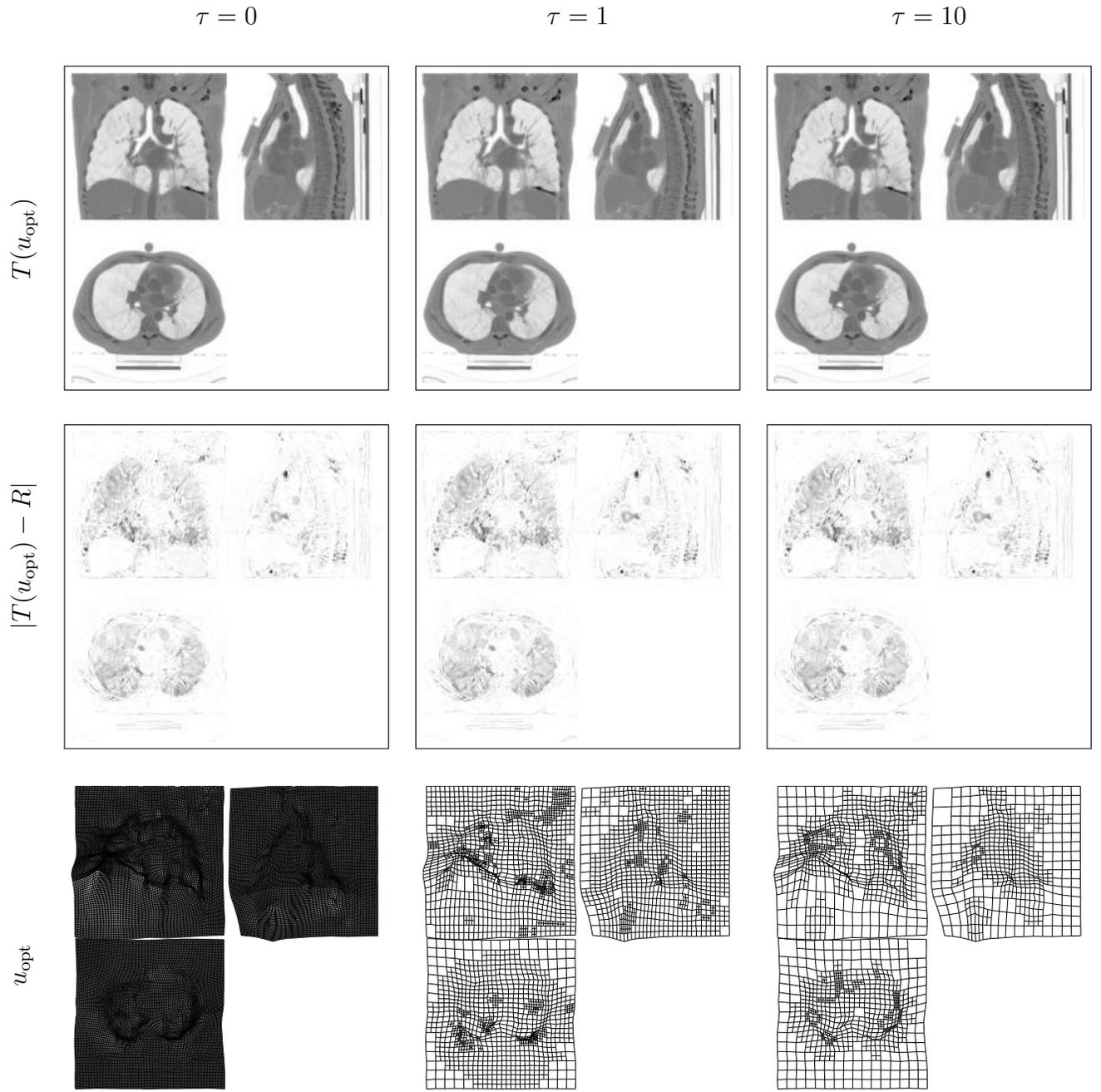


Figure 8: Some results for the 3D CT example

References

- [1] U. Ascher and E. Haber. Grid refinement and scaling for distributed parameter estimation problems. *Inverse Problems*, 17:571–590, 2001.
- [2] W. Bangerth. Adaptive finite element methods for the identification of distributed coefficients in partial differential equations. *PhD Thesis*, University of Heidelberg:Germany, 2002.
- [3] R. Becker. Adaptive finite element methods for optimal control problems. *Habilitation Thesis*, University of 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*, 39:113–132, 2000.
- [5] Marsha Berger. *Adaptive Mesh Refinement for Time-Dependent Partial Differential Equations*. PhD dissertation, Stanford University, 1982. Computer Science Report No. STAN-CS-82-924.
- [6] Morten Bro-Nielsen and Claus Gramkow. Fast fluid registration of medical images. *Lecture Notes in Computer Science*, 1131:267–276, 1996.
- [7] Chaim Broit. *Optimal registration of deformed images*. PhD thesis, Computer and Information Science, 1981.
- [8] L.G. Brown. A survey of image registration techniques. *Surveys*, 24(4):325–376, December 1992.
- [9] Gary Edward Christensen. *Deformable shape models for anatomy*. PhD thesis, Sever Institute of Technology, Washington University, 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, pages 67–84. AMS, 2002.
- [11] K. Eriksson, D. Estep, P. Hansbo, and C. Johnson. Introduction to adaptive methods for differential equations. *Acta Numerica*, pages 105–158, 1995.
- [12] Bernd Fischer and Jan Modersitzki. Fast inversion of matrices arising in image processing. *Num. Algo*, 22:1–11, 1999.
- [13] J. M. Fitzpatrick, D. L. G. Hill, and C. R. Maurer Jr. Image registration, handbook of medical imaging. *Volume 2: Medical Image Processing and Analysis, SPIE*, pages 447–513, 2000.
- [14] Chris Glasbey. A review of image warping methods. *Journal of Applied Statistics*, 25:155–171, 1998.

