GENERAL-PURPOSE MEDICAL IMAGE REGISTRATION

A Dissertation

Submitted to the Faculty

in partial fullfillment of the requirements for the

degree of

Doctor of Philosophy

in

Computer Science

by

Senthil Periaswamy

Dartmouth College

Hanover, NH

April 2003

Examining Committee:

(chair) Hany Farid

John B. Weaver

Dan Rockmore

Souheil Inati

Carol Folt
Dean of the Graduate Studies

Abstract

GENERAL-PURPOSE MEDICAL IMAGE REGISTRATION

Senthil Periaswamy

We have developed a general-purpose registration algorithm for medical images and volumes. The transformation between images is modeled as locally affine but globally smooth, and explicitly accounts for local and global variations in image intensities. An explicit model for missing data is also incorporated, allowing us to simultaneously segment and register images with partial or missing data. The algorithm is built upon a differential multiscale framework and incorporates the expectation-maximization (EM) algorithm. We show that this approach is highly effective in registering a range of clinical medical images.

Dedicated to my father, Dr. Sirappalli Periyannan Periaswamy,

in whose footsteps I follow.

Acknowledgments

This work is based on all that I have learnt from my current and former teachers, mentors and advisors. My contribution is only incremental. I also wish to express my heartfelt thanks to my family and friends, for all their support, and my advisor, Hany Farid, for his guidance, friendship, and for teaching me the art of critical thinking.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Our primary goal is to design and analyze an algorithm for the registration of images. Image registration (also known as image fusion, super-imposition, or matching) can be broadly defined as the process of finding a transformation that aligns one image to another. Our focus in particular is on medical images, which presents some unique challenges, and for which there are a large number of interesting applications. We begin the first chapter with a formal statement of the problem, including an intuitive explanation of why this problem is hard. The importance of registration will also be explored along with a discussion of current techniques. In the second chapter, the details of our algorithm are fully described. In the third chapter, the efficacy of the algorithm on a wide range of simulated and real imagery is demonstrated. In the fourth chapter, the results of a sensitivity analysis of the entire system with respect to the design assumptions are presented. Finally, in Chapter 5, some possible applications of image registration are explored. Readers primarily interested in the details of our registration algorithm, or already familiar with the fundamentals of image registration, are encouraged to begin in Chapter 2.

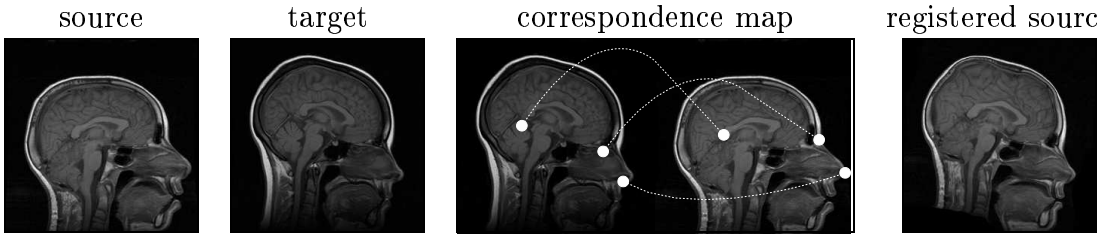


Figure 1.1: The goal of registration is to find corresponding points between a source and target image (columns 1 and 2). Shown in column 3 are corresponding points in the eye, nose and corpus-colusum regions between the two slices. Shown in the last column is the registered source, obtained by warping the source image using the correspondence map.

1.1 Problem description

1.1.1 Definition

Image registration can be defined as the process of finding a mapping that aligns one image to another. With this mapping, a correspondence between pixels in a source image with the pixels in a target image is established. More precisely, the goal of registration is to find a correspondence function, or mapping, $\mathcal{M}(\cdot)$, that takes each spatial coordinate $\vec{x}_t$ from the target image and returns a coordinate $\vec{x}_s$ for the source image:

$$\vec{x}_s = \mathcal{M}(\vec{x}_t). \tag{1.1}$$

Once a registration map $\mathcal{M}(\cdot)$ has been obtained, the source image may be brought into registration with the target image by warping the source using interpolation, as shown in Figure 1.1.[1]

[1]Note that in this figure, a sparse correspondence map is shown. If the mapping is found between all source and target pixels, it is called a dense mapping. A sparse mapping can also be interpolated to obtain a dense mapping.

1.1.2 Representation of the mapping

In order to visualize the registration mapping $\mathcal{M}(\cdot)$, two approaches are used: a *warped grid* approach and a *flow* approach. The warped grid approach takes a regular grid and uses the mapping $\mathcal{M}(\cdot)$ to warp it. The flow approach depicts the mapping at a subset of discrete point with a translation vector. The vector at each source pixel location describes the magnitude and direction to the corresponding target pixel location. Figure 1.2 shows some examples of the grid and flow representations. In the case of the grid representation, it is harder to see small translations (a translated grid is still a grid). In the case of the flow representation, it is harder to visualize the overall effect of the transformation. These two formats will be used interchangeably throughout this thesis to depict the registration mapping.

1.1.3 Applications

Image registration has evolved independently in various research areas, ranging from geo-surveying to medical imaging, each with a number of unique applications. Medical image analysis, in particular, has many challenging and useful applications. The need for registration can arise when images of a given piece of anatomy are taken over a period of time and need to be compared, as is the case for the study of tumor growth (e.g., longitudinal studies), or in functional magnetic resonance imaging (fMRI) studies. In fMRI studies, many magnetic resonance (MR) images are taken of the brain in quick succession, and need to be registered to a higher resolution anatomic image, which is in turn registered to an atlas image. Registration can also be used to characterize normal versus abnormal anatomical shape variations. For example, tumors may be registered to normal or abnormal tumors within the same class. Other applications include labeling and segmentation, in which the image to be

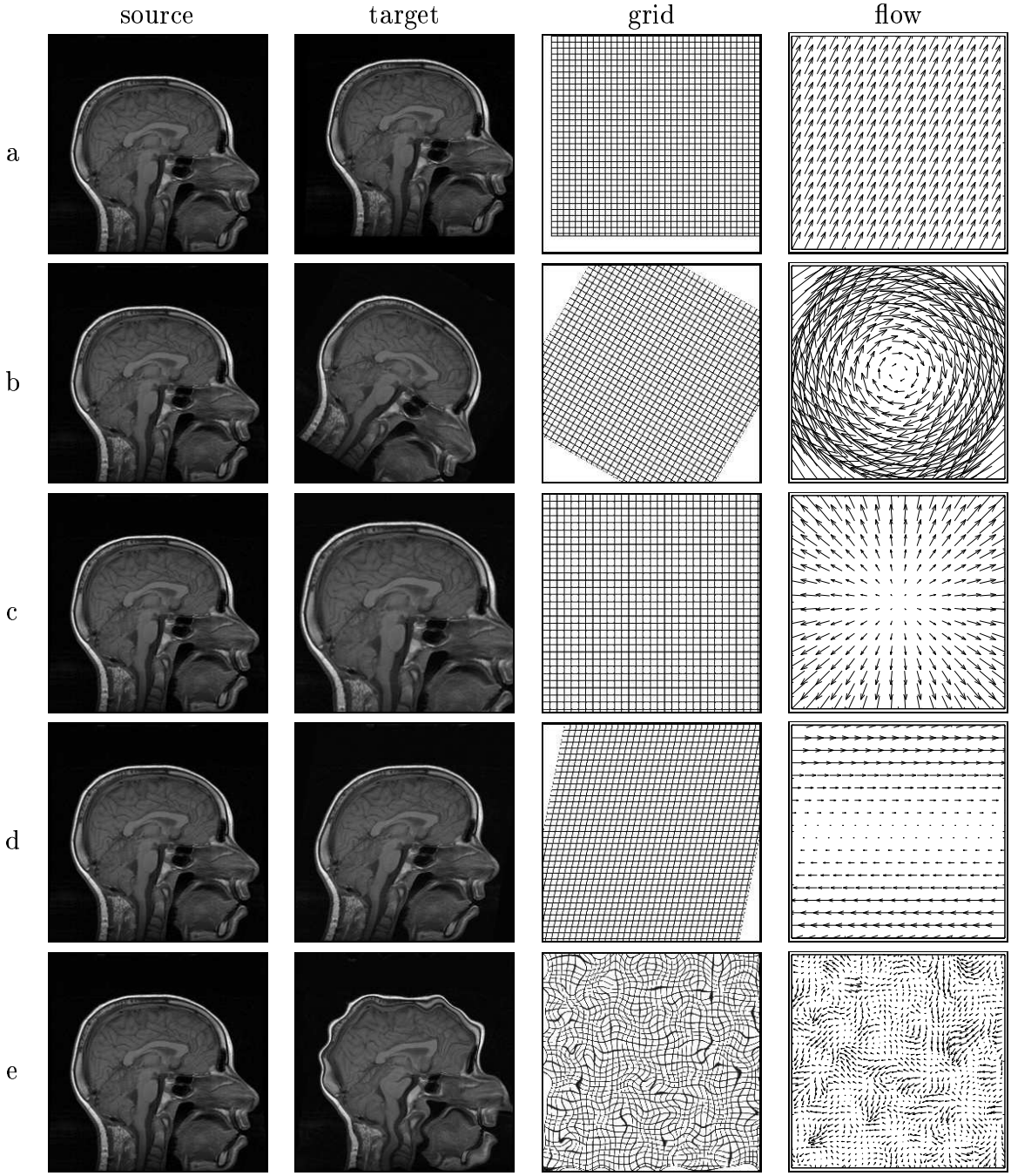


Figure 1.2: Shown above are the grid and flow representations for some sample mappings. Shown from the top to bottom are examples of a translation, rotation, scaling, shear and a random correspondence map.

labeled/segmented is registered to a previously labeled/segmented image (the 'atlas'). This atlas registration circumvents the need for explicit labeling/segmentation.

The images that need to be compared may also be obtained using different hardware devices, each highlighting a specific part of the body (these images are also referred to as multi-modality images). Examples of common multi-modality images in medical imaging include Computed Tomography (CT), X-ray imaging, Positron Emission Tomography (PET) Imaging, and Nuclear Magnetic Resonance (NMR or MR) Imaging. If these images are registered, they can be combined in a more meaningful way to provide an integrated view (image fusion). Applications that involve multi-modality image registration include image-guided surgery, in which the surgeon may be assisted with a 3-D view consisting of fused images of different modalities. This information may also be integrated with positional information from the surgeon's tool, providing further assistance. Another application for registration is radiation therapy planning in nuclear medicine, in which a CT scan is used for dosage calculations, while the contours of the target lesion are easier to outline using MRI. Multi-modality registration between the CT scans and the MRI helps optimize the dosage of radiation.

There are also a large number of applications outside the realm of medical imaging, such as mosaicking and view rectification in remotely sensed data processing, target tracking in defense, face/thumb/retinal recognition in security, video compression, motion estimation and stereo in computer vision, to name just a few. As can be seen, image registration is an important precursor to a wide range of practical applications, each with their own specific challenges. Medical imaging, in particular, has some unique challenges, which are explored next.

1.1.4 Difficulties

Medical image registration poses unique challenges, due to the rich variety of application and the rapid growth of acquisition hardware. We believe that there are three main reasons that make registration difficult. Firstly, the distortions may be both large and small (see Figures 1.3(a)-(b)), making it difficult to find a simple model for the distortion. Secondly, a difficulty arises as the images may be acquired using different hardware devices, or different settings within a given hardware device, leading to multi-modality images, which vary in overall appearance. Note that each of these modalities can be tuned to highlight different features within the body by varying the acquisition parameters, acquisition protocol, or by the introduction of contrast agents in the body. Since each of these modalities have different intensity characteristics, registration becomes hard (Figure 1.3(c)). Thirdly, another difficulty occurs in cases where only partial data is available for registration (Figure 1.3(d)). Many registration algorithms are simply ill equipped to handle this situation, and are likely to fail completely. For the above reasons, it is difficult for any single general-purpose algorithm to perform effectively across the broad range of applications. However, techniques exist that try to address one or more of these problems. Before discussing some of these techniques, we first explore some of the design considerations that go into a registration algorithm.

1.2 Design considerations

In this section, we explore some of the design considerations in formulating a registration algorithm. As discussed in Section 1.1.4, it is unlikely that a single algorithm will completely solve the registration problem; instead there are a few standard techniques and tools that can be used in combination in a registration algorithm. In this

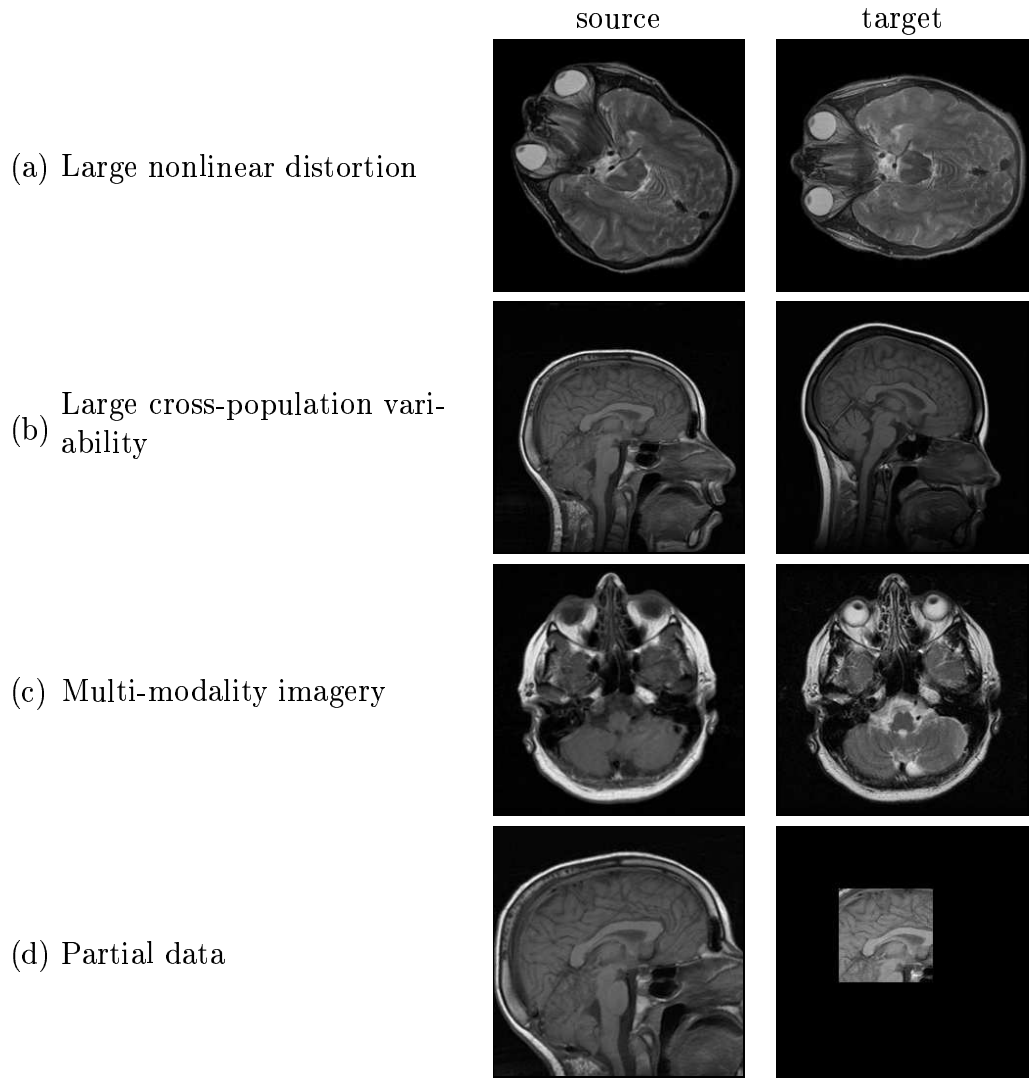


Figure 1.3: Shown above are some of the common difficulties encountered during image registration. There may be both large overall distortions (a) as well as small distortions (b). The images may vary in overall appearance (c). Finally, there may only be partial data available for the registration (d).

section, we will also explore some of these existing tools. We will also discuss some commonly used registration software.

1.2.1 Classification

A large body of literature exists in the field of image registration (surveys can be found in [17, 69, 44, 74, 43, 41]). Our approach towards categorizing this literature is to describe the design decisions that go into most registration algorithms.

In estimating the transformation between two images we must choose: 1.) to estimate the transformation between a small number of extracted landmarks / features, or between the complete unprocessed intensity images; 2.) a model that describes the geometric transform; 3.) whether to and how to explicitly model intensity changes; 4.) an error metric that incorporates the previous three choices; and 5.) a minimization technique for minimizing the error metric, yielding the desired transformation.

Feature Space

Feature-based approaches extract a (typically small) number of corresponding landmarks or features between the pair of images to be registered. The overall transformation is estimated from these features. Common features include corresponding points [61, 26, 14], edges [48, 38], contours [45, 63] or surfaces [53, 28, 24]. These features may be specified manually or extracted automatically. Fiducial markers may also be used as features; these markers are usually selected to be visible in different modalities. Feature-based approaches have the advantage of greatly reducing computational complexity. Depending on the feature extraction process, these approaches may also be more robust to intensity variations that arise during, for example, cross modality registration. Also, features may be chosen to help reduce sensor noise. These

approaches can be, however, highly sensitive to the accuracy of the feature extraction. Intensity-based approaches, on the other hand, estimate the transformation between the entire intensity images. Such an approach is typically more computationally demanding, but avoids the difficulties of a feature extraction stage. Note that the choice of a feature- or intensity-based technique does not restrict the geometric transform; it merely filters the information that is to be used. Recent advances in feature based registration include methods based on level sets [70, 51].

Geometric transform

Independent of the choice of a feature- or intensity-based technique, a model describing the geometric transform is required. A common and straightforward choice is a model that embodies a single global transformation. The problem of estimating a global translation and rotation parameter (referred to as the Orthogonal Procrustes problem) has been studied in detail, and a closed form solution was proposed by Schönemann [61] in 1966. Other closed-form solutions include methods based on singular value decomposition (SVD) [7], eigenvalue-eigenvector decomposition [36] and unit quaternions [35]. One idea for a global transformation model is to use polynomials. For example, a zeroth-order polynomial limits the transformation to simple translations, a first-order polynomial allows for an affine transformation, and, of course, higher-order polynomials can be employed yielding progressively more flexible transformations. For example, the registration package Automated Image Registration (AIR) can employ (as an option) a fifth-order polynomial consisting of 168 parameters (for 3-D registration) [75, 76]. The global approach has the advantage that the model consists of a relatively small number of parameters to be estimated, and the global nature of the model ensures a consistent transformation across the entire image. The disadvantage of this approach is that estimation of higher-order polyno-

mials can lead to an unstable transformation, especially near the image boundaries. In addition, a relatively small and local perturbation can cause disproportionate and unpredictable changes in the overall transformation. An alternative to these global approaches are techniques that model the global transformation as a piecewise collection of local transformations. For example, the transformation between each local region may be modeled with a low-order polynomial, and global consistency is enforced via some form of a smoothness constraint. The advantage of such an approach is that it is capable of modeling highly nonlinear transformations without the numerical instability of high-order global models. The disadvantage is one of computational inefficiency due to the significantly larger number of model parameters that need to be estimated, and the need to guarantee global consistency. Low-order polynomials are, of course, only one of many possible local models that may be employed. Other local models include B-splines [68, 55, 58], thin-plate splines [13, 14], and a multitude of related techniques. The package Statistical Parametric Mapping (SPM) uses the low-frequency discrete cosine basis functions [8, 10], where a bending-energy function is used to ensure global consistency. At least one disadvantage with this particular basis is that it is unable to model, for example, local scale changes. Physics-based techniques that compute a local geometric transform include those based on the Navier-Stokes equilibrium equations for linear elasticity [16, 11, 31] and those based on viscous fluid approaches [31, 29, 30].

Modeling intensity changes

Under certain conditions a purely geometric transformation is sufficient to model the transformation between a pair of images. Under many real-world conditions, however, the images undergo changes in both geometry and intensity (e.g., brightness and contrast). Many registration techniques attempt to remove these intensity differences

with a pre-processing stage, such as histogram matching [47, 72] or homomorphic filtering [49]. The issues involved with modeling intensity differences are similar to those involved in choosing a geometric model. Because the simultaneous estimation of geometric and intensity changes can be difficult, few techniques build explicit models of intensity differences. A few notable exceptions include AIR [75, 76], in which global intensity differences are modeled with a single multiplicative contrast term, and SPM [8, 10] in which local intensity differences are modeled with a basis function approach.

Error metric

Having decided upon a transformation model, the task of estimating the model parameters begins. As a first step, an error function in the model parameters must be chosen. This error function should embody some notion of what is meant for a pair of images to be registered. Perhaps the most common choice is a mean square error (MSE), defined as the mean of the square of the differences (in either feature distance or intensity) between the pair of images. This metric is easy to compute and often affords simple minimization techniques. A variation of this metric is the unnormalized correlation coefficient applicable to intensity-based techniques. This error metric is defined as the sum of the point-wise product of the image intensities, and can be efficiently computed using Fourier techniques [20, 25]. A disadvantage of these error metrics is that images that would qualitatively be considered to be in good registration may still have large errors due to, for example, intensity variations, or slight misalignments. Another error metric (included in AIR) is the ratio of image uniformity (RIU) defined as the normalized standard deviation of the ratio of image intensities. Such a metric is invariant to overall intensity scale differences, but typically leads to nonlinear minimization schemes. Mutual information [21, 71, 65, 67], entropy [19, 66], and the Pearson product moment cross correlation [40] are just a

few examples of other possible error functions. Such error metrics are often adopted to deal with the lack of an explicit model of intensity transformations [57].

Minimization

In the final step of registration, the chosen error function is minimized yielding the desired model parameters. In the most straightforward case, least-squares estimation is used when the error function is linear in the unknown model parameters. This closed-form solution is attractive as it avoids the pitfalls of iterative minimization schemes such as gradient-descent or simulated annealing. Such nonlinear minimization schemes are, however, necessary due to an often nonlinear error function. A reasonable compromise between these approaches is to begin with a linear error function, solve using least-squares, and use this solution as a starting point for a nonlinear minimization. Minimization can be performed using multiscale approach, which helps alleviate the problem of local minima.

Our choices

In developing our general-purpose registration algorithm, we have tried to find a compromise between a flexible and robust technique and computational efficiency. To begin, we have chosen an intensity-based approach so as to avoid the various pitfalls involved in feature selection. Geometrically, we model the transformation with a local affine model and a global smoothness constraint. Intensity variations are explicitly modeled with local changes in brightness and contrast and a global smoothness constraint. These model parameters are simultaneously estimated at each pixel in the image, allowing us to capture nonlinear distortions (in both geometry and intensity). We employ a standard MSE error metric on the intensity values. The minimization involves two steps. First an error function that is linear in the model

parameters is minimized using least-squares. This error function is then augmented with a nonlinear smoothness constraint, and the least-squares solution is used to bootstrap an iterative nonlinear minimization. This entire procedure is built upon a differential multiscale framework, allowing us to capture both large- and small-scale transformations, see also [57, 50] for related techniques. Details of our technique begin in Chapter 2. Before this detailed description, we first explore some of the earlier work and some techniques used in registration.

1.2.2 Related work

In this section, we explore some of the tools and ideas commonly employed in registration. These tools are often combined with other techniques to form a complete registration algorithm.