- [15] A. Ardeshir Goshtasby. *2-D and 3-D Image Registration*. Wiley Press, New York, 2005.
- [16] E. Haber. Quasi-newton methods methods for large scale electromagnetic inverse problems. *Inverse Problems*, 21, 2005.
- [17] E. Haber and S. Heldmann. An octree multigrid method for quasi-static Maxwell's equations with highly discontinuous coefficients. *Journal of Comput. Phys.*, page to appear, 2007.
- [18] E. Haber, S. Heldmann, and J. Modersitzki. An octree method for parametric image registration. 2006.
- [19] E. Haber and J. Modersitzki. Multilevel methods for image registration. *SIAM J. on Scientific Computing*, 27:1594–1607, 2004.
- [20] E. Haber and J. Modersitzki. Numerical methods for volume preserving image registration. *Inverse Problems, Institute of Physics Publishing*, 20(5):1621–1638, 2004.
- [21] E. Haber and J. Modersitzki. Beyond mutual information: A simple and robust alternative. In HP Meinzer, H Handels, A Horsch, and T Tolxdorff, editors, *Bildverarbeitung für die Medizin 2005*, pages 350–354. Springer, 2005.
- [22] E. Haber and J. Modersitzki. A multilevel method for image registration. *SIAM J. Sci. Comput.*, 27(5):1594–1607, 2006.
- [23] J Hajnal, D Hawkes, and D Hill. *Medical Image Registration*. CRC Press, 2001.
- [24] S. Henn and K. Witsch. Multimodal image registration using a variational approach. *SIAM J. Sci. Comp.*, 25:1429–1447, 2003.
- [25] Stefan Henn and Kristian Witsch. Iterative multigrid regularization techniques for image matching. *SIAM J. on Scientific Comp.*, pages 1077–1093, 2001.
- [26] G. Hermosillo. *Variational methods for multimodal image matching*. PhD thesis, Université de Nice, France, 2002.
- [27] G. R. Hjaltason and H. Samet. Speeding up construction of quadtrees for spatial indexing. *The VLDB Journal*, 11:109–137, 2002.
- [28] El Mostafa Kalmoun and Ulrich Rüde. A variational multigrid for computing the optical flow. Technical report, Department of Computer Science, Friedrich-Alexander University Erlangen-Nuremberg, 2003.
- [29] S. Kruger and A. Calway. Image registration using multiresolution frequency domain correlation. *British Machine Vision Conference*, September:316–325, 1998.
- [30] Hava Lester and Simon R. Arridge. A survey of hierarchical non-linear medical image registration. *Pattern Recognition*, 32:129–149, 1999.

- [31] F. Losasso, R. Fedkiw, and S. Osher. Spatially adaptive techniques for level set methods and incompressible flow. *Computers and Fluids*, 35:457–462, 2006.
- [32] F. Losasso, F. Gibou, and R. Fedkiw. Simulating water and smoke with an octree data structure. *SIGGRAPH*, 23:457–462, 2004.
- [33] J. B. Antoine Maintz and Max A. Viergever. A survey of medical image registration. *Medical Image Analysis*, 2(1):1–36, 1998.
- [34] C.R. Maurer and J.M. Fitzpatrick. A review of medical image registration. In *In: Interactive Image-Guided Neurosurgery, American Association of Neurological Surgeons*, pages 17–44, Park Ridge, IL, Aug 1993.
- [35] S. F. McCormick. *Multilevel Adaptive Methods for Partial Differential Equations*. SIAM, 1989.
- [36] J. Modersitzki. *Numerical methods for image registration*. Oxford, 2004.
- [37] J. Modersitzki, G. Lustig, O. Schmitt, and W. Obelöer. Elastic registration of brain images on large PC-clusters. *Elsevier, Future Generation Computer Systems*, 18:115–125, 2001.
- [38] J. Nocedal and S. Wright. *Numerical Optimization*. New York: Springer, 1999.
- [39] Josien PW Pluim, J. B. Antoine Maintz, and Max A. Viergever. Image registration by maximization of combined mutual information and gradient information. *IEEE TMI*, 19:809–814, 2000.
- [40] A. Roche. *Recalage d’images médicales par inférence statistique*. PhD thesis, Université de Nice, Sophia-Antipolis, France, 2001.
- [41] Richard Szeliski and Stéphane Lavallée. Matching 3-D anatomical surfaces with non-rigid deformations using octree-splines. *International Journal of Computer Vision*, 18(2):171–186, 1996.
- [42] Richard Szeliski and H.Y. Shum. Motion estimation with quadtree splines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18, Issue 12:1199 – 1210, 1996.
- [43] Jean-Philippe Thirion. Non-rigid matching using demons, 1996. IEEE 1996.
- [44] U. Trottenberg, C. Oosterlee, and A. Schuller. *Multigrid*. Academic Press, 2001.
- [45] Barbara Zitová and Jan Flusser. Image registration methods: a survey. *Image and Vision Computing*, 21(11):977–1000, 2003.