Principal axes transform

The principal axes transform is also known as the *Hotelling* transform or *Karhunen-Loéve* transform, and is used to align images where the geometric distortion is assumed to be a rigid-body transformation (i.e., translations and rotations). Given a source image $f(x, y)$ and a target image $g(x, y)$, this model takes the form:

$$f(x, y) = g((x\cos\theta + y\sin\theta) - t_x, (-x\sin\theta + y\cos\theta) - t_y). \tag{1.2}$$

The translation is given by (t_x, t_y) and the rotation angle is θ. In this technique, the pixels (or features) of each image are modeled as points in an ellipse, characterized by a mean position or centroid $\vec{m}$, along with the major and minor axes. The distance between the centers of the two ellipses (corresponding to the two images) gives the translation, while the angle between their major axes (the principal axes) gives the

angle of rotation. The centroid $\vec{m}$ and unit vector $\vec{u}$ that corresponds to the direction of the principal axis of a single ellipse is computed as follows. Let $\vec{x}_i$ denote the coordinates of feature i. The centroid $\vec{m}$ of the ellipse is computed as the mean of $\vec{x}_i$:

$$\vec{m} = \frac{1}{n} \sum_{i=1}^{n} \vec{x}_i \ , \tag{1.3}$$

where n is the total number of features. We next need to find the vector $\vec{u}$ in the direction of the principal axis. Let $\vec{y}_i$ denote the set of points $\vec{x}_i$ with the mean subtracted: $\vec{y}_i = \vec{x}_i - \vec{m}$. The vector $\vec{u}$ corresponding to the principal axis, by definition, has the property that it maximizes the variance of the projection of the points $\vec{y}_i$ onto it. The projection of $\vec{y}_i$ onto the vector $\vec{u}$ is given by: $\left(\frac{\vec{y}_i^T \vec{u}}{\vec{u}^T \vec{u}}\right) \vec{u}$. We can write an error function $E(\vec{u})$ to express the variance of the projections as follows:

$$\begin{aligned} E(\vec{u}) &= \sum_i \left[\left(\frac{\vec{y}_i^T \vec{u}}{\vec{u}^T \vec{u}} \right) \vec{u} \right]^2 \\ &= \sum_i \left[\left(\frac{\vec{y}_i^T \vec{u}}{\vec{u}^T \vec{u}} \right) \vec{u} \right]^T \left[\left(\frac{\vec{y}_i^T \vec{u}}{\vec{u}^T \vec{u}} \right) \vec{u} \right] \\ &= \frac{\vec{u}^T \left(\sum_i \vec{y}_i \vec{y}_i^T \right) \vec{u}}{\vec{u}^T \vec{u}} \\ &= \frac{\vec{u}^T C_{yy} \vec{u}}{\vec{u}^T \vec{u}}. \end{aligned} \tag{1.4}$$

In this equation, C_{yy} is the covariance matrix of the points y_i. This function can be maximized by differentiating $E(\vec{u})$ with respect to $\vec{u}$, setting the result to zero, and solving for $\vec{u}$ as follows:

$$\frac{\partial E(\vec{u})}{\partial \vec{u}} = \frac{2 C_{yy} \vec{u}}{\vec{u}^T \vec{u}} - 2 \vec{u}^T C_{yy} \vec{u} \frac{\vec{u}}{\left(\vec{u}^T \vec{u}\right)^2} = 0. \tag{1.5}$$

Rearranging terms yields:

$$
\begin{aligned}
C_{yy}\vec{u} &= \left(\frac{\vec{u}^T C_{yy}\vec{u}}{\vec{u}^T\vec{u}}\right)\vec{u} \\
&= E(\vec{u})\,\vec{u}. \qquad (1.6)
\end{aligned}
$$

From this equation, we can observe that $\vec{u}$ must correspond to an eigenvector of C_{yy}; note also that $E(\vec{u})$ must be an eigenvalue of C_{yy}. Thus the maximum value of $E(\vec{u})$ corresponds to the largest eigenvalue of C_{yy}, and the corresponding eigenvector is $\vec{u}$. In summary, the eigenvector $\vec{u}$ corresponding to the largest eigenvalue of the covariance matrix C_{yy} is the principal axis of the ellipse, which gives us the direction of maximum variance, and the mean of the points y_i gives us the center of the ellipse.

Once we find the centroids and principal axes for the two ellipses corresponding to both images, the difference between the centroids gives the desired translation vector (t_x, t_y), while the angle between the principal axes gives the desired angle. This technique is analytical and simple to compute. The main drawback is the simplicity of the model. Because of this, the technique is widely used in registration problems that do not require high degrees of accuracy, or when the distortion is simple. The technique may be used as a coarse approximation or pre-processing step for other methods.

Correlation-based matching

Correlation-based approaches are often used for template matching or pattern recognition in which the location (translation) of a template or pattern $f_s(\cdot)$ is found in an image $f_t(\cdot)$. The model used in this case is:

$$
f_s(x, y) = f_t(x - t_x, y - t_y). \qquad (1.7)
$$

The translation parameters (t_x, t_y) are determined by first writing a quadratic error function in t_x and t_y:

$$\begin{aligned} E(t_x, t_y) &= \sum_{x,y \in \Omega} [f_t(x - t_x, y - t_y) - f_s(x, y)]^2 \\ &= \sum_{x,y \in \Omega} f_t(x - t_x, y - t_y)^2 + \sum_{x,y \in \Omega} f_s(x, y)^2 - \\ &\quad 2 \sum_{x,y \in \Omega} f_t(x - t_x, y - t_y) f_s(x, y). \end{aligned} \tag{1.8}$$

where the sum is over the spatial support Ω of $f_s(\cdot)$. This error will be minimized when

$$C(t_x, t_y) = \sum_{x,y \in \Omega} f_t(x - t_x, y - t_y) f_s(x, y) \tag{1.9}$$

is maximal, where $C(\cdot)$ is defined as the correlation coefficient. The maximum of $C(\cdot)$ is therefore a reasonable approximation to the minimum of $E(\cdot)$. This technique is referred to as *unnormalized correlation*, *matched filtering* or *template matching*. The correlation coefficient can be efficiently computed for all possible translations using Fourier techniques [20]. Note that the model used is too simplistic for a general-purpose registration algorithm, and is typically used only for a rough alignment of images. Extensions to cope with rotations and scale changes have also been proposed, but these are sensitive to other types of distortions. A technique that incorporates rotations based on the method of De Castro and Morandi [20] is incorporated in our earlier work [60]. The basic method can also be extended to detect changes in scale (Fourier-Mellin Transform). Cox et al. [23] have implemented a real-time 3-D registration algorithm for shifts and rotations based on the work of Eddy et al. [25], used in the fMRI package AFNI.

Mutual information error metric

Some registration techniques employ an error function that implicitly handles variations of contrast and brightness, the most popular being mutual information. Mutual information is a measure of similarity derived from information theory [22, 71]. The mutual information between two random variables X and Y is given by $I(X;Y)$ and is defined as the relative entropy between the joint probability density $p(X,Y)$ and the product of the marginal probability densities $p(X)p(Y)$:

$$I(X;Y) = \sum_{x \in X} \sum_{y \in Y} p(x,y) \log_2 \frac{p(x,y)}{p(x)p(y)}, \tag{1.10}$$

where x, y are the discrete values of the random variables X, Y. In the context of image registration, the images to be registered are assumed to be the pixel intensities drawn from the random variables X and Y; thus x, y represent the range of pixel intensities drawn from random the variables X, Y. When two images are in perfect registration, their mutual information measure is maximized. Registration is performed by first expressing the chosen model (e.g., rotations, translations) using this error metric, and then maximizing this error with respect to the model parameters. Mutual information provides a mathematical basis for handling multi-modality images directly. In order to obtain good estimates of the mutual information, however, it is necessary to have good estimates of the probability density functions $p(x)$, $p(y)$ of the random variables X and Y, and to have good estimates of the joint probability density $p(x,y)$. Thus, a large number of data points are required, making the measure less accurate in local regions. Another drawback is that mutual information is computationally expensive. Registration methods that use mutual information include those by Viola et al. [71] and Collingnon et al. [21].

Method of tiles for non-rigid registration

Many practical registration tasks require estimation of a non-rigid mapping between images. One straightforward approach for non-rigid registration is to split the source and target images into tiles of equal size, and estimate a rigid mapping between each tile independently. The parameter estimates for each tile are treated as a coarse sampling of the registration map, and interpolation (using a smooth function, such as thin-plate splines) is used to obtain the full registration map at every pixel location. A multiscale approach can also be used, starting the tiling at a coarse scale and refining the results at higher scales. Kostelec et al. [56] use spline pyramids for the hierarchical scheme, and estimate a rigid translation and rotation parameter for each tile, using Bookstein's thin-plate splines for the interpolation. Periaswamy et al. [60] use tiles of different sizes, using a Fourier technique to estimate rigid translation and rotation parameters at each tile. These techniques are straightforward to implement, but have a drawback: since the transformations are estimated at each tile independent of the surrounding tiles, it is possible to get non-smooth behavior between the tiles, where tiles meet. These artifacts are undesirable.

Basis function approach for non-rigid registration

One drawback of the previous non-rigid registration scheme (method of tiles) is that there is no way to impose smoothness of the registration map between tile boundaries (since each tile is treated independently). One approach that avoids this problem is the *basis function* approach. In this approach, the source coordinates (x, y) gets mapped into the new coordinates $(\hat{x}, \hat{y})$ by a basis function as follows:

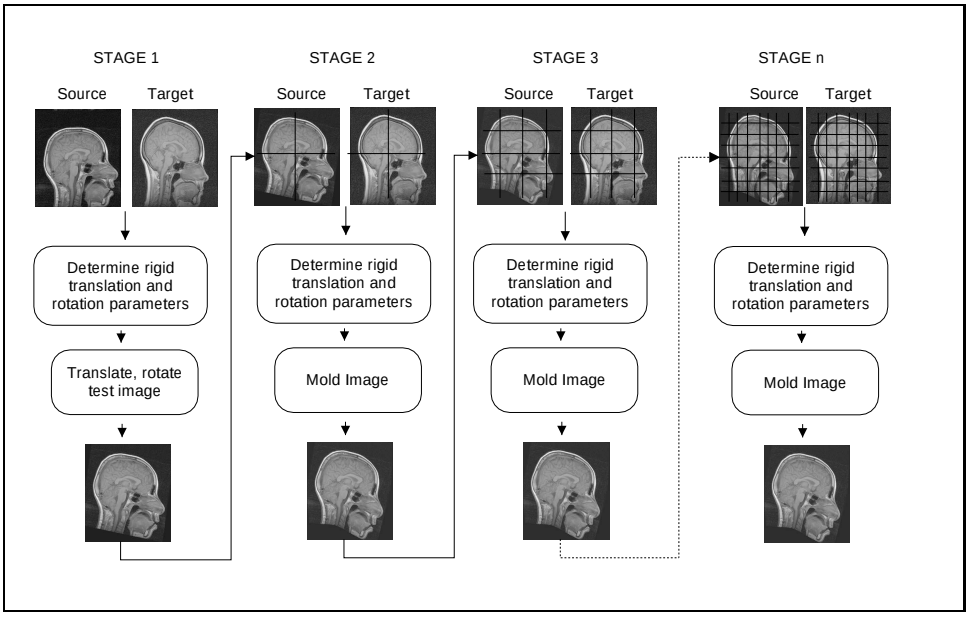


Figure 1.4: Method of tiles, using a multiscale approach (from Periaswamy et al. [60]).

$$\hat{x} = \Delta x + \sum_{j=0}^{N-1} b_j \beta_j(x,y) \tag{1.11}$$

$$\hat{y} = \Delta y + \sum_{j=0}^{N-1} c_j \gamma_j(x,y). \tag{1.12}$$

where Δx, Δy are shifts in x and y, $\beta_j(x,y)$ is the jth basis function for the x-dimension at position (x,y) and $\gamma_j(x,y)$ is the jth basis function for the y-dimension at position (x,y). The coefficients on these basis functions are b_j and c_j, respectively. The basis functions used are typically smooth (for example, the fMRI toolbox SPM uses the lower frequency terms of the DCT basis functions). One advantage of this scheme is that smoothness is implicitly imposed by using smooth basis functions. One disadvantage is that the choice of basis functions may restrict the allowable deformations.

1.2.3 Registration software

SPM

SPM (Statistical Parametric Mapping) is an fMRI toolbox developed by the methodology group at the Wellcome Department of Cognitive Neurology under the supervision of Karl Friston. The registration procedure used in SPM consists of three steps. In the first step, a global affine transform is computed that registers the source and target image to a template image of the same modality, using a Newton-like least-squares method [8]. In the second step, the images are segmented according to tissue types: gray matter, white matter and cerebral-spinal fluid. The gray and white matter partitions are extracted from the images using a modified mixture model cluster analysis method. The model assumes a spatially varying a priori probability of each voxel belonging to a specific tissue type. In the third step, the image segments obtained from step two are registered using a rigid body transformation that simultaneously matches both the gray and white matter partitions from the pair of images. Elastic registration is achieved in the last step by assuming that the warp field can locally be decomposed into a weighted sum of basis functions [9]. The basis functions consist of the low frequency DCT components. A translation term is also estimated at the location of each basis function. Low resolution images are used in the entire process (images are heavily smoothed using a Gaussian window). Estimation of the intensity distortion is also explored in [39, 9] (the current version of SPM - spm99 as of this writing - can estimate these distortions during the segmentation phase).

AIR

AIR (Automated Image Registration) is a registration package developed by R. Woods [77]. The guiding philosophy in AIR is the notion that if two images are in alignment,

the ratio of their intensities remains fairly uniform. Even a slight mis-registration leads to a substantial degree of non-uniformity. The standard deviation of the voxel by voxel ratio measures this non-uniformity; this cost function is called the ratio of image uniformity or RIU. An iterative procedure is used to minimize a normalized form of this quantity in which the registration parameters (three rotation and three translation terms) with the largest partial derivative is adjusted in each iteration [75]. In order to accommodate multi-modality registration (particularly for registration of MRI and PET images), two modifications to the original algorithm were performed [77]. The first modification is to segment the MR image to exclude non-brain structures (e.g., scalp, skull, meninges). The second modification consists of first performing a histogram matching[2] between the MR and PET images (with 256 bins), followed by a segmentation of the images according to the 256 bin values. Each of the segmented MR and PET images (with corresponding bin values) are then registered separately. In both within-modality and cross-modality algorithms, the registration is performed on subsampled images, in decreasing order of subsampling. The most recent versions of AIR (version 3.0 at the time of this writing) allows the user to choose two more cost functions in addition to the RIU cost function [76]. The first additional cost function consists of the mean square error on the raw intensity images, minimized using least-squares. The second additional cost function is also a mean square error on the raw intensity images, but allows global intensity rescaling of the images relative to one another [32, 4]. In addition to improvements with respect to the cost functions, higher order polynomial spatial models have also been implemented (up to a 5th order 3-D spatial transformations, consisting of 168 parameters), and a faster minimization technique has been employed [76].

[2]Denote the two images to be histogram matched as $f_1(\cdot)$ and $f_2(\cdot)$, and $c_2(\cdot)$ as the sampled cumulative distribution function of image $f_2(\cdot)$. The histogram of $f_2(\cdot)$ is made to match that of $f_1(\cdot)$ by mapping each pixel $f_1(x, y)$ to $c_2(f_1(x, y))$.

1.3 Summary

In this chapter, we have introduced the problem of image registration, and highlighted in detail some of the options and challenges faced when designing a registration algorithm. Various aspects of registration have been discussed, beginning with a formal description of the problem, followed by a detailed analysis of the choices we have made, and ending with a discussion of various tools and packages currently used in registration. In developing a registration algorithm, our design choices can be summarized as follows. We choose an intensity-based approach and model the transformation with a local affine model, which is globally smooth. Intensity variations are explicitly modeled with local changes in brightness and contrast and a global smoothness constraint. Registration is performed using a multiscale pyramid scheme. These choices allow us to build a non-rigid registration algorithm that can handle a large number of distortions, including smooth changes in contrast and brightness implicitly, all within a single framework. Details of our registration technique are explained in depth in the next section.

Chapter 2

Methods

2.1 Introduction

In this section, we describe the details of our registration algorithm. This section is split into two main parts; the first part (Section 2.2) describes the basic registration algorithm, while the second part (Section 2.3) explains an extension that allows us to perform registration even when a portion of the images to be registered is occluded. As outlined earlier in Section 1.2, the following design choices have been made. The algorithm is intensity-based; this helps avoid the various pitfalls associated with feature selection. The geometric transform is modeled as smoothly varying affine transformations; this enables us to model a broad range of geometric transformations. Intensity variations are explicitly modeled as smoothly varying contrast and brightness modulations; this avoids a pre-processing step for intensity corrections in the registration, and allows a larger class of images to be registered. Both the geometric transformation and intensity modulation parameters are estimated simultaneously for each pixel location. The error function used is the mean square error (MSE) on the intensity values. With these design choices in mind, a framework for registra-

tion is developed, including a discussion of the practical design issues necessary for a successful implementation.

2.2 Registration algorithm

The problem of image registration between a source and target image is formulated within a differential (non-feature-based) framework. This formulation borrows from various areas of motion estimation [37, 42, 73, 2, 34, 3, 33, 12]. Estimation of the registration parameters is described in three stages. In the first stage, the transformation is formulated as being purely affine (Section 2.2.1). In the second stage, this purely affine geometric transform is extended to also implicitly account for contrast and brightness modulations (Section 2.2.2). Finally, in the third stage, a smoothness constraint is imposed on all the locally estimated geometric and intensity parameters (Section 2.2.3). Following this description, we also explain the necessary details to make the algorithm work in practice (Section 2.2.4), along with and extension for 3-D volumes (Section 2.2.6).

2.2.1 Local affine

Denote $f(x, y, t)$ and $f(\hat{x}, \hat{y}, t-1)$ as the source and target images, respectively. [1] We begin by assuming that the image intensities between images are conserved (this assumption will be relaxed later), and that the motion between images can be modeled locally by an affine transform:

$$f(x, y, t) = f(m_1 x + m_2 y + m_5, m_3 x + m_4 y + m_6, t-1), \tag{2.1}$$

[1] We adopt the slightly unconventional notation of denoting the source and target image with a temporal parameter t. This is done for consistency within our differential formulation.

where m_1, m_2, m_3, m_4 are the linear affine parameters, and m_5, m_6 are the translation parameters. These parameters are estimated locally for each small spatial neighborhood, but for notational convenience their spatial parameters are dropped. In order to estimate these parameters, we define the following quadratic error function to be minimized:

$$E(\vec{m}) = \sum_{x,y \in \Omega} \left[f(x,y,t) - f(m_1 x + m_2 y + m_5, m_3 x + m_4 y + m_6, t-1) \right]^2, \qquad (2.2)$$

where $\vec{m} = \begin{pmatrix} m_1 & \dots & m_6 \end{pmatrix}^T$, and Ω denotes a small spatial neighborhood. Since this error function is nonlinear in its unknowns, it cannot be minimized analytically. To simplify the minimization, we approximate this error function using a first-order truncated Taylor series expansion:

$$\begin{aligned} E(\vec{m}) \approx \sum_{x,y \in \Omega} & \left(f(x,y,t) - [f(x,y,t) + (m_1 x + m_2 y + m_5 - x) f_x(x,y,t) + \right. \\ & \left. (m_3 x + m_4 y + m_6 - y) f_y(x,y,t) - f_t(x,y,t)] \right)^2, \end{aligned} \qquad (2.3)$$

where $f_x(\cdot)$, $f_y(\cdot)$, $f_t(\cdot)$ are the spatial/temporal derivatives of $f(\cdot)$. This error function further reduces to:

$$\begin{aligned} E(\vec{m}) \approx \sum_{x,y \in \Omega} & \left[f_t(x,y,t) - (m_1 x + m_2 y + m_5 - x) f_x(x,y,t) - \right. \\ & \left. (m_3 x + m_4 y + m_6 - y) f_y(x,y,t) \right]^2. \end{aligned} \qquad (2.4)$$

Note that this quadratic error function is now linear in its unknowns, $\vec{m}$. This error function may be expressed more compactly in vector form as:

$$E(\vec{m}) = \sum_{x,y \in \Omega} \left[k - \vec{c}^{\,T} \vec{m} \right]^2, \qquad (2.5)$$

where the scalar k and vector $\vec{c}$ are given as:

$$k = f_t + xf_x + yf_y \tag{2.6}$$

$$\vec{c} = (xf_x \quad yf_x \quad xf_y \quad yf_y \quad f_x \quad f_y)^T, \tag{2.7}$$

where again, for notational convenience, the spatial/temporal parameters of $f_x(\cdot)$, $f_y(\cdot)$, and $f_t(\cdot)$ are dropped. This error function can now be minimized analytically by differentiating with respect to the unknowns:

$$\frac{dE(\vec{m})}{d\vec{m}} = \sum_{x,y\in\Omega} -2\vec{c}\left[k - \vec{c}^{\,T}\vec{m}\right], \tag{2.8}$$

setting this result equal to zero, and solving for $\vec{m}$ yielding:

$$\vec{m} = \left[\sum_{x,y\in\Omega} \vec{c}\vec{c}^{\,T}\right]^{-1}\left[\sum_{x,y\in\Omega} \vec{c}k\right]. \tag{2.9}$$

This solution assumes that the first term, a 6×6 matrix, is invertible. This can usually be guaranteed by integrating over a large enough spatial neighborhood Ω with sufficient image content. With this approach a dense locally affine mapping can be found between a source and target image.

2.2.2 Intensity variations

Inherent in the model outlined in the previous section is the assumption that the image intensities between the source and target are unchanged, i.e., the assumption of brightness constancy. This assumption is likely to fail under a number of circumstances. To account for intensity variations, we incorporate into our model an explicit change of local contrast and brightness [59]. Specifically, our initial model,

Equation (2.1), now takes the form:

$$m_7 f(x,y,t) + m_8 \;=\; f(m_1 x + m_2 y + m_5, m_3 x + m_4 y + m_6, t-1), \quad (2.10)$$

where m_7 and m_8 are two new (also spatially varying) parameters that embody a change in contrast and brightness, respectively. Note that these parameters have been introduced in a linear fashion. As before, this error function is approximated with a first-order truncated Taylor series expansion to yield:

$$E(\vec{m}) \;=\; \sum_{x,y \in \Omega} \left[k - \vec{c}^{\,T} \vec{m} \right]^2, \quad (2.11)$$

where the scalar k and vector $\vec{c}$ are now given as:

$$k \;=\; f_t - f + x f_x + y f_y \quad (2.12)$$

$$\vec{c} \;=\; (x f_x \quad y f_x \quad x f_y \quad y f_y \quad f_x \quad f_y \quad -f \quad -1)^T. \quad (2.13)$$

Minimizing this error function is accomplished as before by differentiating $E(\vec{m})$, setting the result equal to zero and solving for $\vec{m}$. The solution takes the same form as in Equation (2.9), with k and $\vec{c}$ as defined above.

Intensity variations are typically a significant source of error in differential motion estimation. The addition of the contrast and brightness terms allows us to accurately register images in the presence of local intensity variations. It is possible, of course, to fully explain the mapping between images with only a brightness modulation. In the next section we describe how to avoid such a degenerate solution.

2.2.3 Smoothness

Until now, we have assumed that the local affine and contrast/brightness parameters are constant within a small spatial neighborhood, Equation (2.11). There is a natural

trade-off in choosing the size of this neighborhood. A larger area makes it more likely that the matrix $\sum_{x,y\in\Omega} \vec{c}\vec{c}^{\,T}$ in Equation (2.9) will be invertible. A smaller area, however, makes it more likely that the brightness constancy assumption will hold. We can avoid balancing these two issues by replacing the constancy assumption with a smoothness assumption [34]. That is, we assume that the model parameters $\vec{m}$ vary smoothly across space. A smoothness constraint on the contrast/brightness parameters has the added benefit of avoiding a degenerate solution where a pure brightness modulation is used to describe the mapping between images. To begin, we augment the error function in Equation (2.11) as follows:

$$E(\vec{m}) \;=\; E_b(\vec{m}) + E_s(\vec{m}), \tag{2.14}$$

where $E_b(\vec{m})$ is defined as in Equation (2.11) without the summation:

$$E_b(\vec{m}) \;=\; \left[k - \vec{c}^{\,T}\vec{m}\right]^2, \tag{2.15}$$

with k and $\vec{c}$ as in Equations (2.12) and (2.13). The new quadratic error term $E_s(\vec{m})$ embodies the smoothness constraint:

$$E_s(\vec{m}) \;=\; \sum_{p=1}^{8} \lambda_p \left[\left(\frac{\partial m_p}{\partial x} \right)^2 + \left(\frac{\partial m_p}{\partial y} \right)^2 \right], \tag{2.16}$$

where λ_p is a positive constant that controls the relative weight given to the smoothness constraint on parameter m_p. This error function is again minimized by differentiating, setting the result equal to zero and solving, $dE\left(\vec{m}\right)/d\vec{m} = dE_b\left(\vec{m}\right)/d\vec{m} + dE_s\left(\vec{m}\right)/d\vec{m} = 0$. The derivative of $E_b(\vec{m})$ is:

$$\frac{dE_b\left(\vec{m}\right)}{d\vec{m}} \;=\; -2\vec{c}\left[k - \vec{c}^{\,T}\vec{m}\right]. \tag{2.17}$$

To compute the derivative of $\frac{dE_s(\vec{m})}{d\vec{m}}$, we first use discrete approximations to the $x-$ and $y-$ derivatives of m_p (as explained in [34]) as follows:

$$\frac{dm_p(i,j)}{dx} = m_p(i,j) - m_p(i+1,j) \tag{2.18}$$

$$\frac{dm_p(i,j)}{dy} = m_p(i,j) - m_p(i,j+1). \tag{2.19}$$

We can now express the smoothness error of a particular parameter m_p as follows:

$$\begin{aligned} E_s\left(m_p\left(i,j\right)\right) &= \lambda_p \left[\left(\frac{\partial m_p\left(i,j\right)}{\partial x} \right)^2 + \left(\frac{\partial m_p\left(i,j\right)}{\partial y} \right)^2 \right] \\ &= \lambda_p \left[\left(m_p(i,j) - m_p(i+1,j)\right)^2 + \left(m_p(i,j) - m_p(i,j+1)\right)^2 \right]. \end{aligned} \tag{2.20}$$

The derivative of this smoothness error of term p with respect to the parameter can now be written as:

$$\begin{aligned} \frac{dE_s\left(m_p\left(i,j\right)\right)}{dm_p\left(i,j\right)} &= 2\lambda_p \left[m_p(i,j) - m_p(i+1,j)\right] + 2\left[m_p(i,j) - m_p(i,j+1)\right] \\ &= 4\lambda_p m_p(i,j) - 2\lambda_p \left[m_p\left(i+1,j\right) + m_p\left(i,j+1\right)\right] \\ &= 4\lambda_p \left[m_p\left(i,j\right) - \bar{m}_p\left(i,j\right)\right], \end{aligned} \tag{2.21}$$

where $\bar{m}_p\left(i,j\right)$ denotes $\left(m_p\left(i+1,j\right) + m_p\left(i,j+1\right)\right)/2$, or the local average of $\bar{m}_p$ about the point (i,j). Using vector notation, we can succinctly represent the smoothness derivative as:

$$\frac{dE_s\left(\vec{m}\right)}{d\vec{m}} = 2L(\overline{\vec{m}} - \vec{m}), \tag{2.22}$$

where $\overline{\vec{m}}$ is the component-wise average of $\vec{m}$ over a small spatial neighborhood, and L is an 8×8 diagonal matrix with diagonal elements λ_i, and zero off the diagonal.

Setting $dE_b(\vec{m})/d\vec{m} + dE_s(\vec{m})/d\vec{m} = 0$, and solving for $\vec{m}$ at each pixel location yields an enormous linear system which is intractable to solve. Instead $\vec{m}$ is expressed in the following iterative form [34]:

$$\vec{m}^{(j+1)} = \left(\vec{c}\vec{c}^{\,T} + L\right)^{-1}\left(\vec{c}k + L\overline{\vec{m}}^{(j)}\right). \tag{2.23}$$

On each iteration j, $\overline{\vec{m}}^{(j)}$ is estimated from the current $\vec{m}^{(j)}$. The initial estimate $\vec{m}^{(0)}$ is estimated from the closed-form solution of Section 2.2.2.

The use of a smoothness constraint has the benefit that it yields a dense locally affine but globally smooth transformation. The drawback is that the minimization is no longer analytic. We have found, nevertheless, that the iterative minimization is quite stable and converges relatively quickly (on the order of forty iterations, for the results presented here).

2.2.4 Implementation details

While the formulation given in the previous sections is relatively straightforward there are a number of implementation details that are critical for a successful implementation.

First, in order to simplify the minimization, the error function of Equation (2.15) was derived through a Taylor-series expansion. A more accurate estimate of the actual error function can be determined using a Newton-Raphson style iterative scheme [62]. In particular, on each iteration, the estimated transformation is applied to the source image, and a new transformation is estimated between the newly warped source and target image. As few as five iterations greatly improves the final estimate.

Second, the required spatial/temporal derivatives have finite support, thus fundamentally limiting the amount of motion that can be estimated. A coarse-to-fine

scheme is adopted in order to contend with larger motions [42, 5]. A Gaussian pyramid [1] is built for both source and target images, and the local affine and contrast/brightness parameters estimated at the coarsest level. These parameters are used to warp the source image in the next level of the pyramid. A new estimate is computed at this level, and the process repeated through each level of the pyramid. The transformations at each level of the pyramid are accumulated yielding a single final transformation. This multiscale approach is critical given the differential nature of our measurements, allowing us to register images with large motions.

Finally, the estimation of the spatial/temporal derivatives is a crucial step. Spatial/temporal derivatives of discretely sampled images are often computed as differences between neighboring sample values. Such differences are typically poor approximations to derivatives and lead to substantial errors. In computing derivatives we employ a set of derivative filters specifically designed for multi-dimensional differentiation [27]. The spatial/temporal derivatives are estimated as follows. The images are first pre-filtered in time (using the two-tap filter [0.5 0.5]). The derivative in x is then estimated by first pre-filtering the result in y (using the 3-tap prefilter [0.223755 0.552490 0.223755]), followed by differentiating in x (using the derivative filter [-0.453014 0.0 0.453014]). Similarly, the derivative in y is estimated by first pre-filtering the result in x (using the 3-tap prefilter [0.223755 0.552490 0.223755]), followed by differentiating in y (using the derivative filter [-0.453014 0.0 0.453014]). The derivative in time is estimated by by first pre-filtering in space (in x and y) using the 3-tap prefilter [0.223755 0.552490 0.223755]), followed by applying the two-tap derivative filter [0.5 -0.5] in time to the result. These filters significantly improve the resulting registration. With these implementation details in mind, the complete system for the registration algorithm is presented next.

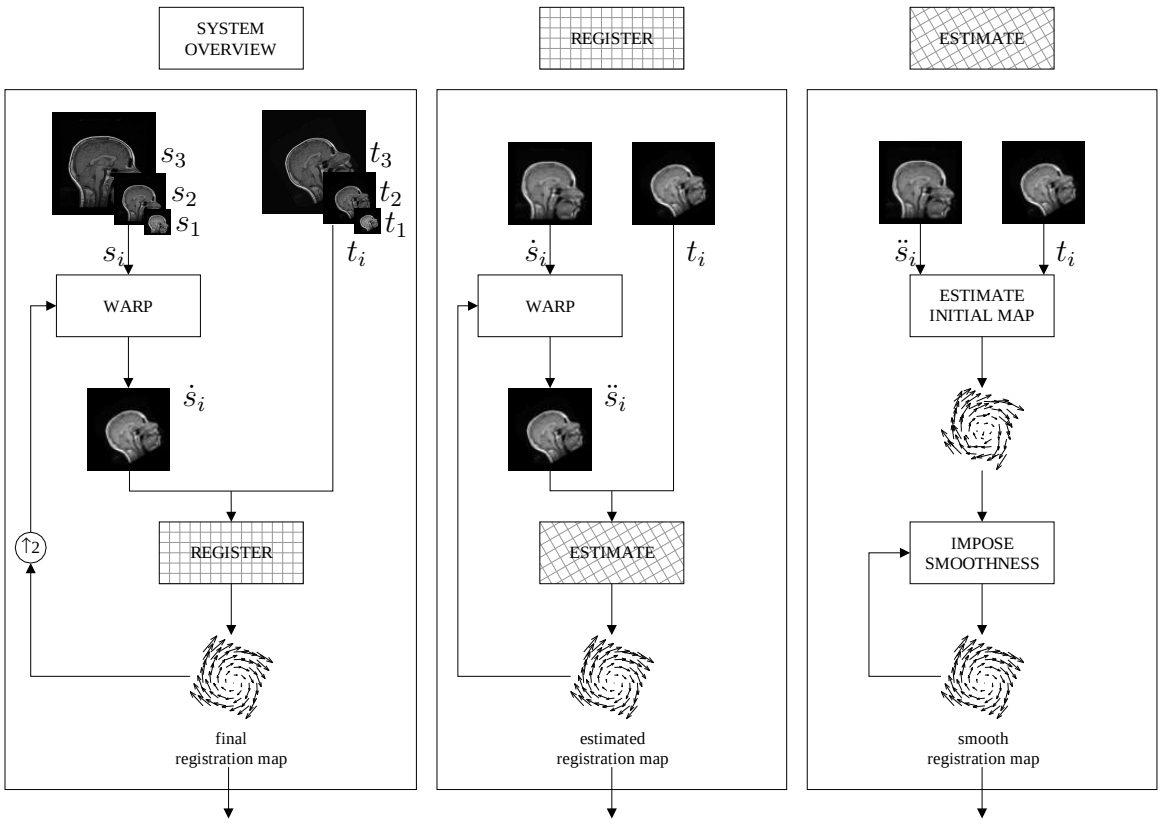


Figure 2.1: System overview for the registration algorithm, split into three sections, depicted by the three columns in the figure. The first column depicts the multiscale estimation of the registration map, from a coarse to fine scale. The second column depicts the estimation of the registration map within each scale. The final column depicts the process by which a smooth registration map is obtained. In the first two columns, the algorithm is bootstrapped with an initial registration map consisting entirely of zeros.

2.2.5 System overview

The multiscale registration algorithm proceeds as follows (see first column in Figure 2.1). A Gaussian pyramid is first built for both the source and target images. The source and target images at the coarsest scale are then registered to obtain an initial estimation of the registration map. This initial estimate is used to warp the source image at the next scale. The warped source image is then registered with its corresponding target image. This process is repeated at each level of the pyramid. A

single registration map is maintained by accumulating successive estimated registration maps at each scale. This multiscale approach allows us to recover large motions as well as small motions.

Within each scale, the registration map is determined in an iterative fashion as follows (see the second column in Figure 2.1). After an initial estimation of the registration parameters, the source image is warped with the estimated parameters, and registered again with the target image. During each of these iterations (referred to as accuracy-iterations), successive intermediate registration maps are accumulated to form a single registration map. The iterations are stopped when the average displacement of the estimated motion is less than 0.1 pixels. This iterative approach (see also [62]) helps improve the accuracy of the registration, as it leads to a more accurate Taylor-series approximation of the error function (Equation (2.3)).

Within each scale, and within each accuracy-iteration, a smooth registration map is obtained as follows (see the third column in Figure 2.1). Given a source and target image, an estimate of the registration map without smoothness is first obtained. This initial estimate is used to bootstrap the nonlinear iterative estimation of a smooth registration map. These iterations are referred to as smoothness-iterations. These three components together form the complete registration algorithm. Note that the algorithm is bootstrapped with an initial registration map consisting of all zeros.

2.2.6 Extension to 3-D

The extension of this algorithm from 2-D images to 3-D volumes is relatively straightforward.

Local affine model

Denote $f(x, y, z, t)$ and $f(\hat{x}, \hat{y}, \hat{z}, t-1)$ as the source and target volumes, respectively. Making the same local affine assumption as in Equation (2.1) yields:

$$\begin{aligned} f(x, y, z, t) &= f(m_1 x + m_2 y + m_3 z + m_{10}, m_4 x + m_5 y + m_6 z + m_{11}, \\ & \qquad m_7 x + m_8 y + m_9 z + m_{12}, t-1), \end{aligned} \tag{2.24}$$

where $\vec{m} = (m_1, ..., m_{12})$ consists of the linear affine parameters $(m_1, ..., m_9)$, and the translation parameters (m_{10}, m_{11}, m_{12}). As before, we define an error function $E(\vec{m})$, approximate it with a first-order truncated Taylor series expansion, differentiate with respect to the unknowns, set the result equal to zero and solve to obtain:

$$\vec{m} = \left[\sum_{x,y,z \in \Omega} \vec{c}\,\vec{c}^{\,T} \right]^{-1} \left[\sum_{x,y,z \in \Omega} \vec{c}\,k \right], \tag{2.25}$$

where $\vec{m}$ is now of size 1×12, and k and $\vec{c}$ are defined as follows:

$$k = f_t + x f_x + y f_y + x f_z \tag{2.26}$$

$$\vec{c} = \begin{pmatrix} x f_x & y f_x & z f_x & x f_y & y f_y & z f_y & x f_z & y f_z & z f_z & f_x & f_y & f_z \end{pmatrix}^T, \tag{2.27}$$

where the spatial/temporal derivatives of $f(\cdot)$ are $f_x(\cdot)$, $f_y(\cdot)$, $f_z(\cdot)$, $f_t(\cdot)$. Again for notational convenience, the spatial parameters are dropped.

Intensity variations

The contrast and brightness terms (m_{13} and m_{14}) are derived as in the 2-D case, Equations (2.10) and (2.11). The resulting solution is once again in the same form as in Equation (2.25), with $\vec{m}$ now a vector with 14 elements, and k and $\vec{c}$ defined as:

$$k = f_t - f + xf_x + yf_y + xf_z \tag{2.28}$$

$$\vec{c} = \begin{pmatrix} xf_x & yf_x & zf_x & xf_y & yf_y & zf_y & xf_z & yf_z & zf_z & f_x & f_y & f_z & -f & -1 \end{pmatrix}^T . \tag{2.29}$$

Smoothness

Smoothness is incorporated as before (Section 2.2.3) by augmenting the error function $E(\vec{m})$ with a smoothness term:

$$E(\vec{m}) = E_b(\vec{m}) + E_s(\vec{m}), \tag{2.30}$$

where $E_b(\vec{m})$ is the error function from Section 2.2.6 given by:

$$E_b(\vec{m}) = \left[k - \vec{c}^T \vec{m}\right]^2, \tag{2.31}$$

with k and $\vec{c}$ as in Equations (2.28) and (2.29), and $E_s(\vec{m})$ is:

$$E_s(\vec{m}) = \sum_{i=1}^{14} \lambda_i \left[\left(\frac{\partial m_i}{\partial x}\right)^2 + \left(\frac{\partial m_i}{\partial y}\right)^2 + \left(\frac{\partial m_i}{\partial z}\right)^2 \right]. \tag{2.32}$$

An estimate of $\vec{m}$ is obtained once again by differentiating $E(\vec{m})$, setting the result equal to zero, and solving, giving the same iterative solution as in Equation (2.23):

$$\vec{m}^{(j+1)} = \left(\vec{c}\vec{c}^T + L\right)^{-1} \left(\vec{c}k + L\overline{\vec{m}}^{(j)}\right), \tag{2.33}$$

with k and $\vec{c}$ given by Equations (2.28) and (2.29). L is a 14×14 diagonal matrix with diagonal elements λ_i, and zero off the diagonal.

2.3 Registration with partial data

2.3.1 Introduction

When performing registration in practice, it is possible that either the source or target image has only partial data (i.e., some portion of the data is missing or occluded). This situation arises commonly in many practical registration cases. For example if a subject moves, certain regions may go out of the view of the scanning device, causing an incomplete scan. Deliberate occlusions can occur when an image has been modified so as to keep the patient's identity anonymous, as shown in Figure 2.2 (the fMRI data center[2] at Dartmouth College stores brain volumes in this format). Most registration algorithms are simply ill equipped to handle missing data, and usually completely fail. Given the common occurrence of this problem, it would be of great practical importance to register such images. In this chapter, we provide an extension to our basic registration algorithm that implicitly handles missing data. The extension involves the well known Expectation Maximization (EM) algorithm. We first outline the EM algorithm, and then describe its incorporation into our registration algorithm.

2.3.2 The Expectation Maximization algorithm

Expectation Maximization (EM) is a useful technique that can be applied to maximum-likelihood parameter estimation problems. The algorithm was discovered and employed independently by several different researchers. A seminal paper written by Dempster [6] brought all these ideas together, proved convergence, and coined the term "EM algorithm". Since then, EM has become a staple of maximum-likelihood-based techniques. In this section, we first briefly outline the EM algorithm in the

[2] `http://www.fmridc.org`

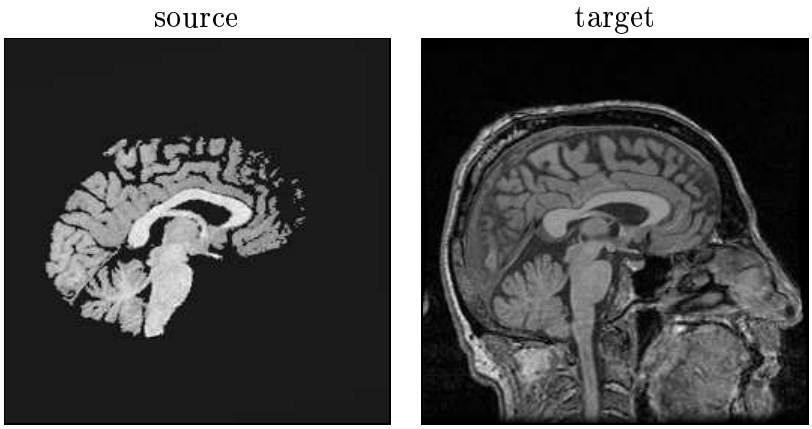


Figure 2.2: In some applications, the images to be registered may only contain partial data. For example, the source image shown above consists of a sagital image with the skull stripped, while the target image consists of a complete sagital image from a different person. It would be useful if registration could still be performed under these circumstances.

context of a simple example, and then show how this technique can be used to register images that contain only partial data.

EM Example

We are given a set of n data points $\vec{q}(i) = (x(i), y(i))$, $i = 1, \cdots, n$, and are told that the points come from two lines of the form:

$$y(i) = a_1 x(i) + b_1 + n_1(i) \text{ or } y(i) = a_2 x(i) + b_2 + n_2(i), \tag{2.34}$$

where the model parameters for each line are $M_1 = (a_1, b_1)$, and $M_2 = (a_2, b_2)$; $n_1(i)$, $n_2(i)$ consists of white Gaussian noise. Figure 2.3 shows a sample instance of data points. The problem is to estimate the model parameters M_1, M_2 (i.e., fit the data) and determine which model each data point belongs to (i.e., segment the data). Note that if we are given the model parameters, we can easily segment each of the data points $\vec{q}(i)$ by computing a residual $r_k(i) = |a_k x(i) + b_k - y(i)|$ for each

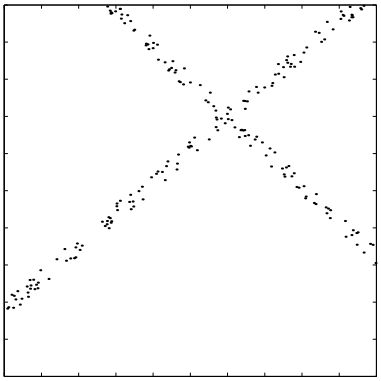


Figure 2.3: Data from two line models corrupted with noise.

model k, and choosing the model that minimizes this residual for that point. On the other hand, if we were told which points belonged to which model, then the model parameters can be fit using its corresponding points by solving the over-constrained set of linear equations:

$$\begin{bmatrix} x_k(1) & 1 \\ x_k(2) & 1 \\ \vdots & \vdots \\ x_k(n) & 1 \end{bmatrix} \begin{bmatrix} a_k \\ b_k \end{bmatrix} = \begin{bmatrix} y_k(1) \\ y_k(2) \\ \vdots \\ y_k(n) \end{bmatrix}, \tag{2.35}$$

where all $x_k(\cdot)$ and $y_k(\cdot)$ belong to model k ($k = 1, 2$). This yields the following least-squares solution:

$$\begin{bmatrix} a_k \\ b_k \end{bmatrix} = \begin{bmatrix} \sum_i x(i)^2 & \sum_i x(i) \\ \sum_i x(i) & \sum_i 1 \end{bmatrix}^{-1} \begin{bmatrix} \sum_i x(i)y(i) \\ \sum_i y(i) \end{bmatrix}. \tag{2.36}$$

However, in our sample problem, we need to simultaneously estimate the parameters as well as segment the data, making this a much harder problem to solve. One approach to solving this problem is to use EM, a two-step iterative procedure described as follows. Initially, a set of random model parameters are chosen. In the first step, the probability that each point $\vec{q}(i)$ belongs to model k is estimated as weight $w_k(i)$.

In the second step, the model parameters are updated using a weighted least-squares estimate. These two steps are iterated until convergence of the estimated parameters. A detailed derivation of this process follows.

Complete data likelihood

The *complete data likelihood* $L(M_1, M_2)$ of observing all n points $\vec{q}(1),\ \vec{q}(2),\ \vec{q}(3),\ \cdots\ \vec{q}(n)$ with model parameters M_1 and M_2 is given by:

$$L(M_1, M_2) = \prod_i p(Q = \vec{q}(i)), \tag{2.37}$$

where each point is considered independent of each other. Here, $p(Q = \vec{q}(i))$ is the probability of observing point $\vec{q}(i)$; Q is a random variable. Using the law of total probability, we can rewrite this probability as:

$$p(Q = \vec{q}(i)) = p(Q = \vec{q}(i)|Q \in M_1)\, P(Q \in M_1) + p(Q = \vec{q}(i)|Q \in M_2)\, P(Q \in M_2), \tag{2.38}$$

where $p(Q \in M_k)$ is the probability of model k. Assume for simplicity that in our example, it is equally likely that the points came from either model. We assume the distributions of the residuals are Gaussian, which leads to the following probabilities:

$$p(Q = \vec{q}(i)|M_1) = e^{-(a_1 x(i) + b_1 - y(i))^2/\sigma_1^2} = e^{-r_1(i)^2} \tag{2.39}$$

$$p(Q = \vec{q}(i)|M_2) = e^{-(a_2 x(i) + b_2 - y(i))^2/\sigma_2^2} = e^{-r_2(i)^2}. \tag{2.40}$$

Here, $p(Q = \vec{q}(i)|M_k)$ is the probability of observing point $\vec{q}(i)$ assuming that it came from model M_k, and σ_k is a constant and is proportional to the amount of noise in model k. The log-likelihood function (which is easier to maximize) can be written as:

$$\begin{aligned}
\log L\left(M_1, M_2\right) &= \log \prod_i p\left(Q = \vec{q}(i)\right) \\
&= \sum_i \log\left(p\left(Q = \vec{q}(i)\right)\right) \\
&= \sum_i \log\left(p\left(Q = \vec{q}(i) | Q \in M_1\right) P\left(Q \in M_1\right) + p\left(Q = \vec{q}(i) | Q \in M_2\right) P\left(Q \in M_2\right)\right) \\
&= \log\left(P\left(Q \in M_1\right)\right) + \sum_i \log\left(p\left(Q = \vec{q}(i) | Q \in M_1\right) + p\left(Q = \vec{q}(i) | Q \in M_2\right)\right) \\
&= c + \sum_i \log\left(e^{-\left(a_1 x(i) + b_1 - y(i)\right)^2} + e^{-\left(a_2 x_i + b_2 - y(i)\right)^2}\right). \qquad (2.41)
\end{aligned}$$

The last two equations assume the priors $P\left(Q \in M_1\right) = P\left(Q \in M_2\right)$ and $c = \log P\left(Q \in M_1\right)$.

Solving for model parameters

In order to determine the model parameters $M_k = \left(a_k, b_k\right)$, we maximize the log-likelihood function by differentiating with respect these parameters, setting equal to zero, and solve for the parameters. Differentiating the log-likelihood function with respect to a_k yields:

$$\begin{aligned}
\frac{d \log L\left(M_1, M_2\right)}{d a_k} &= \sum_i \frac{-2x(i)\left(a_k x(i) + b_k - y(i)\right) e^{-\left(a_k x(i) + b_k - y(i)\right)^2}}{e^{-\left(a_1 x(i) + b_k - y(i)\right)^2} + e^{-\left(a_2 x(i) + b_k - y(i)\right)^2}} \\
&= \sum_i \frac{-2x(i)\left(a_k x(i) + b_k - y(i)\right) e^{-r_k^2(i)}}{e^{-r_1(i)^2} + e^{-r_2(i)^2}} \\
&= \sum_i -2x(i)\left(m_k x(i) + b_k - y(i)\right) w_k\left(i\right), \qquad (2.42)
\end{aligned}$$

where $w_k(i) = \frac{e^{-r_k(i)^2/\sigma_k^2}}{e^{-r_1(x,y)^2/\sigma_1^2} + e^{-r_2(x,y)^2/\sigma_2^2}}$, and is the probability that point i belongs to model k. Differentiating the log-likelihood function with respect to b_k yields:

$$\begin{aligned}
\frac{d\log L\left(M_1,M_2\right)}{db_k} &= \sum_i \frac{-2\left(a_k x(i)+b_k-y(i)\right)e^{-\left(a_k x(i)+b_k-y(i)\right)^2}}{e^{-\left(a_1 x(i)+b_k-y(i)\right)^2}+e^{-\left(a_2 x(i)+b_k-y(i)\right)^2}} \\
&= \sum_i \frac{-2\left(a_k x(i)+b_k-y(i)\right)e^{-r_1(i)^2}}{e^{-r_1(i)^2}+e^{-r_2(i)^2}} \\
&= \sum_i -2\left(a_k x(i)+b_k-y(i)\right)w_k\left(i\right). \qquad (2.43)
\end{aligned}$$

Setting Equations (2.42) and (2.43) to zero and rearranging terms yields the following weighted least-squares problem:

$$\left[\begin{array}{cc} \sum_i x_i^2\cdot w_k\left(i\right) & \sum_i x_i\cdot w_k\left(i\right) \\ \sum_i x_i\cdot w_k\left(i\right) & \sum_i 1\cdot w_k\left(i\right)\end{array}\right]\left[\begin{array}{c} a_k \\ b_k\end{array}\right]=\left[\begin{array}{c} \sum_i x_i y_i\cdot w_k\left(i\right) \\ \sum_i y_i\cdot w_k\left(i\right)\end{array}\right], \qquad (2.44)$$

from which the model parameters a_k, b_k can be determined as:

$$\left[\begin{array}{c} a_k \\ b_k\end{array}\right]=\left[\begin{array}{cc} \sum_i x_i^2\cdot w_k\left(i\right) & \sum_i x_i\cdot w_k\left(i\right) \\ \sum_i x_i\cdot w_k\left(i\right) & \sum_i 1\cdot w_k\left(i\right)\end{array}\right]^{-1}\left[\begin{array}{c} \sum_i x_i y_i\cdot w_k\left(i\right) \\ \sum_i y_i\cdot w_k\left(i\right)\end{array}\right]. \qquad (2.45)$$

Note that this equation is similar to Equation (2.36) without the weights $w_k\left(i\right)$.

EM algorithm

The EM algorithm that estimates the model parameters M_1 and M_2 along with the probability assignments for each point can now be explained as a two step process as follows:

1. Initialization: Assign values for the fixed constants σ_1^2, σ_2^2. Assign initial estimates for the model parameters a_1, a_2, b_1, b_2.

2. E-Step: Compute the weights:

   $w_1(i) = \frac{e^{-r_1(i)^2/\sigma_1^2}}{e^{-r_1(i)^2/\sigma_1^2}+e^{-r_1(i)^2/\sigma_2^2}}$, and $w_2(i) = \frac{e^{-r_2(i)^2/\sigma_2^2}}{e^{-r_1(i)^2/\sigma_1^2}+e^{-r_2(i)^2/\sigma_2^2}}$.

   This corresponds to the segmentation step.

3. M-Step: Use the weights $w_1(i)$ and $w_2(i)$ to estimate the least-squares solutions for a_k, b_k:

$$\begin{bmatrix} a_k \\ b_k \end{bmatrix} = \begin{bmatrix} \sum_i x_i^2 \cdot w_k(i) & \sum_i x_i \cdot w_k(i) \\ \sum_i x_i \cdot w_k(i) & \sum_i 1 \cdot w_k(i) \end{bmatrix}^{-1} \begin{bmatrix} \sum_i x_i y_i \cdot w_k(i) \\ \sum_i y_i \cdot w_k(i) \end{bmatrix}.$$

   This corresponds to the estimation step.

Steps 1 and 2 are repeated until convergence of the estimated parameters. Though this algorithm is guaranteed to converge [6], it is susceptible to local minima. In order to increase the likelihood of finding the correct global minimum, the algorithm can be restarted with different starting points. In this manner, both the model parameters can be found along with a fit of the data. In the next section, we explain how this EM algorithm is employed to deal with registration in the presence of partial data.

2.3.3 Extension for registration with partial data

In the previous sample problem, we showed how to simultaneously segment the data and find the model parameters that fit the data. In this section, we show how this problem is mapped to our registration problem, i.e., registration in the presence of an occlusion or missing data. Inherent to the registration algorithm is the assumption that each region in the source image has a corresponding match in the target image. As illustrated in Figure 2.2, this need not always be the case. Under such situations, our registration algorithm typically fails. One way to contend with partial or missing data is to employ a pre-processing segmentation step. We propose, however,

a more unified approach in which the registration and segmentation are performed simultaneously.

We begin by assuming that each pixel in the source and target are either related through the intensity and geometric model of Equation (2.10), denoted as model M_1, or cannot be explained by this transformation and therefore belongs to an "outlier" model M_2. Pixels belonging to the outlier model are those that do not have a corresponding match between the source and target images. Assuming that the pixels are spatially independent and identically distributed (iid), the likelihood of observing a pair of images is given by:

$$L(\vec{m}) = \prod_{x,y\in\Omega} P(\vec{q}(x,y)), \tag{2.46}$$

where, $\vec{q}(x,y)$ denotes the tuple of source and target image intensities $(m_7 f(x,y,t) + m_8, f(m_1x + m_2y + m_5, m_3x + m_4y + m_6, t-1))$, Equation (2.10). To simplify the optimization of the likelihood function, we consider the log-likelihood function:

$$\begin{aligned}\log[L(\vec{m})] &= \log\left[\prod_{x,y\in\Omega} P(\vec{q}(x,y))\right] \\ &= \sum_{x,y\in\Omega} \log\left[P(\vec{q}(x,y)|M_1)P(M_1) + P(\vec{q}(x,y)|M_2)P(M_2)\right]. \end{aligned}\tag{2.47}$$

Assuming that the priors on the models, $P(M_1)$ and $P(M_2)$, are equal, the log-likelihood function simplifies to:

$$\log[L(\vec{m})] = \sum_{x,y\in\Omega} \log\left[P(\vec{q}(x,y)|M_1) + P(\vec{q}(x,y)|M_2)\right], \tag{2.48}$$

where the factored additive constant is ignored for purposes of maximization. We assume next that the conditional probabilities take the following form.

$$\log[L(\vec{m})] = \sum_{x,y\in\Omega} \log\left[e^{-r^2(x,y)/\sigma^2} + e^{-c^2}\right]. \tag{2.49}$$

For model M_1 we assume a Gaussian distribution (with variance σ), where $r(x,y)$ is the residual error between the source and target defined as:

$$r(x,y) = [(m_7 f(x,y,t) + m_8) - (f(m_1 x + m_2 y + m_5, m_3 x + m_4 y + m_6, t-1))]^2. \quad (2.50)$$

For model M_2 we assume a uniform distribution (i.e., c is a constant). The log-likelihood function is maximized by differentiating, setting the result equal to zero and solving for $\vec{m}$:

$$\frac{d \log[L(\vec{m})]}{d\vec{m}} = \sum_{x,y \in \Omega} \frac{\frac{dr^2(x,y)}{d\vec{m}} e^{-r^2(x,y)/\sigma^2}}{e^{-r^2(x,y)/\sigma^2} + e^{-c^2}} = \sum_{x,y \in \Omega} \frac{dr^2(x,y)}{d\vec{m}} w(x,y) = 0, \quad (2.51)$$

where $w(\cdot)$ is defined to be the ratio of the exponential distributions. As in the previous sections, the residual $r(\cdot)$ is linearized with respect to the model parameters $\vec{m}$. The derivative of the residual, $dr^2(x,y)/d\vec{m}$, is then substituted into the above to yield:

$$\sum_{x,y \in \Omega} -2\vec{c}[k - \vec{c}^{\,T}\vec{m}]w \;\; = \;\; 0, \quad (2.52)$$

with $\vec{c}$ and k given by Equations (2.12) and (2.13), and, as before, all spatial parameters are dropped for notational convenience. Solving for the model parameters then yields the maximum likelihood estimator:

$$\vec{m} \;\; = \;\; \left[\sum_{x,y \in \Omega} (\vec{c}\vec{c}^{\,T}) w \right]^{-1} \left[\sum_{x,y \in \Omega} (\vec{c}k) w \right]. \quad (2.53)$$

Note that this solution is a weighted version of the earlier least-squares solution, Equation (2.9), where the weighting, w, is proportional to the likelihood that each pixel belongs to model M_1. As before (Section 2.2.3), a smoothness constraint can be imposed to yield the following iterative estimator:

$$\vec{m}^{(j+1)} \;\; = \;\; \left((\vec{c}\vec{c}^{\,T})w + L\right)^{-1} \left((\vec{c}k)w + L\overline{\vec{m}}^{(j)}\right). \quad (2.54)$$

This estimator for $\vec{m}$, however, requires an estimate of the weight w which itself requires an estimate of $\vec{m}$. The expectation/maximization algorithm (EM) is used to resolve this circular estimator, and proceeds as follows.

EM algorithm

1. Initialization: Assign the fixed constants σ^2, c^2. Assign initial estimates for $\vec{m}$.
2. E-Step: Compute the weights $w(x,y) = \frac{e^{-r(x,y)^2/\sigma^2}}{e^{-r(x,y)^2/\sigma^2}+e^{-c^2}}$.
3. M-Step: Use the weights $w(x,y)$ to estimate the solution for $\vec{m}$ as:

$$\vec{m}^{(j+1)} = \left((\vec{c}\,\vec{c}^{\,T})w + L\right)^{-1}\left((\vec{c}\,k)w + L\overline{\vec{m}}^{(j)}\right). \tag{2.55}$$

The last two steps are repeated until the difference between successive estimates of $\vec{m}$ is below a specified threshold.

The E-step is the segmentation stage, where pixels that do not have a corresponding match between source and target images have a close to zero weight w. These pixels are therefore given less consideration in the M-step which estimates the registration parameters $\vec{m}$. The EM algorithm allows for simultaneous segmentation and registration, and hence allows us to contend with missing or partial data.

2.3.4 Modification to the basic algorithm

From the previous section, we can see that the only modification necessary to handle partial data in the registration is to use a weighted least-squares solution instead of the original least-squares solution. Note that we have to carefully choose the initial parameters for σ^2, c^2, which are determined empirically. Recall from Section 2.2.4 that the original implementation is iterative in nature. The iterative EM algorithm can be integrated with the original iterative algorithm (Section 2.2.5) by simply updating

the weights between each Taylor-series iteration, and using these updated weights to estimate the registration parameters in the accuracy iterations.

2.3.5 Extension to 3-D

The generalization of the algorithm from 2-D images to 3-D volumes is relatively straight-forward. Briefly, to accommodate a 3-D affine transformation, an additional six affine parameters are added to the geometric and intensity transformation model of Equation (2.10). Linearization and minimization of this constraint proceeds as in the 2-D case. The smoothness constraint of Equation (2.16) takes on an additional $(\partial m_i/\partial z)^2$ term, and the iterative estimator of Equation (2.23) is of the same form, with k and $\vec{c}$ accommodating a different set of, now 3-D, spatial/temporal derivatives. The solution of Section (2.3.3) proceeds in a similar manner, with the initial constraint of Equation (2.46) updated to accommodate the 3-D geometric and intensity transformation model.

2.4 Summary

In this chapter, we have explained in detail our registration algorithm, using a differential framework. Our registration model incorporates both a geometric mapping that is locally affine and globally smooth, and contrast/brightness modulations that are globally smooth. This flexible model makes it practical to register images in many real-world applications. We have also shown how the basic registration algorithm can be extended to handle missing data using an EM framework, further extending the usefulness of the technique. This extension is simple and easily integrated within our original framework. In the next section, we demonstrate the efficacy of our algorithm on a broad range of both synthetic and clinical imagery.

Chapter 3

Results

3.1 Introduction

The previous section described the theory and implementation details of our registration algorithm. In this section examples of 2-D and 3-D registration results are presented for a broad range of medical images, demonstrating the efficacy of the algorithm. Synthetic images with known geometric and intensity maps are used to quantify the results. In addition to these synthetic registration examples, many clinical registration cases are also presented. Though the registration algorithm has several tunable parameters, they are all help fixed across all experiments.

3.2 2-D registration examples

In all of the 2-D examples shown here (Figures 3.2-3.27), the source and target are 256×256, 8-bit grayscale images with intensity values scaled into the range $[0, 1]$. In order to contend with border effects, each image is padded with zeros to a size of 288×288. A four-level Gaussian pyramid is constructed for both the source and

target image. Each level of the pyramid is obtained by convolving the previous level using a separable 5-tap lowpass filter [0.05 0.25 0.4 0.25 0.05], followed by down-sampling by a factor of two. The finest scale consists of the original image. At each pyramid level a single global affine map is first estimated as in Section 2.2.2, with Ω, the spatial integration window, defined to be the entire image. The local affine and contrast/brightness parameters $\vec{m}$ are then estimated as in Section 2.2.2, with $\Omega = 5 \times 5$ pixels. This estimate of $\vec{m}$ is used to bootstrap the smoothness iterations, Equation (2.23). In each iteration, $\lambda_i = 1 \times 10^{11}$, $i = 1, ..., 8$ and $\overline{m_i}$ is computed by convolving with the 3×3 kernel $[\ 1 \quad 4 \quad 1 \ ; \ 4 \quad 0 \quad 4 \ ; \ 1 \quad 4 \quad 1]\ /20$. After forty iterations (inner loop, smoothness iterations), the source is warped according to the final estimate, and this process is repeated five times (outer loop, Taylor-series iterations). The final estimate is then up-sampled, and used as an initial estimate in the next finer level of the pyramid (by initially warping the source image at the new pyramid level according to this estimate). This entire process is repeated at each level of the pyramid. Although a contrast/brightness map is estimated, it is not applied when warping the source image. In order to minimize artifacts during warping (due to interpolation), we accumulate successive maps and apply a single map to the original source image at each scale. In order to minimize edge artifacts, all convolutions are performed with a mirror-symmetric boundary. Most of the parameters described above were empirically chosen and are generally consistent with commonly used parameters in the motion estimation literature - they were held fixed in all of the examples shown here. In general we find that the particular choice of these parameters is not crucial. The algorithm has been implemented using MatLab, and requires approximately 4 minutes per 256×256 image on a 2.8 GHz Linux machine.

3.2.1 Synthetic experiments

To test our registration algorithm, we generated synthetic data by applying a locally smooth geometric and contrast/brightness map to a target image. These smooth maps were generated by specifying a random displacement parameter at equally spaced points along a coarse rectilinear grid. On average each pixel was transformed by ± 8 pixels (not including a possible global affine map), the multiplicative contrast variation was between 0.8 and 1.0, and the additive brightness variation between 0.0 and 0.2 (with image intensities in $[0, 1]$).

Shown in Figures 3.2-3.7 are a set of sample results. Shown in panels (a) and (b) is a synthetically generated source and target image, respectively. Shown in panel (c) is the source image after registration. Shown in panels (d) and (e) are the error images (showing the edges only) before and after registration respectively. Shown in panels (f) and (g) are the applied synthetic contrast and brightness maps, and shown in panels (i) and (j) are the estimated maps. Note that while there is a trade-off between the estimated contrast and brightness maps, errors in this estimate do not impact the estimated geometric map. Shown in panel (h) is the synthetic registration map as applied to a rectilinear grid. Shown in panel (k) is the result of applying the inverse of the estimated map to panel (h). If the estimate was perfect, the result should be a rectilinear grid. Notice that in the areas of image content, this is nearly the case. Shown in Figure 3.1 are the results from an extreme and completely unrealistic synthetic map, which we show to illustrate the robustness and flexibility of our registration technique. Unlike the previous examples, the model used here is one of translation only (i.e., no affine or contrast/brightness terms), and the smoothness parameters on these translation terms were reduced to 1×10^{-2}. These small changes were necessary to accommodate the extreme nature of the synthetic

figure	description	intensity		geometry	
		pre	post	mean	median
3.2	Mammograms	0.25	0.02	0.88	0.18
3.3	Chest x-ray	0.39	0.05	3.12	0.28
3.4	MRI axial	0.15	0.01	0.92	0.06
3.5	MRI sagital	0.13	0.01	0.43	0.10
3.6	MRI axial T1/T2	0.16	0.01	0.10	0.07
3.7	MRI tumor	0.06	0.01	0.12	0.02

Table 3.1: Results of 2-D synthetic experiments showing the intensity errors before and after registration, and the error in the estimated geometric map. Shown in the intensity column are the mean square error in intensities before and after the registration. The source and target intensities are orinally in the range $[0, 1]$. Shown in the geometry column are the mean and median (in pixels) of the error between the estimated registration map and the applied registration map.

map. Table 3.1 quantifies these results, in both intensity- and geometry-space. The error in intensity is reported as the mean square error (MSE) in pixel intensities before and after registration, with the contrast and brightness maps applied. The error in geometry is obtained by computing the error between the estimated registration map and the applied registration map. Recall that the registration map is represented as a displacement vector at each pixel location. The error in registration maps at each pixel location is computed as the euclidean distance, or L^2-norm, given by $\|v_1 - v_2\|$.

3.2.2 Clinical experiments

In the previous section results using a synthetic map were presented. The random geometric and intensity distortions were chosen to represent a large class of possible practical distortions. In this section, the registration algorithm is run on a number of real clinical images. Note that in this case, the actual registration map is unknown, hence the results cannot be verified directly. Visual comparison is used to judge the results. The parameters used in these examples are all kept fixed, and are the same

as in the previous section.

Shown in Figures 3.8-3.27 are a set of sample results, for pairs of source and target images, (a) and (b). Shown in panel (c) is the result of registration. Shown in panels (d) and (e) are the error images (showing the edges only) before and after registration respectively. Shown in panels (f) and (g) are the estimated contrast/brightness maps, and shown in panel (h) is the estimated map. These results demonstrate the efficacy of the algorithm over a broad range of imaging hardware (e.g., MRI, X-rays, photos, ultrasound). Even in the presence of significant intensity variations, the registered source is in good agreement with the target image. Table 3.2 quantifies these results in intensity space. The error in intensity is reported as the mean square error (MSE) in pixel intensities before and after registration, with the contrast and brightness maps applied.

3.3 3-D registration examples

In all of the 3-D examples shown here (Figures 3.28-3.35), the source and target are $64 \times 64 \times 64$, 16-bit grayscale volumes with intensity values scaled into the range $[0, 1]$. These experiments closely follow the 2-D experiments, with the following parameters: a two-level Gaussian pyramid, ten inner loop iterations, four outer loop iterations, $\Omega = 5 \times 5 \times 5$ and $\lambda_i = 1 \times 10^{11}$, $i = 1, ..., 14$. The current MatLab implementation requires approximately 30 minutes on a 2.8 GHz Linux machine with 1 GB of memory. Figures 3.28-3.35 show 3-D registration results for both synthetic and clinical cases. Iso-surface views for all these volumes are presented in Figures 3.36 and 3.37.

figure	description	MSE pre-registration	MSE post-registration
3.8	MRI sagitals, different subjects	0.17	0.05
3.9	MRI axials, different subjects	0.20	0.09
3.10	MRI axials T1/T2, same subject (1)	0.12	0.06
3.11	MRI axials T1/T2, same subject (2)	0.12	0.06
3.12	MRI axials T1/T2, same subject (3)	0.14	0.06
3.13	MRI axials T1/T2, same subject (4)	0.22	0.08
3.14	MRI axials T1/T2, same subject (5)	0.09	0.04
3.15	Mammograms	0.14	0.03
3.16	Chest X-rays	0.17	0.04
3.17	Pelvic tumor	0.09	0.04
3.18	Lung tumor	0.14	0.07
3.19	Ultrasound	0.11	0.03
3.20	Photo during brain operation (1)	0.09	0.08
3.21	Photo during brain operation (2)	0.27	0.09
3.22	MRI sagitals pre/post operation	0.26	0.08
3.23	EPI experiment	0.03	0.02
3.24	Photon density/MRI T1	0.09	0.02
3.25	CT/photo, axial (1)	0.19	0.08
3.26	CT/photo, axial (2)	0.27	0.10
3.27	CT/photo, axial (3)	0.14	0.07

Table 3.2: Results of 2-D clinical experiments showing the MSE in intensity before and after registration (with the contrast and brightness maps applied).

3.3.1 Synthetic experiments

Shown in Figure 3.28 are results using a synthetically warped volume with only local geometric distortions along with an intensity distortion. Shown in the first and second columns are source and target volumes, respectively. Shown in the third column is the registered source. Each row corresponds to a specific z-slice in the volume. Shown along the top row are results from a synthetically generated grid volume, which illustrate the extent of the synthetic map and the results of registration.

Shown in Figure 3.29 are results using a synthetically warped volume with intensity and both local and global geometric distortions. These results demonstrate the

figure	description	MSE pre-registration	MSE post-registration
3.28	local distortions	0.05	0.01
3.29	local and global distortions	0.12	0.01

Table 3.3: Results of 3-D synthetic experiments showing the MSE in intensity before and after registration.

figure	description	MSE pre-registration	MSE post-registration
3.30	MR brain volume (1)	0.10	0.04
3.31	MR brain volume (2)	0.09	0.03
3.32	MR brain volume (3)	0.11	0.05
3.33	MR brain volume (4)	0.11	0.04
3.34	MR brain volume (5)	0.10	0.04
3.35	MR brain volume (6)	0.10	0.05

Table 3.4: Results of 3-D clinical experiments showing the MSE in intensity before and after registration.

ability of our technique to register volumes with both large and small overall differences. Shown in Table 3.3 are the MSE in intensities before and after the registration for these results.

3.3.2 Clinical experiments

Shown in Figures 3.30-3.35 are clinical examples of registration between different subjects. Shown in the first and second columns are the source and target volumes, respectively. Shown in the third column is the registered source. These results demonstrate the ability of our technique to register clinical images, even in the presence of significant within-subject differences. Shown in Table 3.4 are the MSE in intensities before and after the registration for these results.

3.4 Partial data registration examples

Section 2.3 describes in detail an extension of the registration algorithm that can handle images with missing data using Expectation Maximization. In this section, registration results are presented for such images and volumes. As before, all parameters are kept constant (same as in Section 3.2 for 2-D and Section 3.3 for 3-D) for all the results presented. In addition to the regular registration parameters, two additional parameters are used in the extension: σ^2 and b^2. For all the examples shown, $\sigma^2 = 0.01$ and $b = e^{0.001}$. In this case, the algorithm is sensitive to these two constants. There is no guarantee that these same constants would work for other examples.

3.4.1 2-D synthetic and clinical experiments

In this set of synthetic results (Figures 3.38-3.42), images with known distortions and missing data are used. The ability of the algorithm to recover these known distortions, even in the case of severe occlusions, is demonstrated. The geometric model used in these examples is global affine, in order to better illustrate the ability of the algorithm to handle these wide range of distortions. Shown in Table 3.5 are the MSE in intensities before and after registration. Table 3.6 show the affine parameters before and after registration. In the next set of clinical results (Figures 3.43- 3.54), clinical data containing missing data are registered, and the results presented. In this case, the distortions are not known in advance and cannot be quantitatively verified. Figures 3.43 and 3.44 use a global affine geometric model. Figures 3.45-3.54 contain both global and local geometric distortions. Figures 3.45-3.50 are results with missing data, while Figures 3.51-3.54 show the result of applying the algorithm to images without missing data (these images have also been registered previously in

figure	description	MSE pre-registration	MSE post-registration
3.38	sagital images with missing data + noise	0.26	0.18
3.39	partial sagital images	0.14	0.15
3.40	sagitals, white+black squares	0.30	0.20
3.41	sagitals, white squares	0.41	0.40
3.42	segmented brain with MR	0.27	0.25

Table 3.5: Results of 2-D synthetic experiments with partial data, with global distortions, showing the MSE in intensity before and after registration.

Section 3.2.2). This demonstrates that the algorithm remains effective in registration even when there are no occlusions. It can be seen that the registration algorithm can recover clinical distortions even in the presence of missing data. In particular, the results for Figure 3.45, with more than half the brain missing, highlight clearly the effectiveness of the registration algorithm. Shown in Table 3.7 are the MSE in intensities before and after registration.

3.4.2 3-D synthetic and clinical experiments

In Figures 3.55 and 3.56, a portion of the brain volume has been removed, and a global distortion applied (synthetic experiments). Figures 3.57 and 3.58 demonstrate the ability of the algorithm to recover unknown distortions in clinical images. In Figure 3.57, a head volume that has been segmented and the brain extracted is registered with another complete head volume from a different subject. Figure 3.58 shows the result before and after an operation on a pig within an MRI coil. As demonstrated by these results, the algorithm can effectively register in synthetic and clinical settings with partial data. Table 3.8 shows the MSE in intensities before and after the registration. Iso-surface views for all these volumes are presented in Figure 3.59.

figure	affine parameters pre-registration	affine parameters post-registration
3.38	$\begin{bmatrix} 1.28 & -0.23 \\ 0.23 & 1.28 \end{bmatrix} \begin{bmatrix} 1.91 \\ -2.20 \end{bmatrix}$	$\begin{bmatrix} 1.28 & -0.23 \\ 0.23 & 1.28 \end{bmatrix} \begin{bmatrix} 1.87 \\ -2.27 \end{bmatrix}$
3.39	$\begin{bmatrix} 1.28 & -0.23 \\ 0.23 & 1.28 \end{bmatrix} \begin{bmatrix} 1.91 \\ -2.20 \end{bmatrix}$	$\begin{bmatrix} 1.28 & -0.23 \\ 0.23 & 1.28 \end{bmatrix} \begin{bmatrix} 1.82 \\ -2.28 \end{bmatrix}$
3.40	$\begin{bmatrix} 1.28 & -0.23 \\ 0.23 & 1.28 \end{bmatrix} \begin{bmatrix} 1.91 \\ -2.20 \end{bmatrix}$	$\begin{bmatrix} 1.28 & -0.23 \\ 0.23 & 1.28 \end{bmatrix} \begin{bmatrix} 1.82 \\ -2.27 \end{bmatrix}$
3.41	$\begin{bmatrix} 0.99 & -0.17 \\ 0.17 & 0.99 \end{bmatrix} \begin{bmatrix} 0.00 \\ 0.00 \end{bmatrix}$	$\begin{bmatrix} 1.00 & -0.17 \\ 0.13 & 0.97 \end{bmatrix} \begin{bmatrix} -1.00 \\ 3.30 \end{bmatrix}$
3.42	$\begin{bmatrix} 0.82 & -0.19 \\ -0.43 & 1.18 \end{bmatrix} \begin{bmatrix} 2.08 \\ -5.00 \end{bmatrix}$	$\begin{bmatrix} 0.82 & -0.19 \\ -0.43 & 1.18 \end{bmatrix} \begin{bmatrix} 2.02 \\ -5.02 \end{bmatrix}$

Table 3.6: Results of 2-D synthetic experiments with partial data, with global distortions, showing affine parameters before and after registration. The six parameters are represented in matrix form as follows: $\begin{bmatrix} m_1 & m_2 \\ m_3 & m_4 \end{bmatrix} \begin{bmatrix} m_5 \\ m_6 \end{bmatrix}$, where $m_1 ... m_4$ are the affine parameters and m_5, m_6 are the translation parameters.

3.5 Registration failures

In all of the above examples, only successful registration examples are shown. In this section, we present examples where the registration is unsuccessful. These failures are usually caused by a violation of one or more of the design assumptions (see Section 4 for a detailed analysis). Some common violations are as follows. In the first example, Figure 3.60, the target image consists of the source image rotated by 90 degrees. In this case, the distortion is too large and cannot be detected even in the coarsest scale, causing the algorithm to fail. In the next example, Figure 3.61, a PET image is registered to an MR image. In this case, the likely cause of failure is the large non-smooth intensity distortions. In panel (d) of this figure, the estimated brightness and

figure	description	MSE pre-registration	MSE post-registration
3.43	Segmented brain, different subjects, global	0.28	0.26
3.44	MR/MR coil, global	0.20	0.19
3.45	Segmented brain, different subjects, sagital	0.26	0.24
3.46	Segmented brain, different subjects, axial	0.29	0.22
3.47	MRI coil, pig experiment	0.04	0.03
3.48	MRI sagitals, with missing data	0.16	0.10
3.49	MRI sagitals, with noise	0.20	0.14
3.50	Chest X-rays, with noise	0.24	0.19
3.51	MRI sagitals, different subjects	0.17	0.09
3.52	MRI axials, different subjects	0.20	0.13
3.53	Mammograms	0.14	0.05
3.54	Chest X-rays	0.17	0.26

Table 3.7: Results of 2-D clinical experiments with partial data, showing the MSE in intensity before and after registration.

figure	description	MSE pre-registration	MSE post-registration
3.55	MR partial brain (1)	0.16	0.13
3.56	MR partial brain (2)	0.14	0.08
3.57	MRI stripped brain	0.15	0.12
3.58	MRI coil pig experiment	0.05	0.03

Table 3.8: Results of 3-D experiments with partial data, showing the MSE in intensity before and after registration.

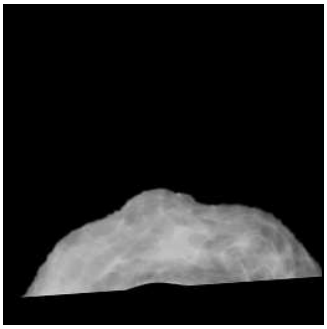
contrast maps (shown in panels (g) and (h) respectively) are applied to the registered source from panel (c); note that this image appears similar to the target. This shows that the registration algorithm has chosen to explain most of the distortion with the brightness/contrast maps. In the final example, Figure 3.62, the source and target images consists of two photographs of the same brain, taken at different times during an operation. It can be seen that in the source image, a portion of the image in the top left half has been covered up, but exposed in the target image. In the region of the occlusion, there are no corresponding points, causing the algorithm to fail (note that in all these examples, the registration algorithm is run without the extension

Figure 3.1: Shown is a target image, a source image exposed to an extreme transform, and the result of our registration.

for missing data, to illustrate specific failures). These three examples illustrate only a few of the many possible cases where a design assumption is violated. However, note that the design assumptions are reasonable and cover many practical cases, as illustrated by the large number of clinical registration examples.

3.6 Summary

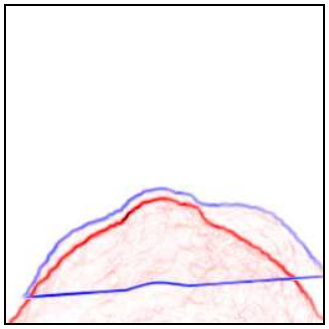
In this section, a large number of registration examples were presented for 2-D images and 3-D volumes with known and unknown distortions, with and without missing data, demonstrating the efficacy of the algorithm. As can be seen, the algorithm remains relatively robust in the presence of various types of intensity distortions as well as occlusions. The algorithm is able to successfully capture a broad range of distortions over a broad range of imagery. The carefully chosen design assumptions, as outlined in Section 1.2, along with a careful implementation, has proven to work well. In the next section, the extent of distortions (both in geometry and intensity) that can be estimated is analyzed, highlighting and quantifying the capabilities and limits of the registration algorithm.



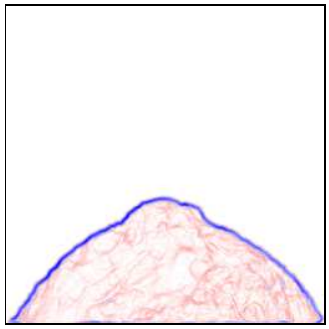
(a) source

(b) target

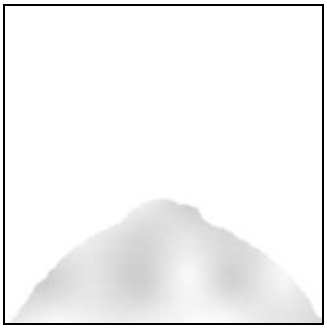
(c) registered source



(d) error before registration

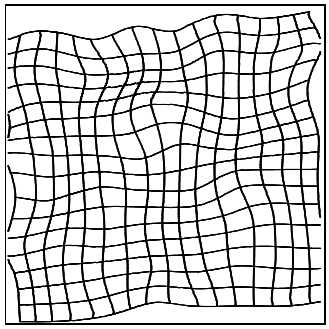


(e) error after registration

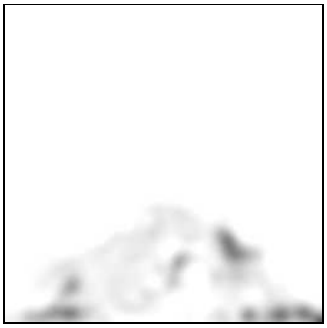


(f) applied contrast

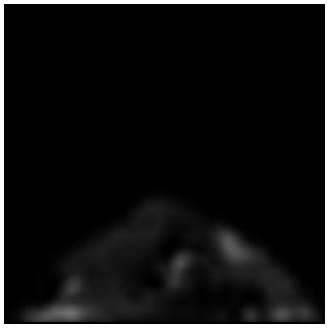
(g) applied brightness



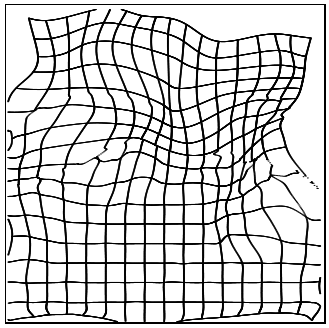
(h) applied geometric map



(i) estimated contrast



(j) estimated brightness

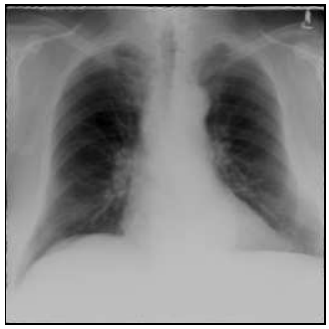


(k) rectified geometric map

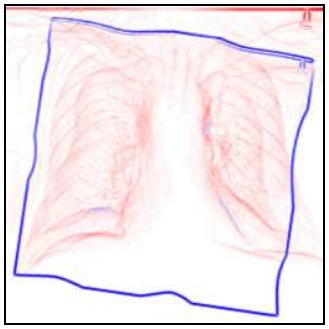
Figure 3.2: Synthetic result: Mammograms with random warp.

(a) source

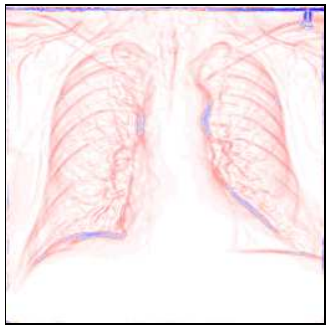
(b) target



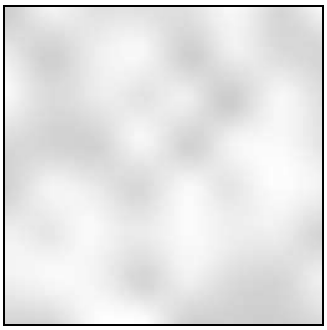
(c) registered source



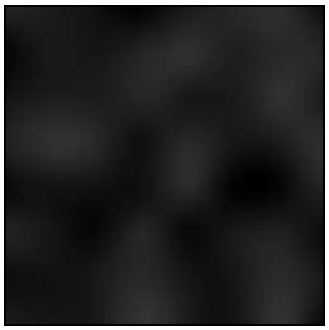
(d) error before registration



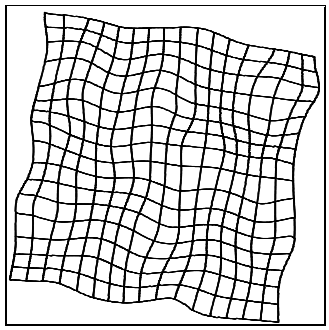
(e) error after registration



(f) applied contrast



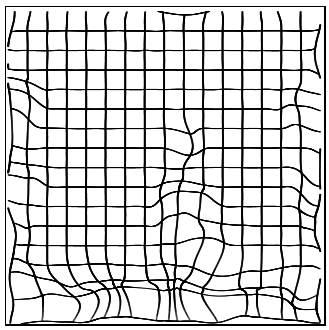
(g) applied brightness



(h) applied geometric map

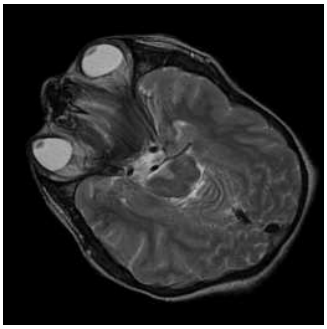
(i) estimated contrast

(j) estimated brightness

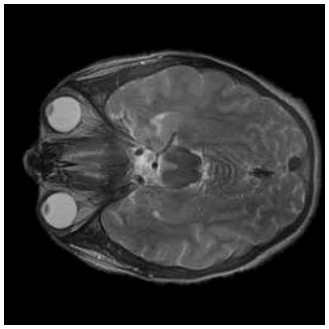


(k) rectified geometric map

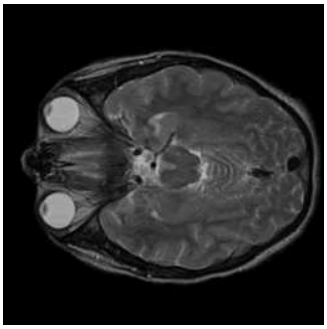
Figure 3.3: Synthetic result: Chest X-ray with random warp.



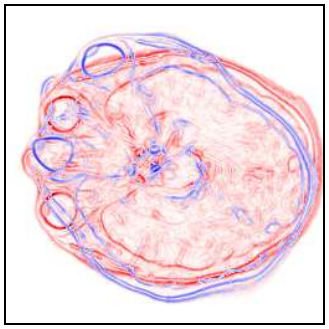
(a) source



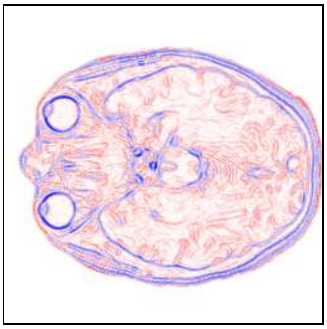
(b) target



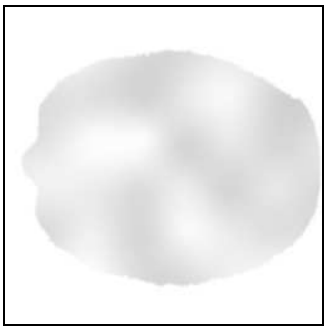
(c) registered source



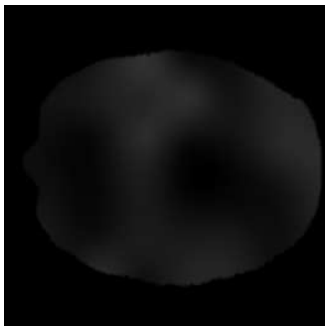
(d) error before registration



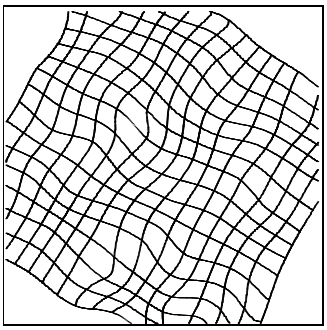
(e) error after registration



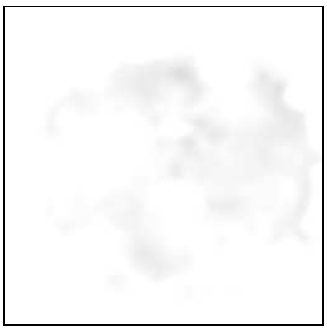
(f) applied contrast



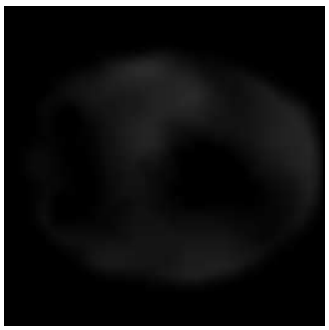
(g) applied brightness



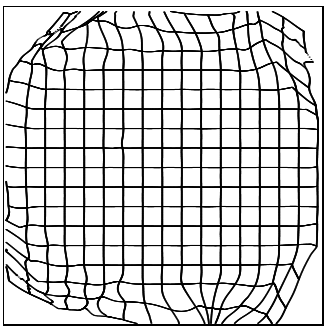
(h) applied geometric map



(i) estimated contrast

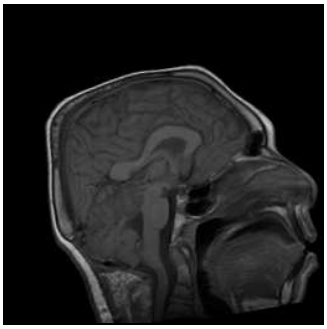


(j) estimated brightness

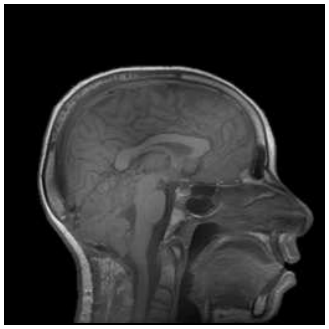


(k) rectified geometric map

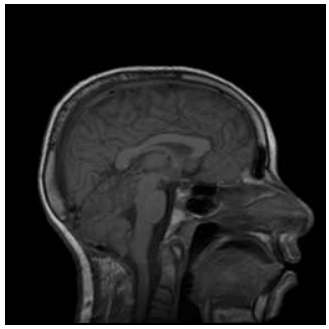
Figure 3.4: Synthetic result: MRI axials with random warp.



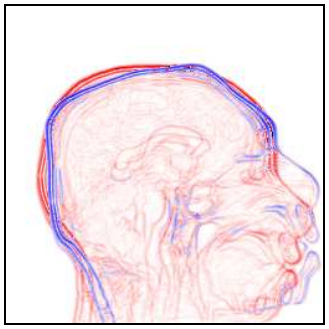
(a) source



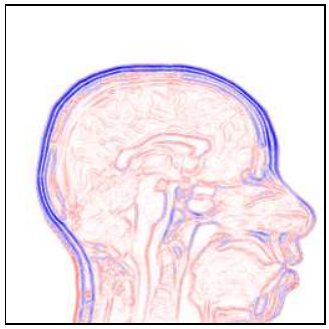
(b) target



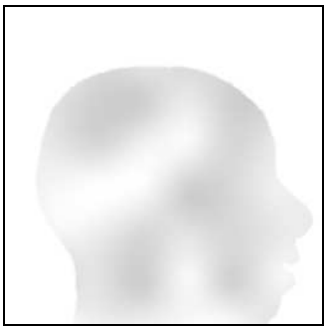
(c) registered source



(d) error before registration

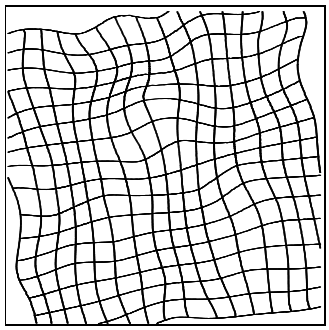


(e) error after registration

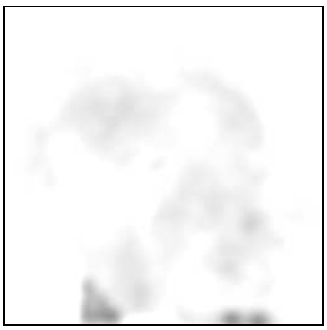


(f) applied contrast

(g) applied brightness

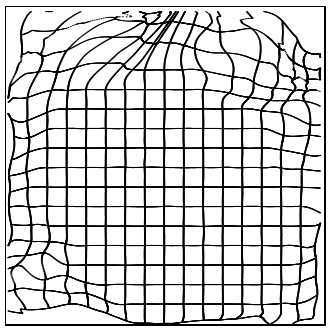


(h) applied geometric map



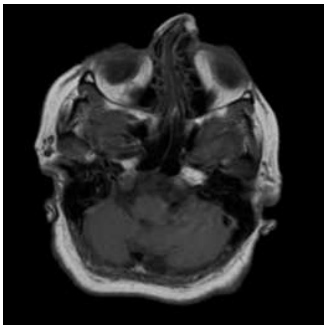
(i) estimated contrast

(j) estimated brightness

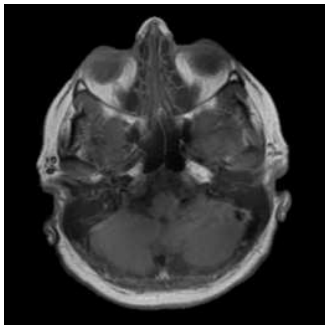


(k) rectified geometric map

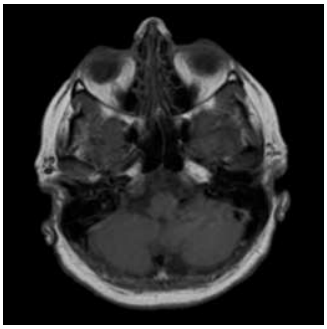
Figure 3.5: Synthetic result: MRI sagitals with random warp.



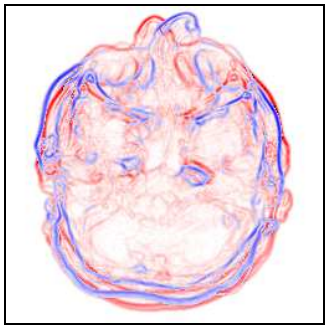
(a) source



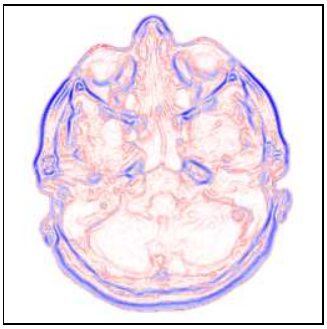
(b) target



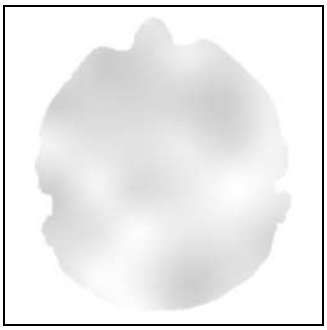
(c) registered source



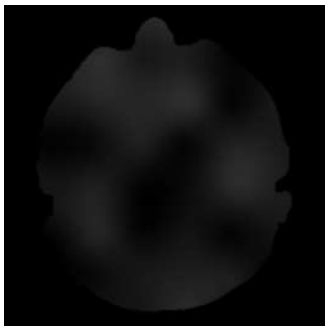
(d) error before registration



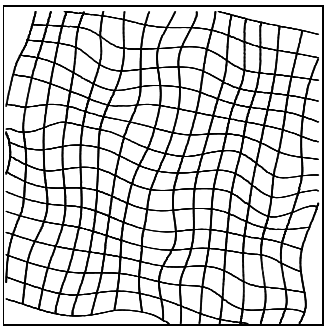
(e) error after registration



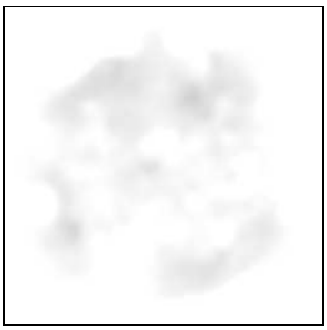
(f) applied contrast



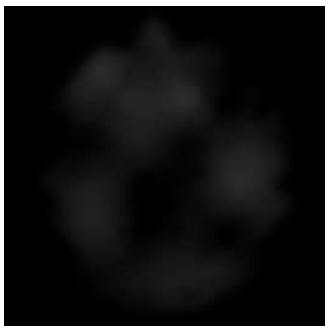
(g) applied brightness



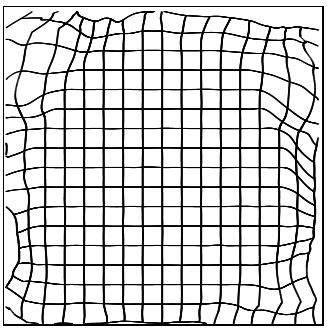
(h) applied geometric map



(i) estimated contrast

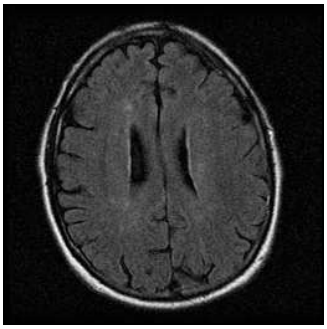


(j) estimated brightness

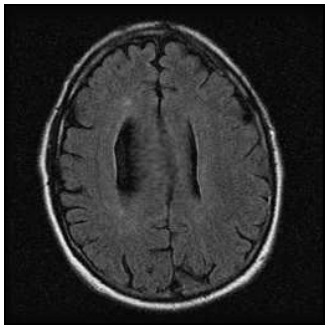


(k) rectified geometric map

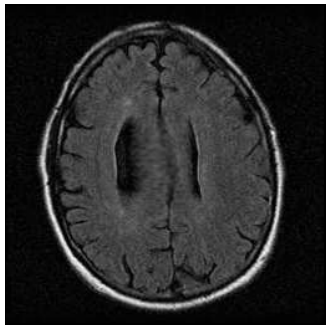
Figure 3.6: Synthetic result: MRI T1/T2 axials with random warp.



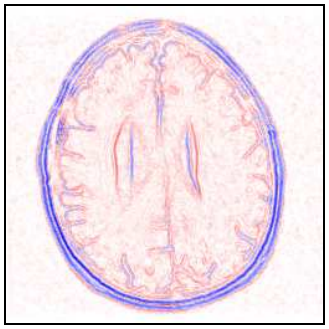
(a) source



(b) target

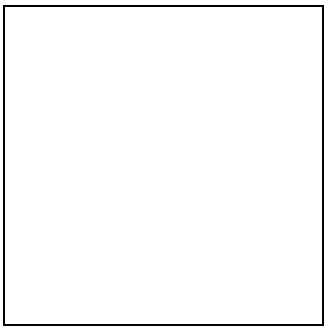


(c) registered source

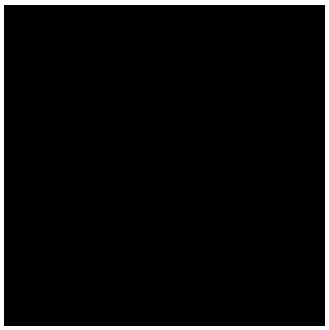


(d) error before registration

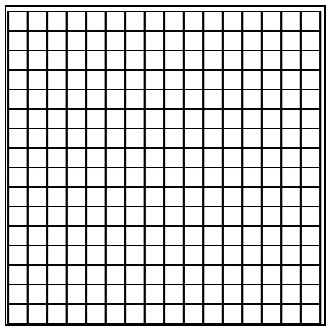
(e) error after registration



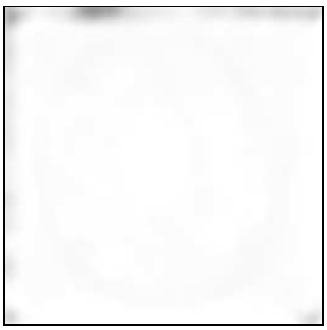
(f) applied contrast



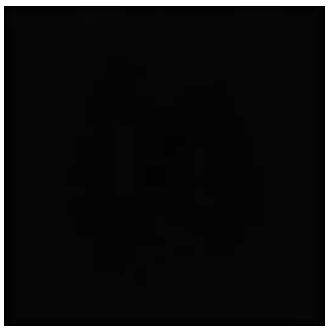
(g) applied brightness



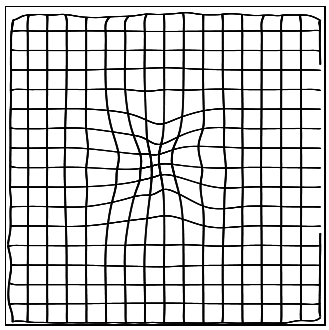
(h) applied geometric map



(i) estimated contrast

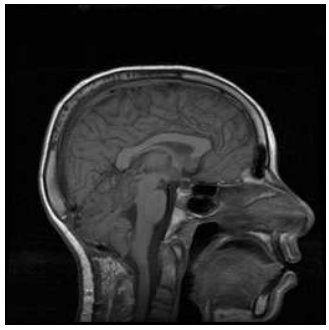


(j) estimated brightness

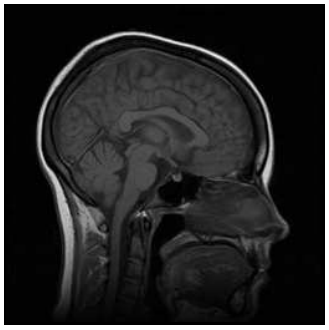


(k) rectified geometric map

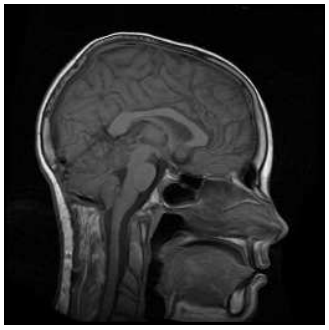
Figure 3.7: Synthetic result: MRI tumor.



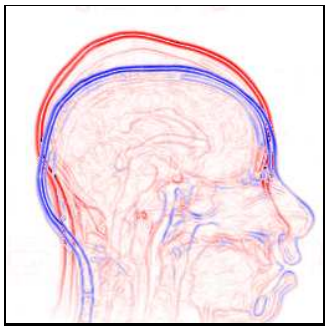
(a) source



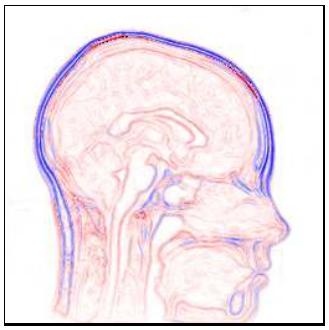
(b) target



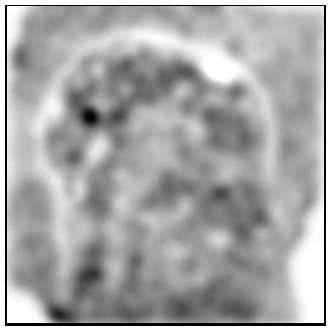
(c) registered source



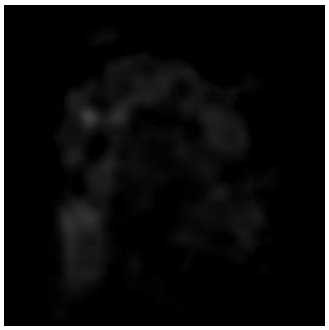
(d) error before registration



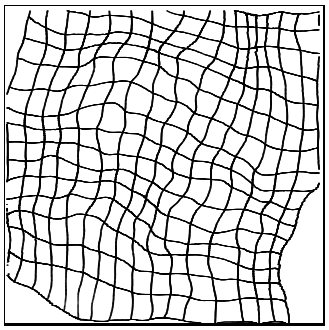
(e) error after registration



(f) estimated contrast

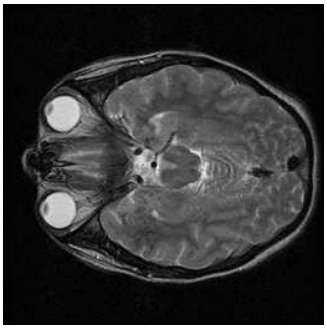


(g) estimated brightness

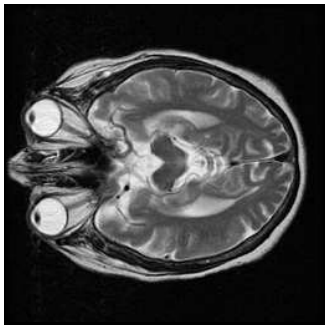


(h) estimated geometric map

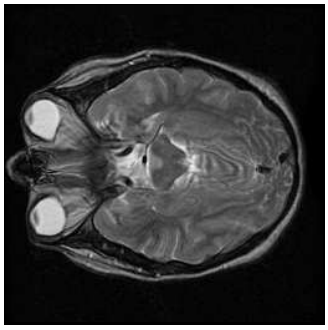
Figure 3.8: Clinical result: MRI sagitals, different subjects.



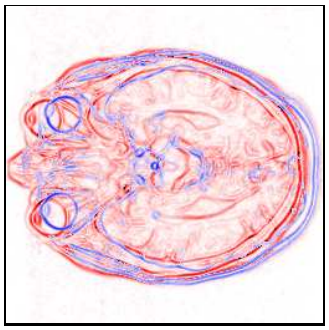
(a) source



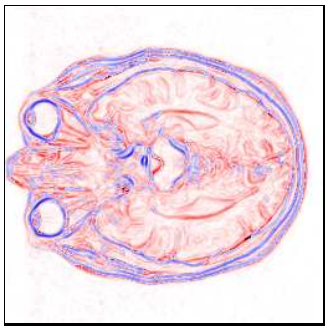
(b) target



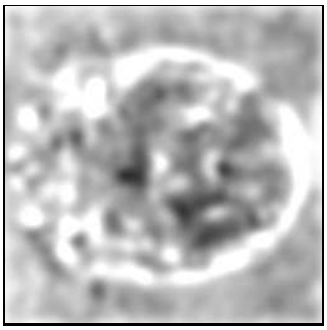
(c) registered source



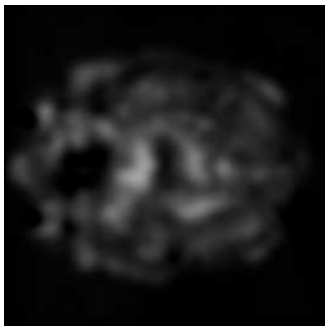
(d) error before registration



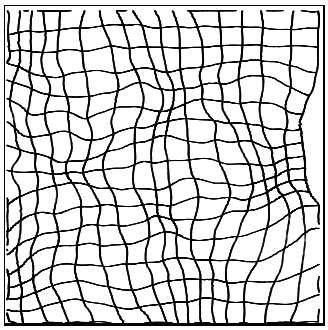
(e) error after registration



(f) estimated contrast

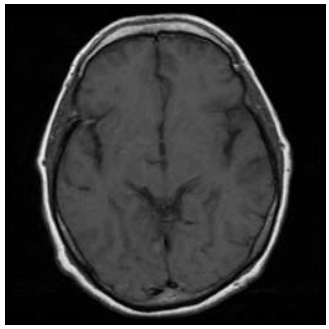


(g) estimated brightness

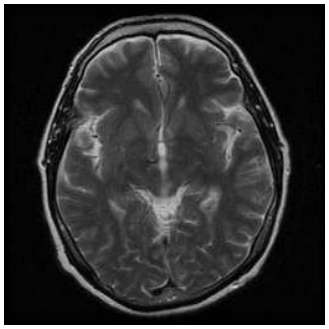


(h) estimated geometric map

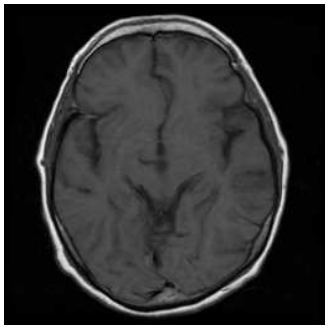
Figure 3.9: Clinical result: MRI axials, different subjects.



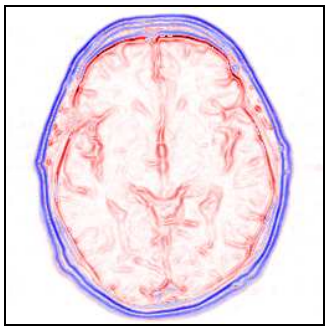
(a) source



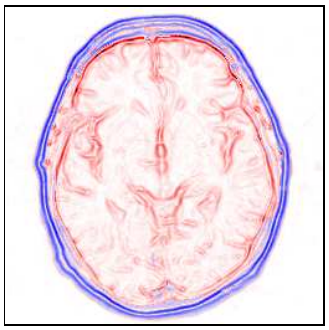
(b) target



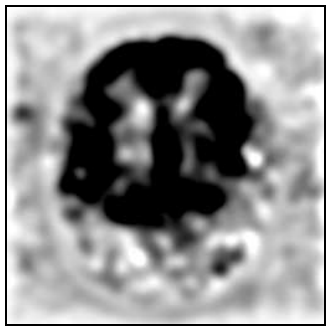
(c) registered source



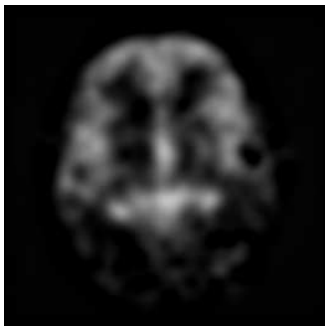
(d) error before registration



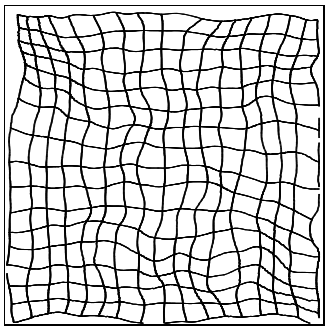
(e) error after registration



(f) estimated contrast

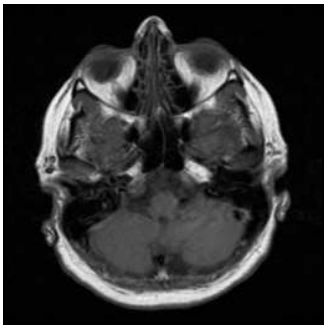


(g) estimated brightness

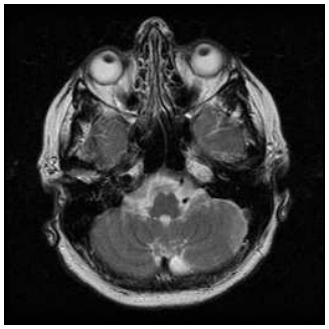


(h) estimated geometric map

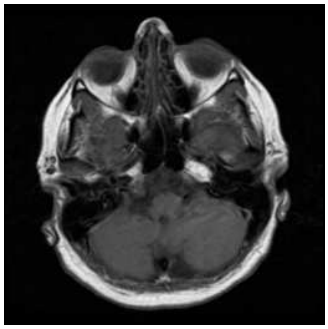
Figure 3.10: Clinical result: MRI axials T1/T2, same subject (1).



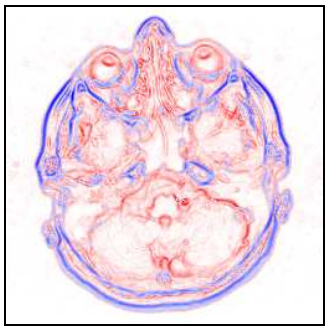
(a) source



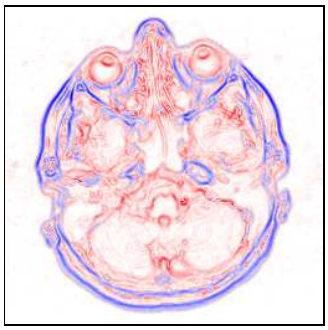
(b) target



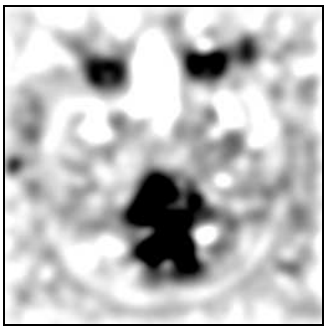
(c) registered source



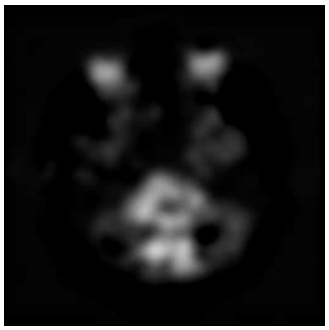
(d) error before registration



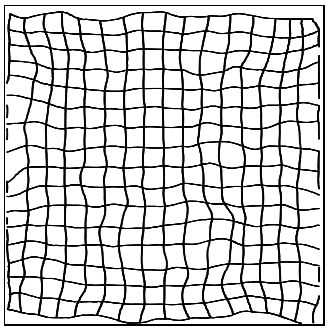
(e) error after registration



(f) estimated contrast

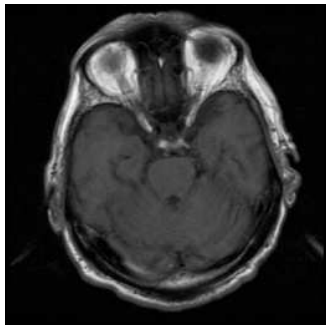


(g) estimated brightness

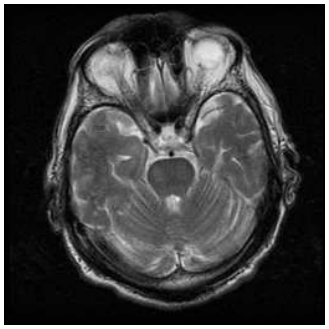


(h) estimated geometric map

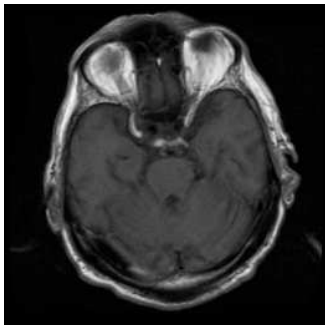
Figure 3.11: Clinical result: MRI axials T1/T2, same subject (2).



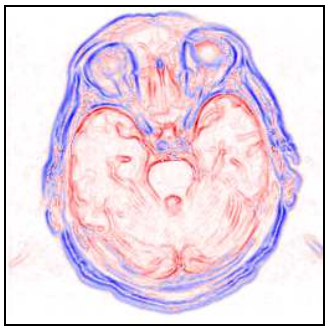
(a) source



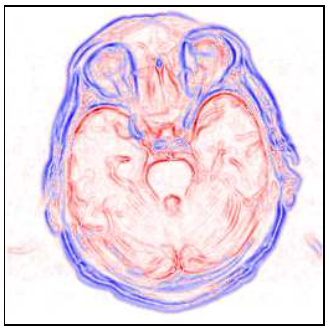
(b) target



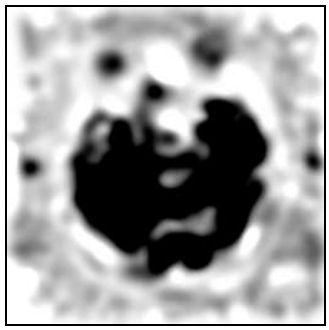
(c) registered source



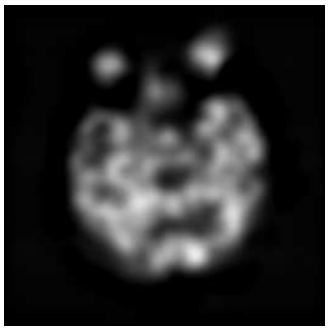
(d) error before registration



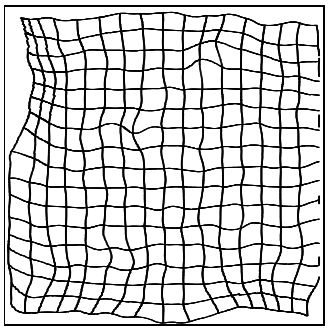
(e) error after registration



(f) estimated contrast

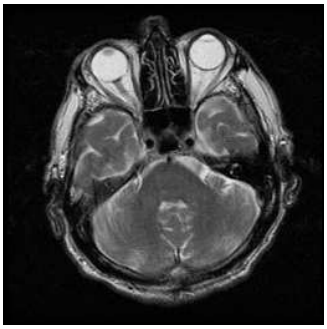


(g) estimated brightness

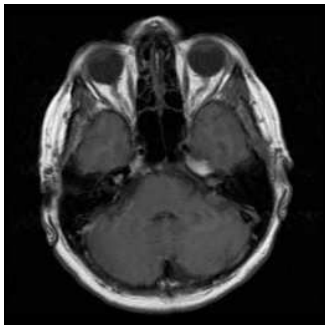


(h) estimated geometric map

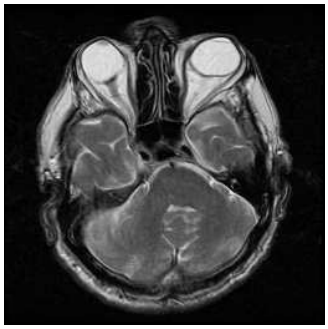
Figure 3.12: Clinical result: MRI axials T1/T2, same subject (3).



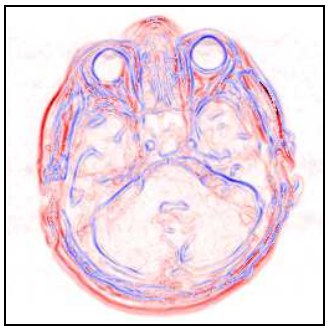
(a) source



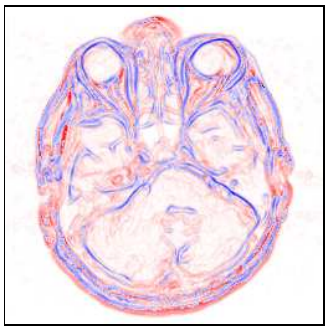
(b) target



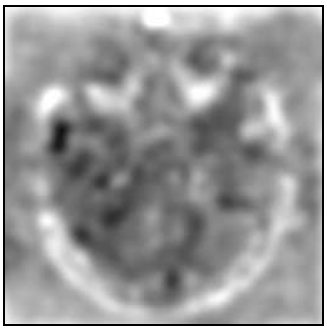
(c) registered source



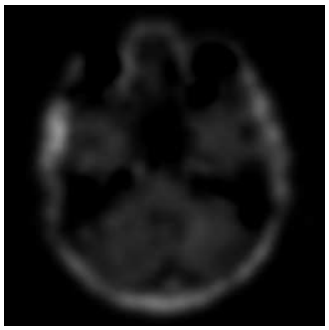
(d) error before registration



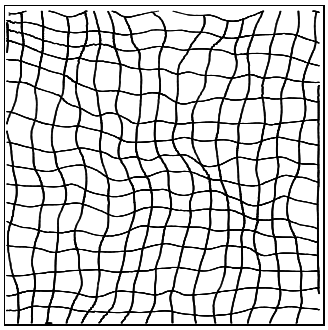
(e) error after registration



(f) estimated contrast

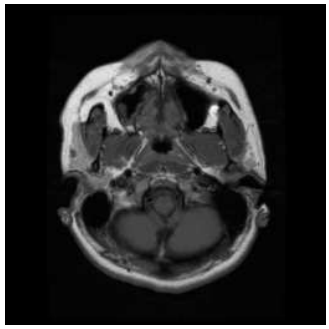


(g) estimated brightness

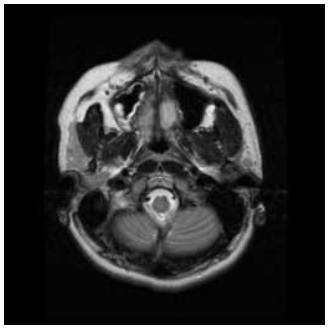


(h) estimated geometric map

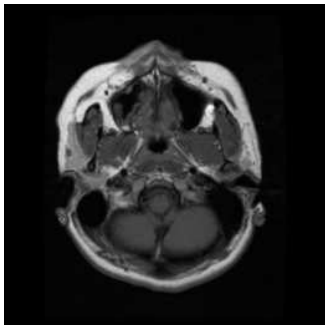
Figure 3.13: Clinical result: MRI axials T1/T2, same subject (4).



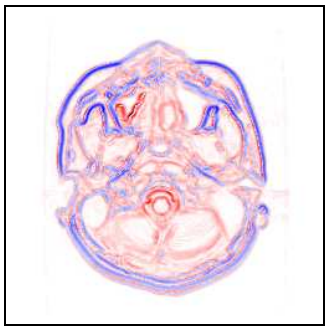
(a) source



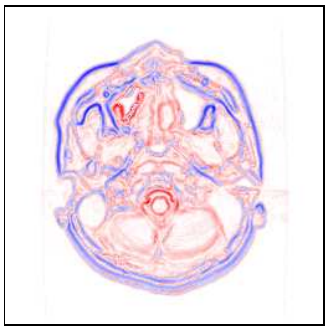
(b) target



(c) registered source

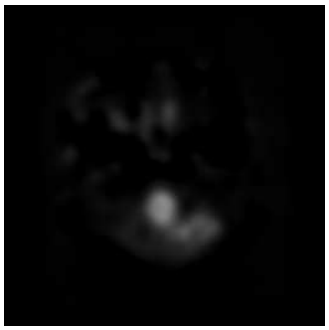


(d) error before registration

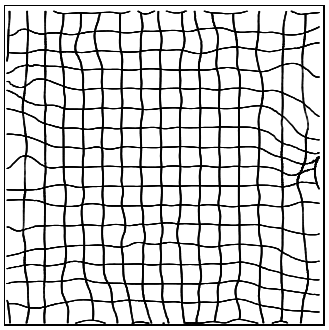


(e) error after registration

(f) estimated contrast



(g) estimated brightness

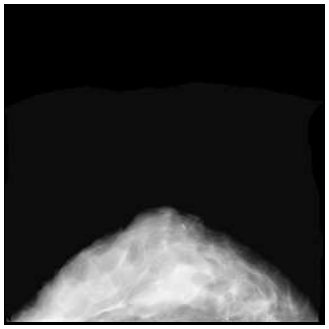


(h) estimated geometric map

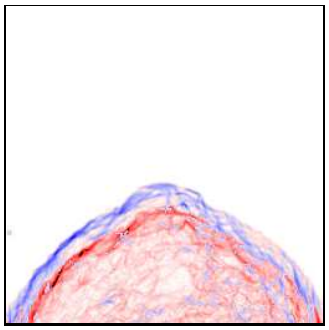
Figure 3.14: Clinical result: MRI axials, T1/T2, same subject (5).

(a) source

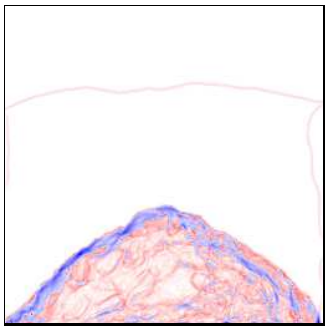
(b) target



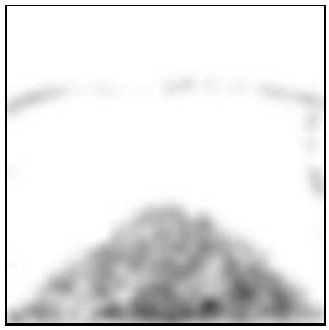
(c) registered source



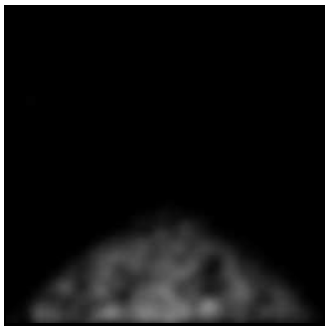
(d) error before registration



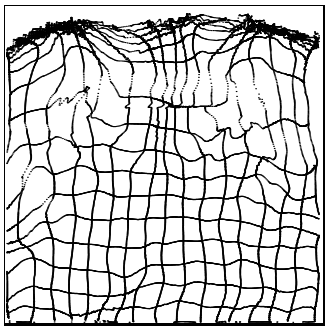
(e) error after registration



(f) estimated contrast



(g) estimated brightness



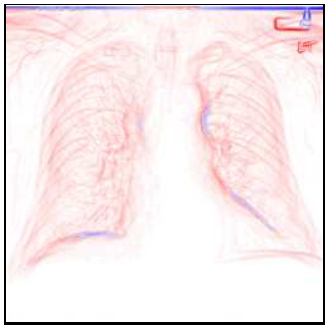
(h) estimated geometric map

Figure 3.15: Clinical result: Mammograms.

(a) source

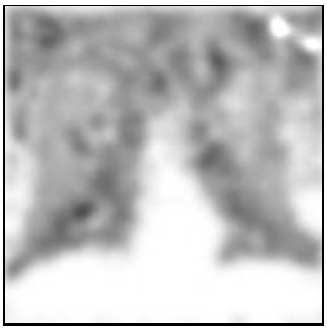
(b) target

(c) registered source

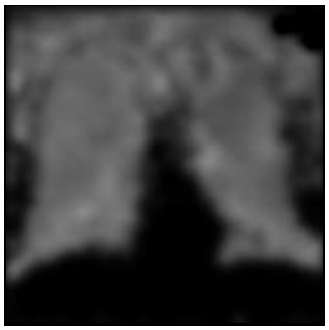


(d) error before registration

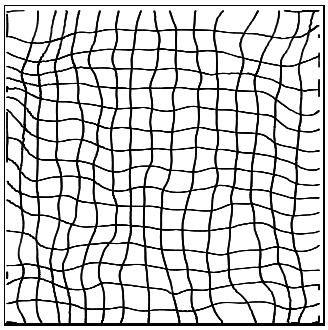
(e) error after registration



(f) estimated contrast

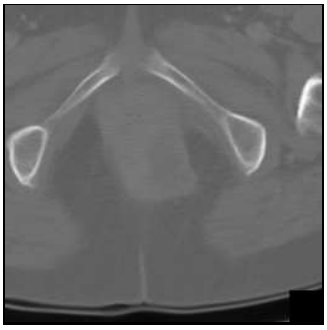


(g) estimated brightness

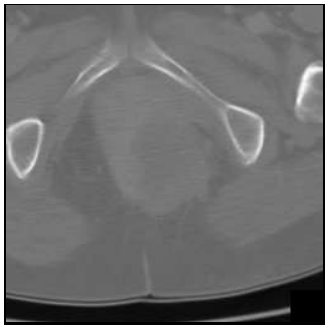


(h) estimated geometric map

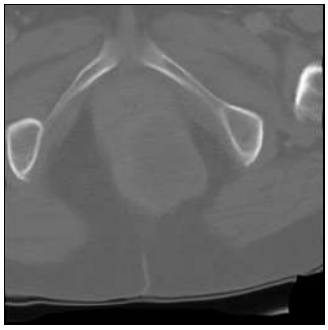
Figure 3.16: Clinical result: Chest X-rays.



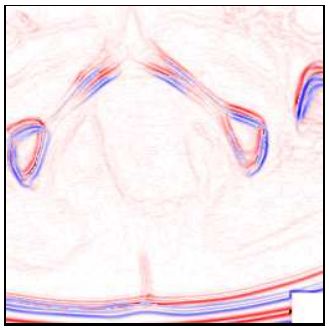
(a) source



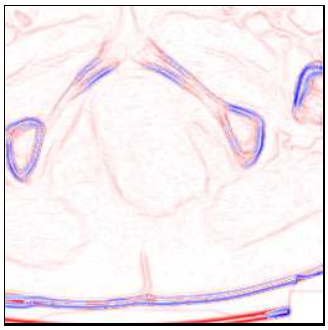
(b) target



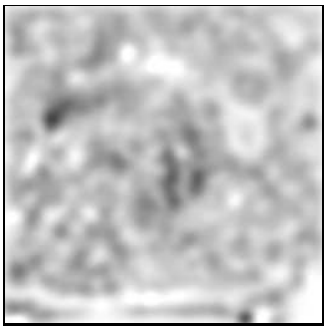
(c) registered source



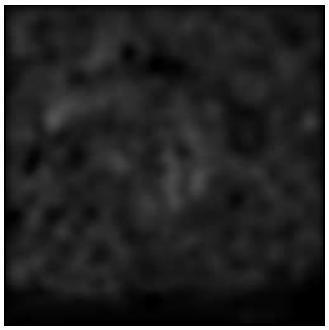
(d) error before registration



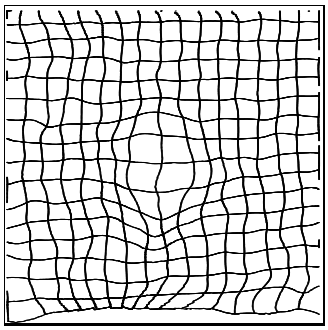
(e) error after registration



(f) estimated contrast

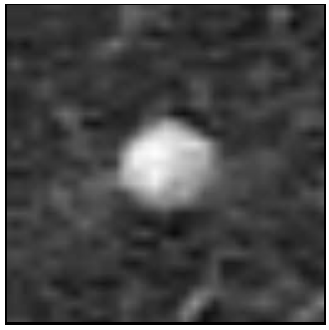


(g) estimated brightness



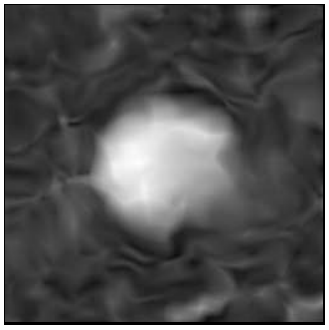
(h) estimated geometric map

Figure 3.17: Clinical result: Pelvic tumor.

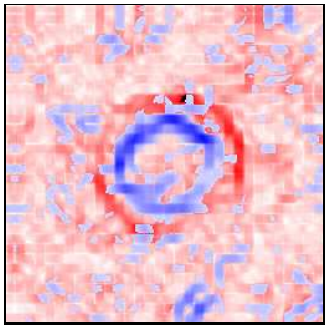


(a) source

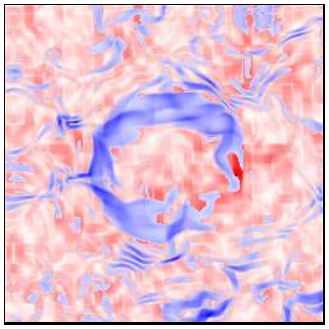
(b) target



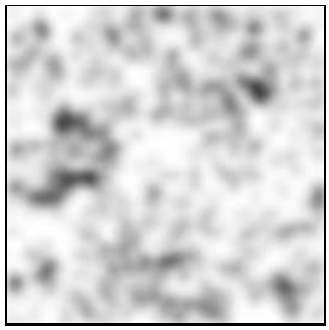
(c) registered source



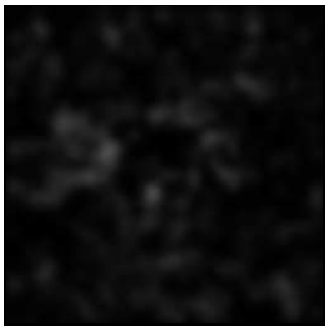
(d) error before registration



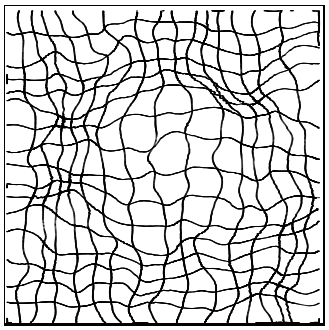
(e) error after registration



(f) estimated contrast



(g) estimated brightness



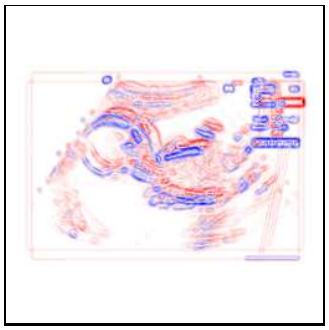
(h) estimated geometric map

Figure 3.18: Clinical result: Lung tumor.

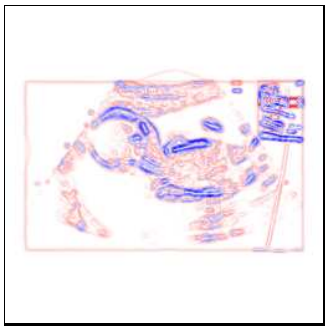
(a) source

(b) target

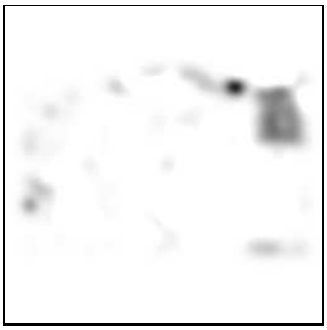
(c) registered source



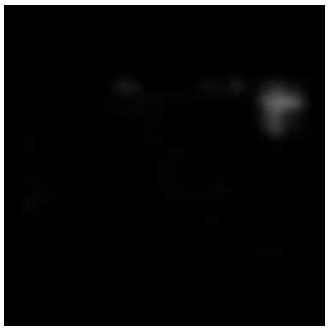
(d) error before registration



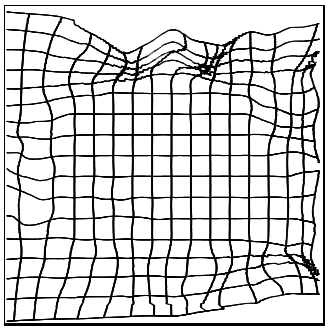
(e) error after registration



(f) estimated contrast

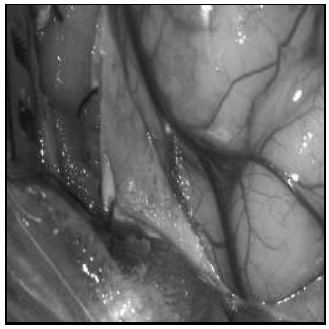


(g) estimated brightness

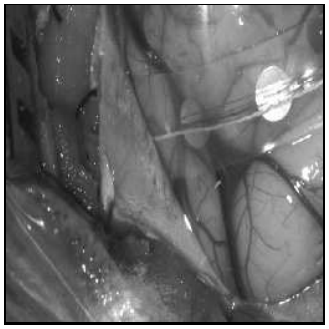


(h) estimated geometric map

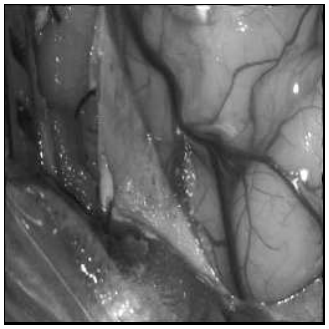
Figure 3.19: Clinical result: Ultrasound of baby, same subject.



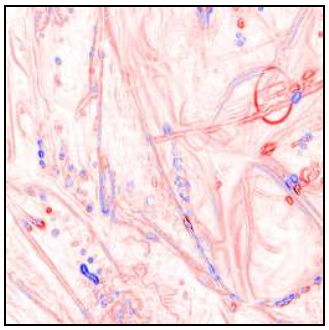
(a) source



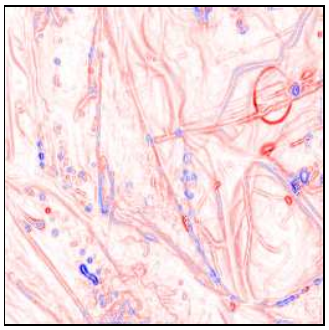
(b) target



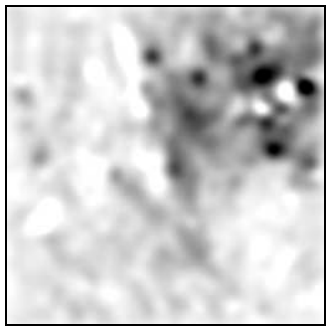
(c) registered source



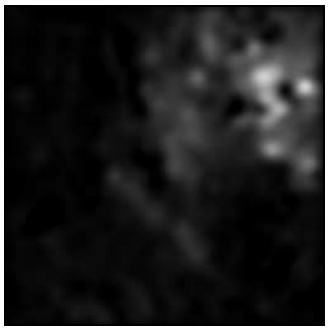
(d) error before registration



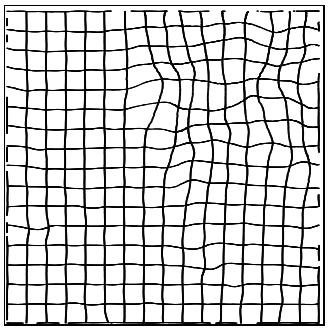
(e) error after registration



(f) estimated contrast

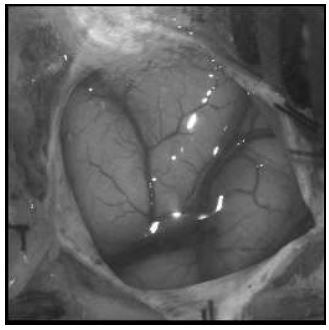


(g) estimated brightness

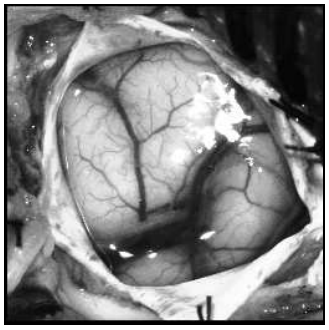


(h) estimated geometric map

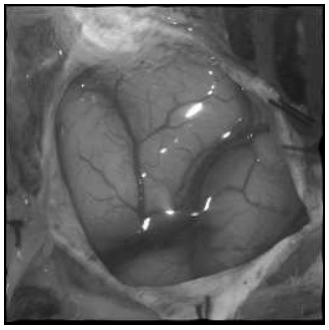
Figure 3.20: Clinical result: Photographs of brain during operation (1).



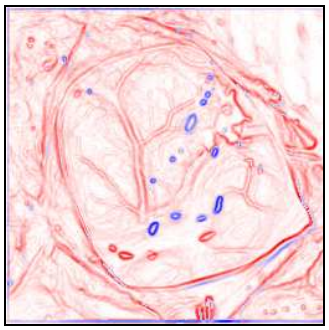
(a) source



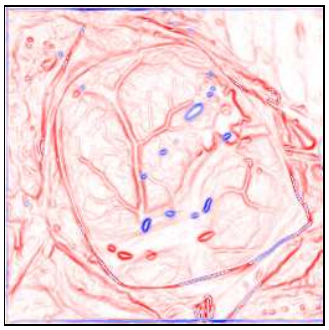
(b) target



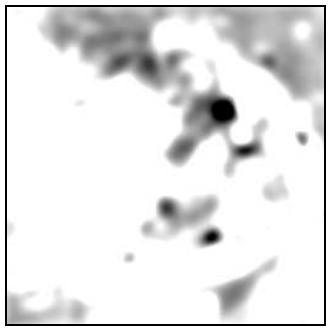
(c) registered source



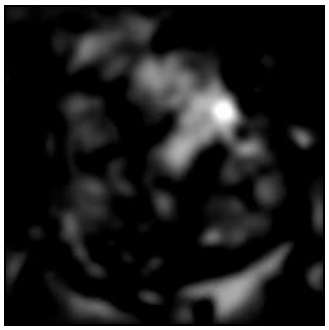
(d) error before registration



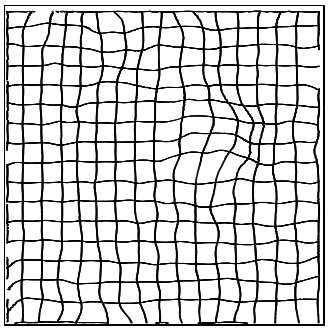
(e) error after registration



(f) estimated contrast

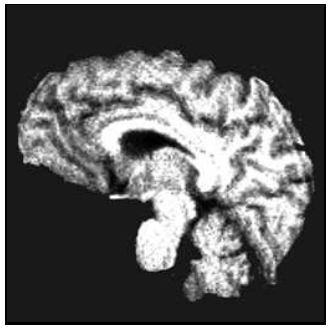


(g) estimated brightness

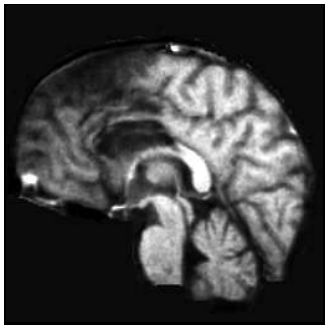


(h) estimated geometric map

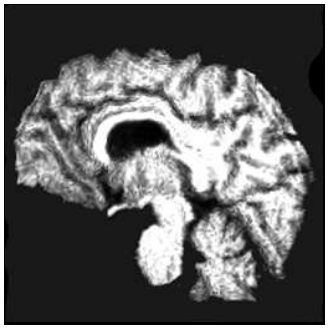
Figure 3.21: Clinical result: Photographs of brain during operation (2).



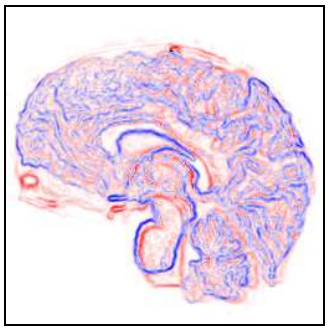
(a) source



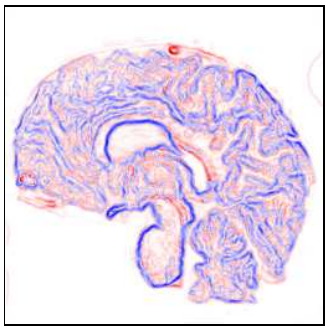
(b) target



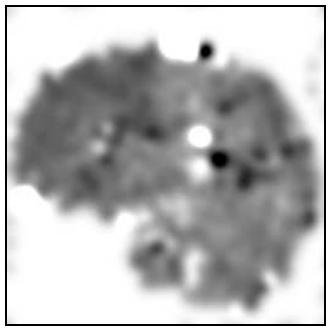
(c) registered source



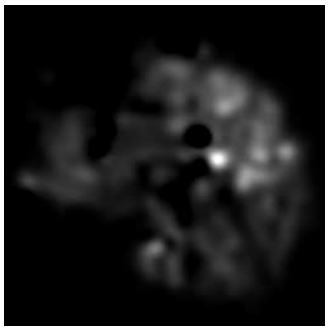
(d) error before registration



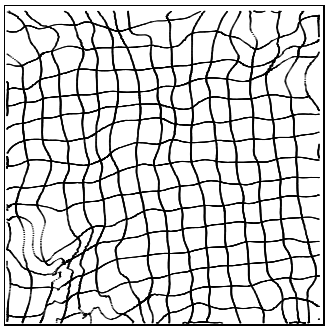
(e) error after registration



(f) estimated contrast

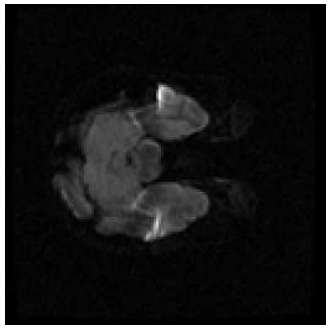


(g) estimated brightness

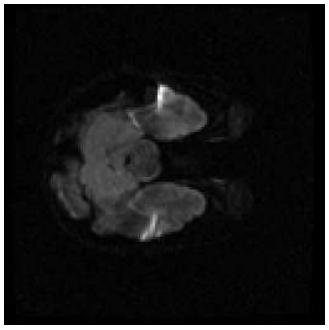


(h) estimated geometric map

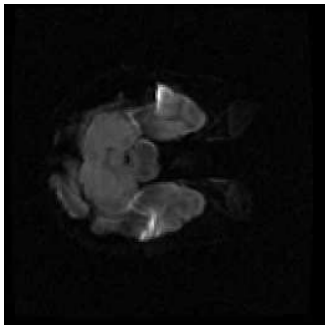
Figure 3.22: Clinical result: MRI sagitals pre/post operation.



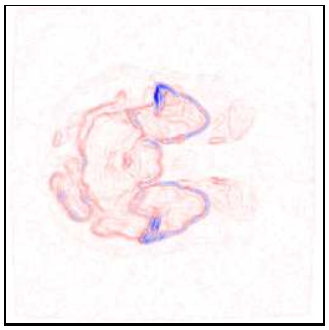
(a) source



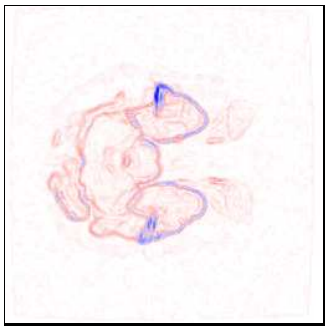
(b) target



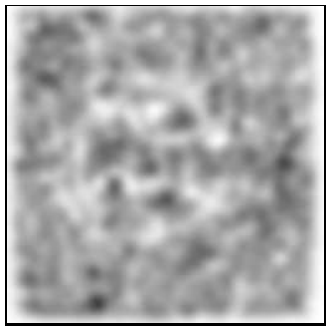
(c) registered source



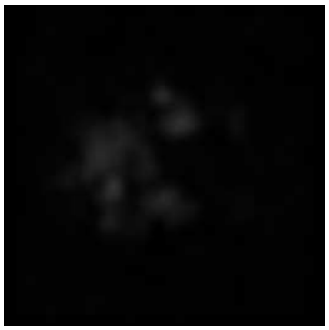
(d) error before registration



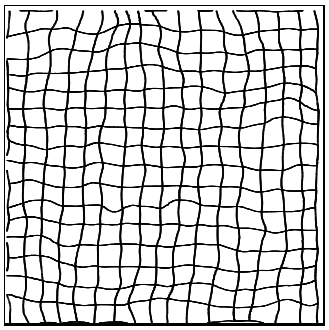
(e) error after registration



(f) estimated contrast

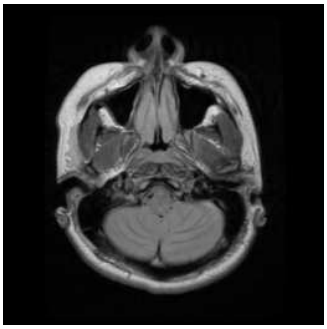


(g) estimated brightness

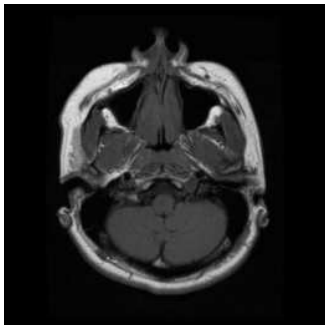


(h) estimated geometric map

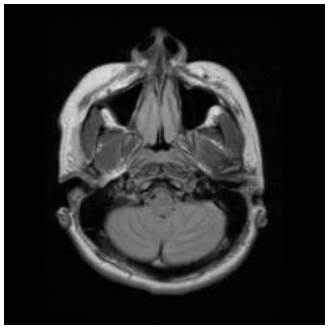
Figure 3.23: Clinical result: EPI expermiment.



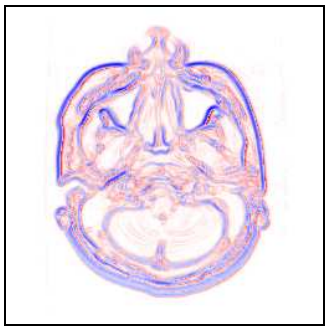
(a) source



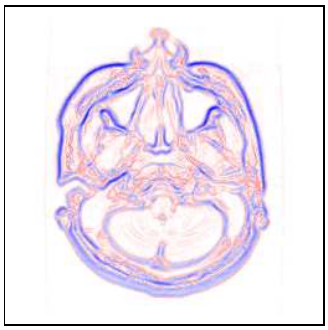
(b) target



(c) registered source

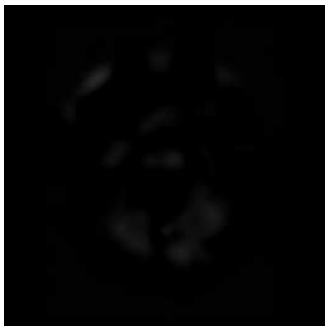


(d) error before registration

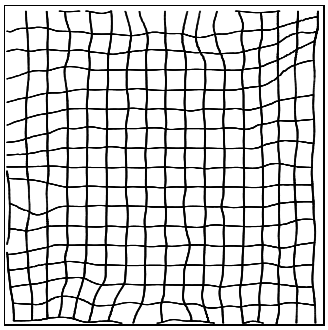


(e) error after registration

(f) estimated contrast

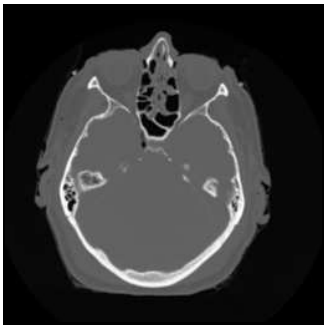


(g) estimated brightness

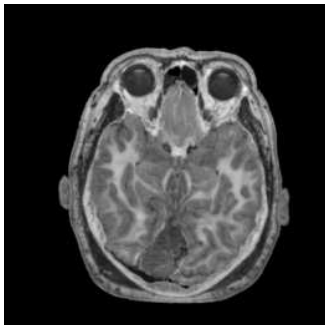


(h) estimated geometric map

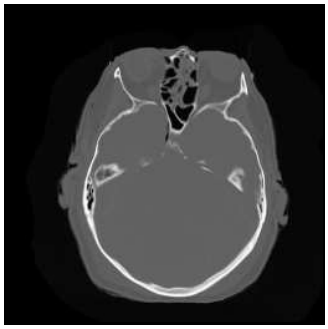
Figure 3.24: Clinical result: Photon-density/MRI T1.



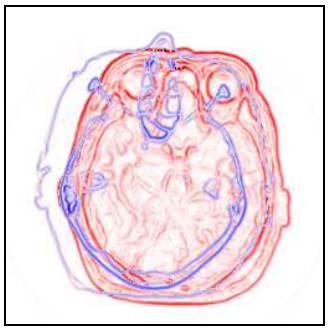
(a) source



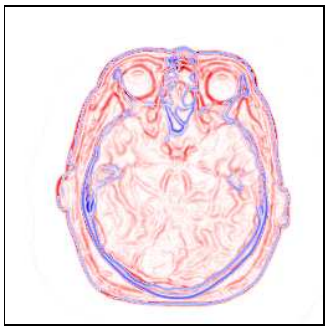
(b) target



(c) registered source

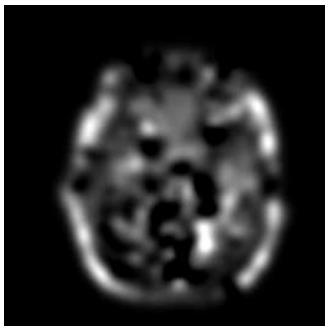


(d) error before registration

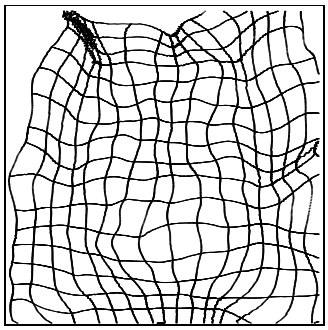


(e) error after registration

(f) estimated contrast

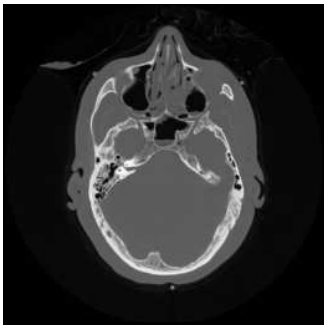


(g) estimated brightness

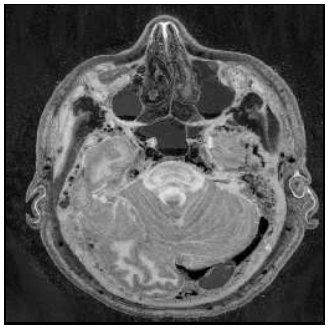


(h) estimated geometric map

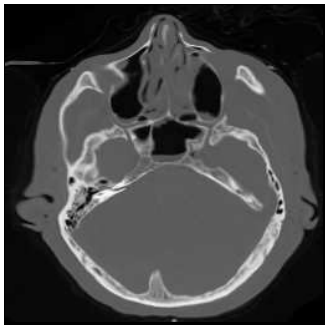
Figure 3.25: Clinical result: CT/photograph, axial (1).



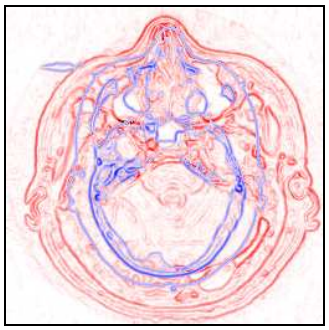
(a) source



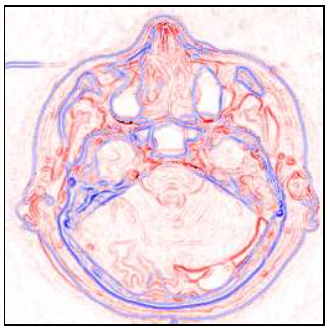
(b) target



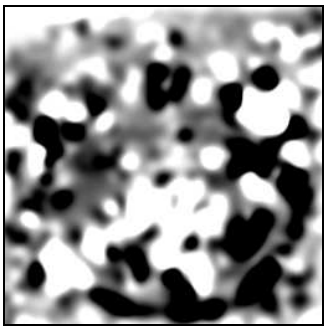
(c) registered source



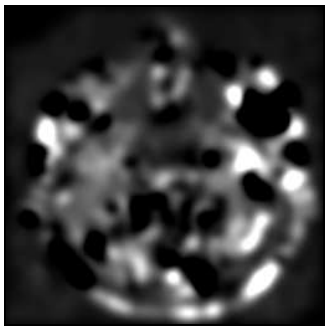
(d) error before registration



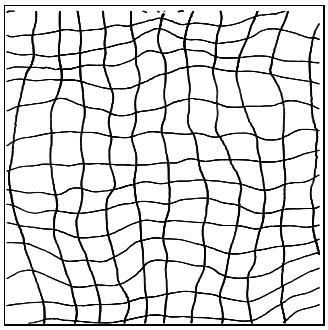
(e) error after registration



(f) estimated contrast

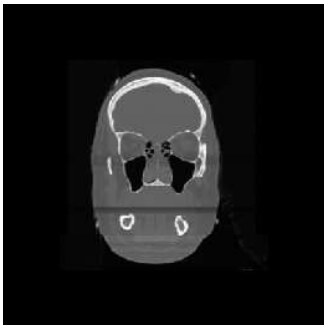


(g) estimated brightness

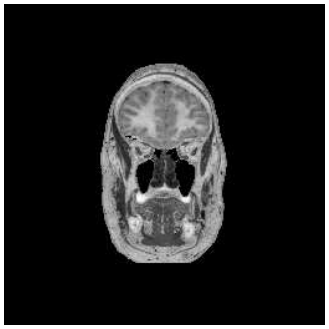


(h) estimated geometric map

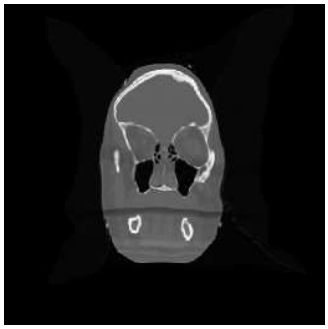
Figure 3.26: Clinical result: CT/photograph, axial (2).



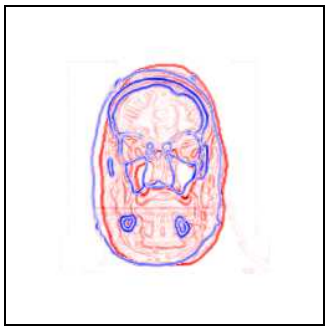
(a) source



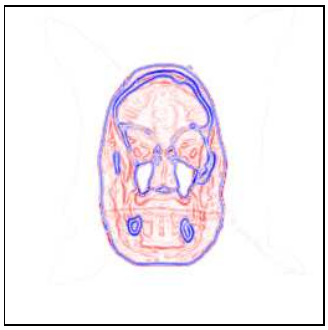
(b) target



(c) registered source

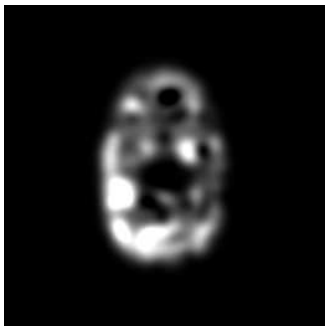


(d) error before registration

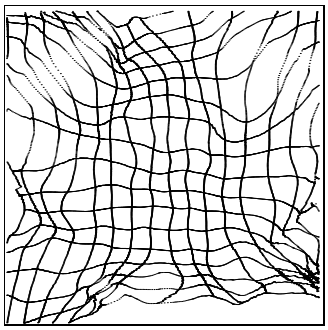


(e) error after registration

(f) estimated contrast



(g) estimated brightness



(h) estimated geometric map

Figure 3.27: Clinical result: CT/photograph, axial (3).

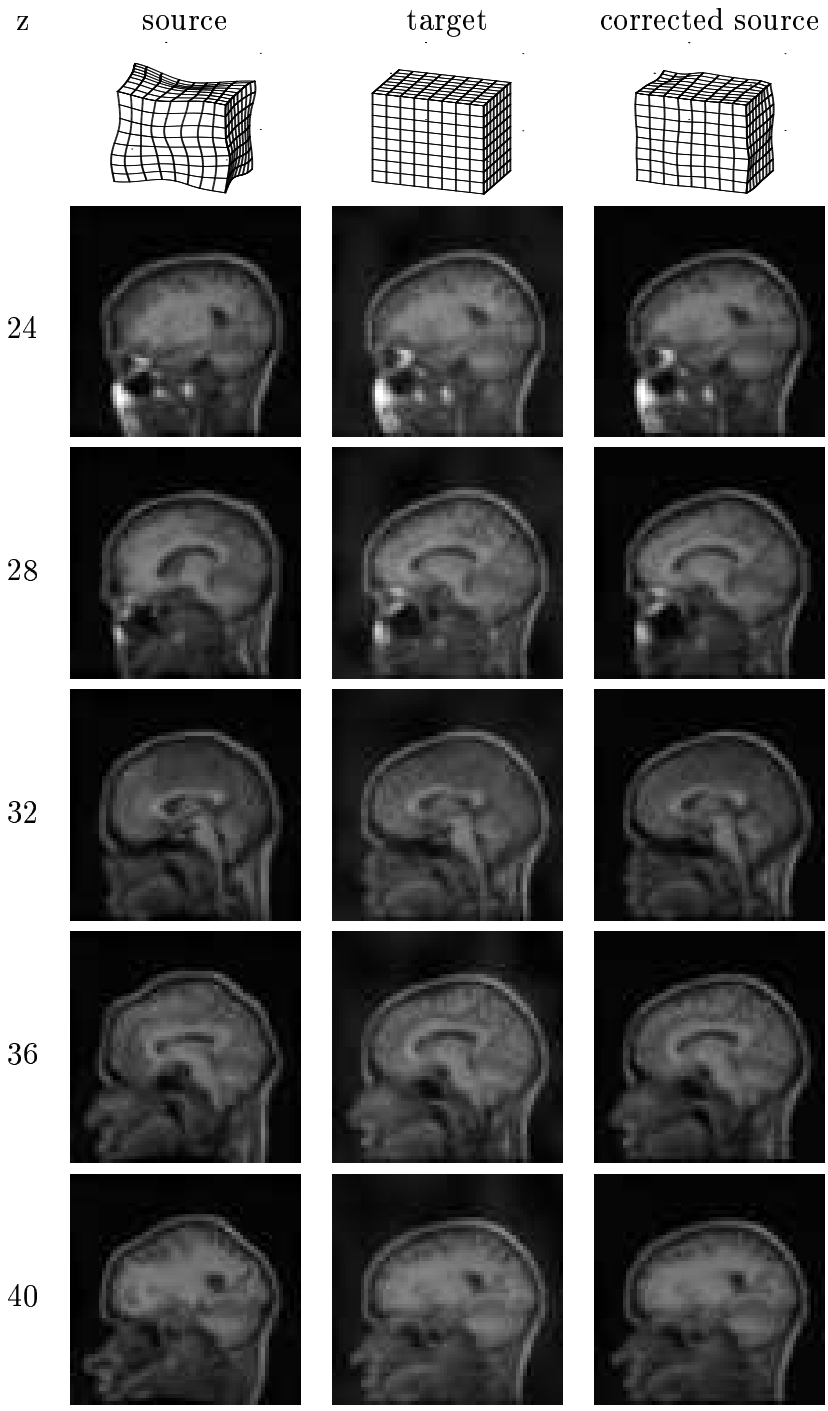


Figure 3.28: 3-D synthetic result: MR brain volumes (1). See also Figure 3.36.

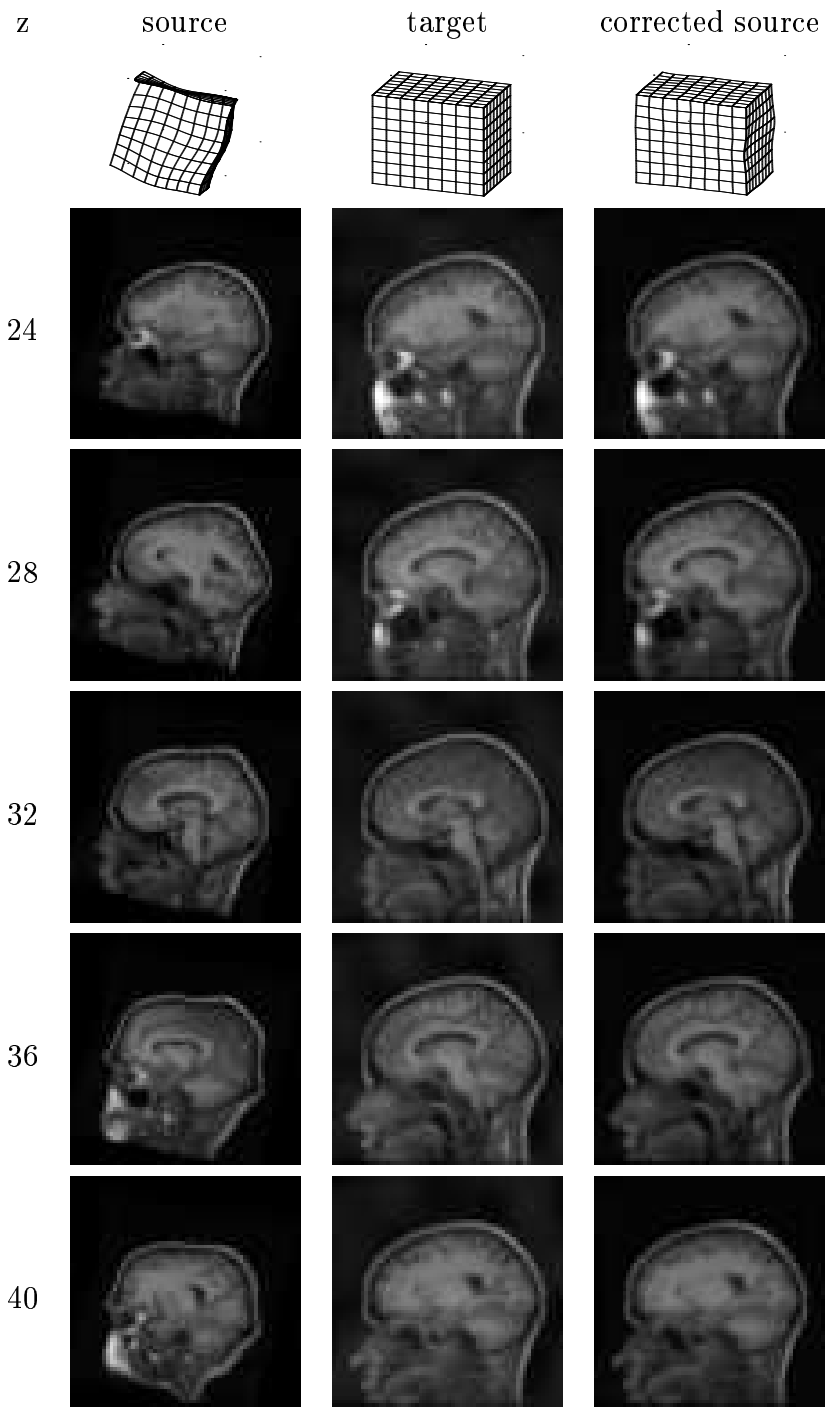


Figure 3.29: 3-D synthetic result: MR brain volumes (2). See also Figure 3.36.

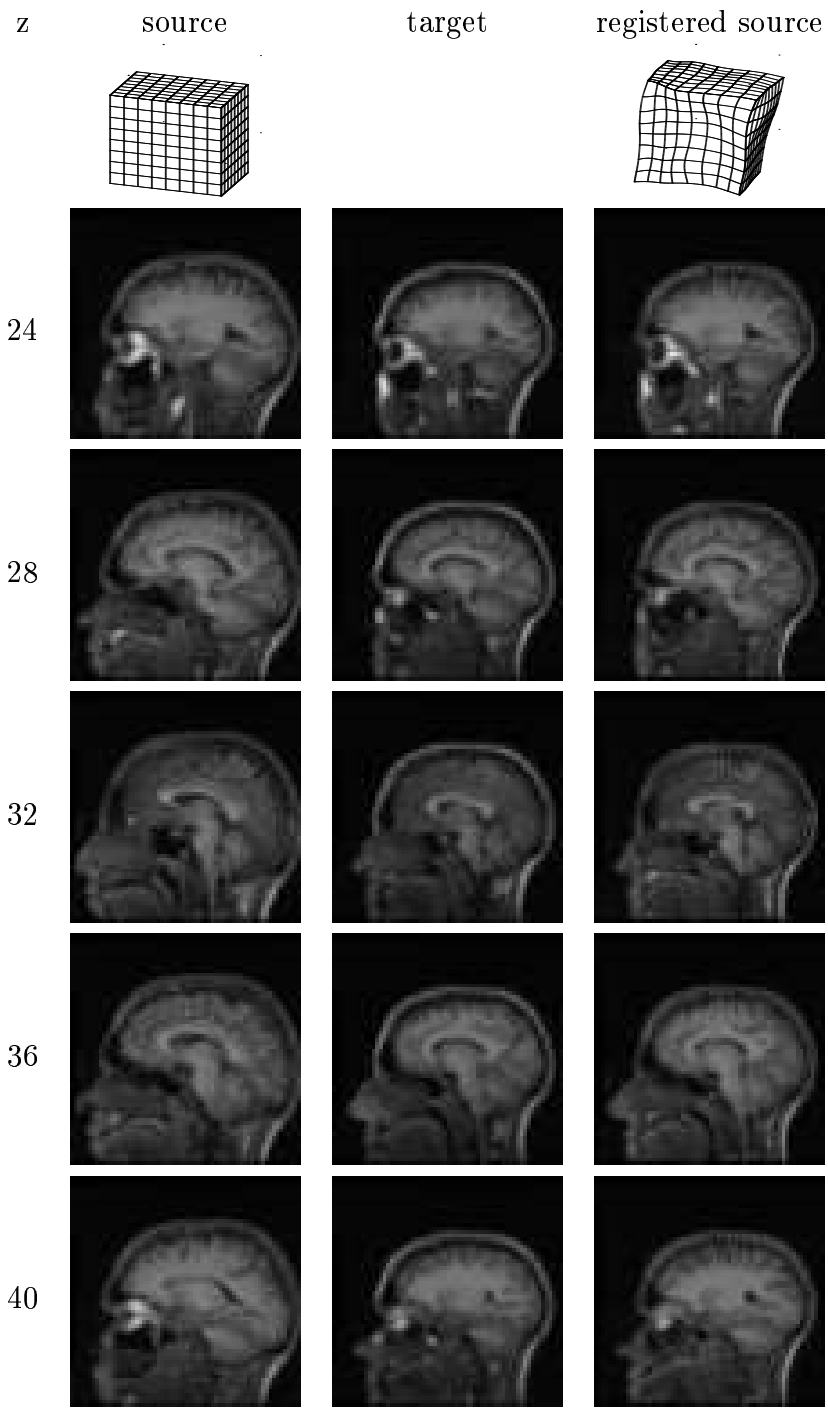


Figure 3.30: 3-D clinical result: MR brain volumes (1). See also Figure 3.36.

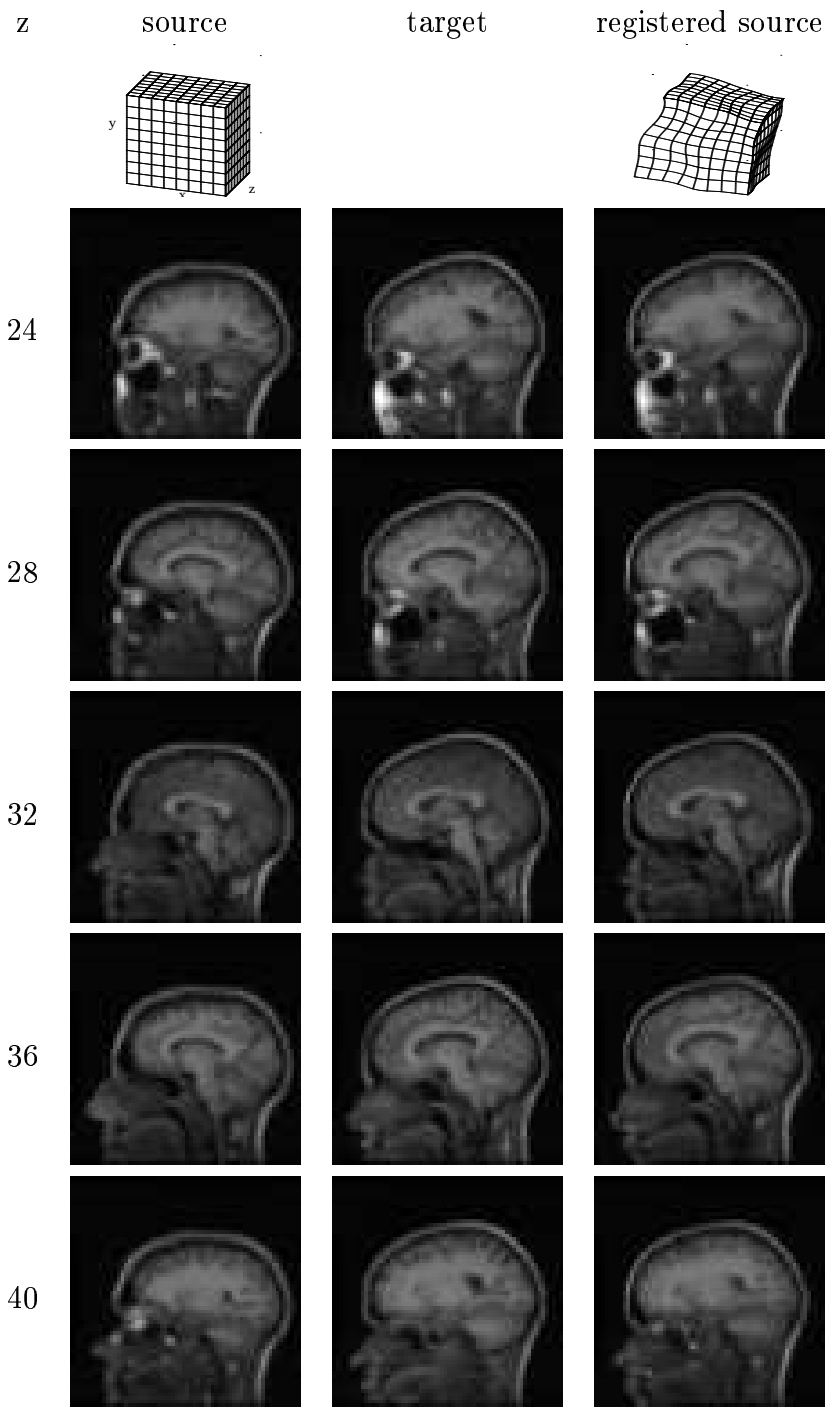


Figure 3.31: 3-D clinical result: MR brain volumes (2). See also Figure 3.36.

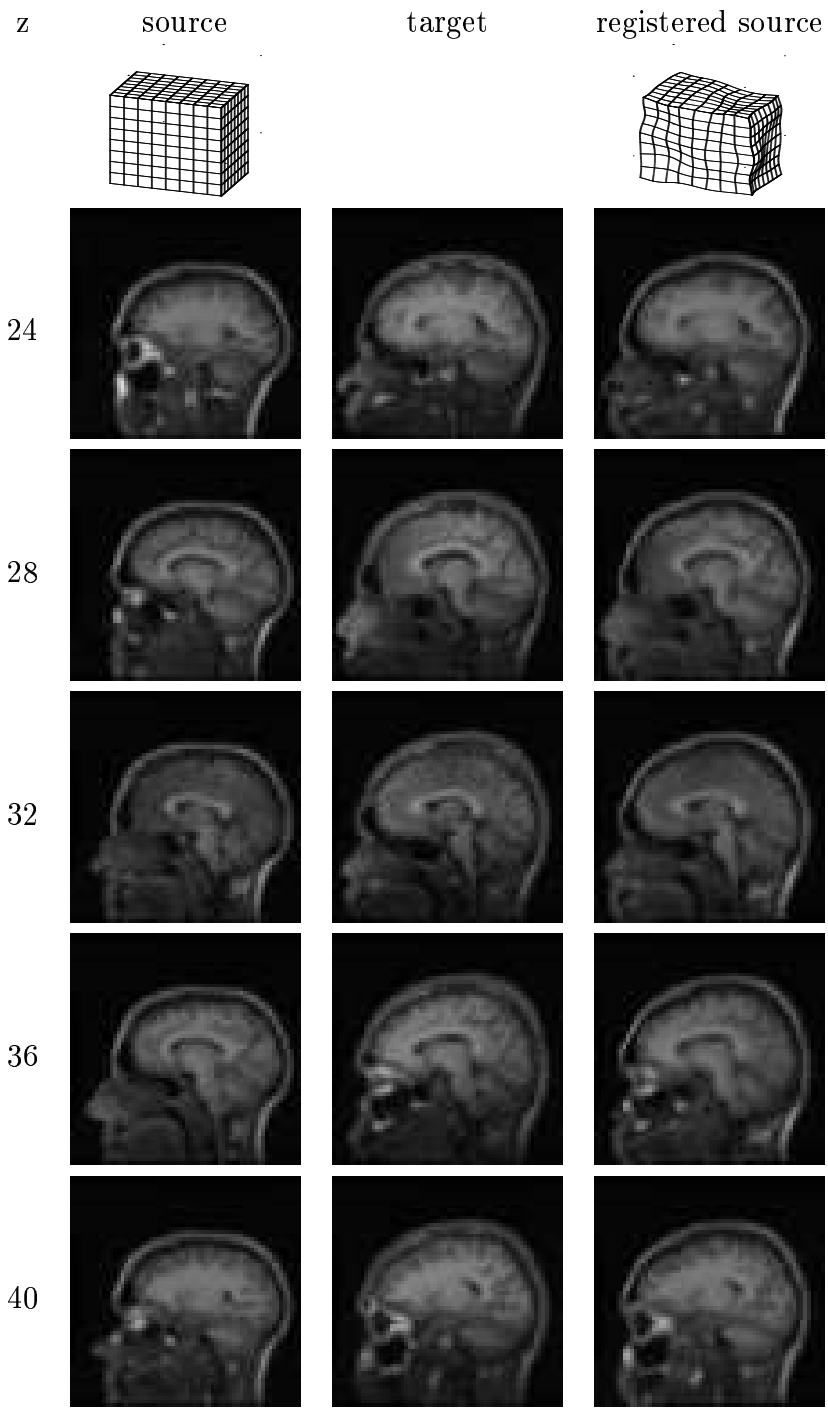


Figure 3.32: 3-D clinical result: MR brain volumes (3). See also Figure 3.37.

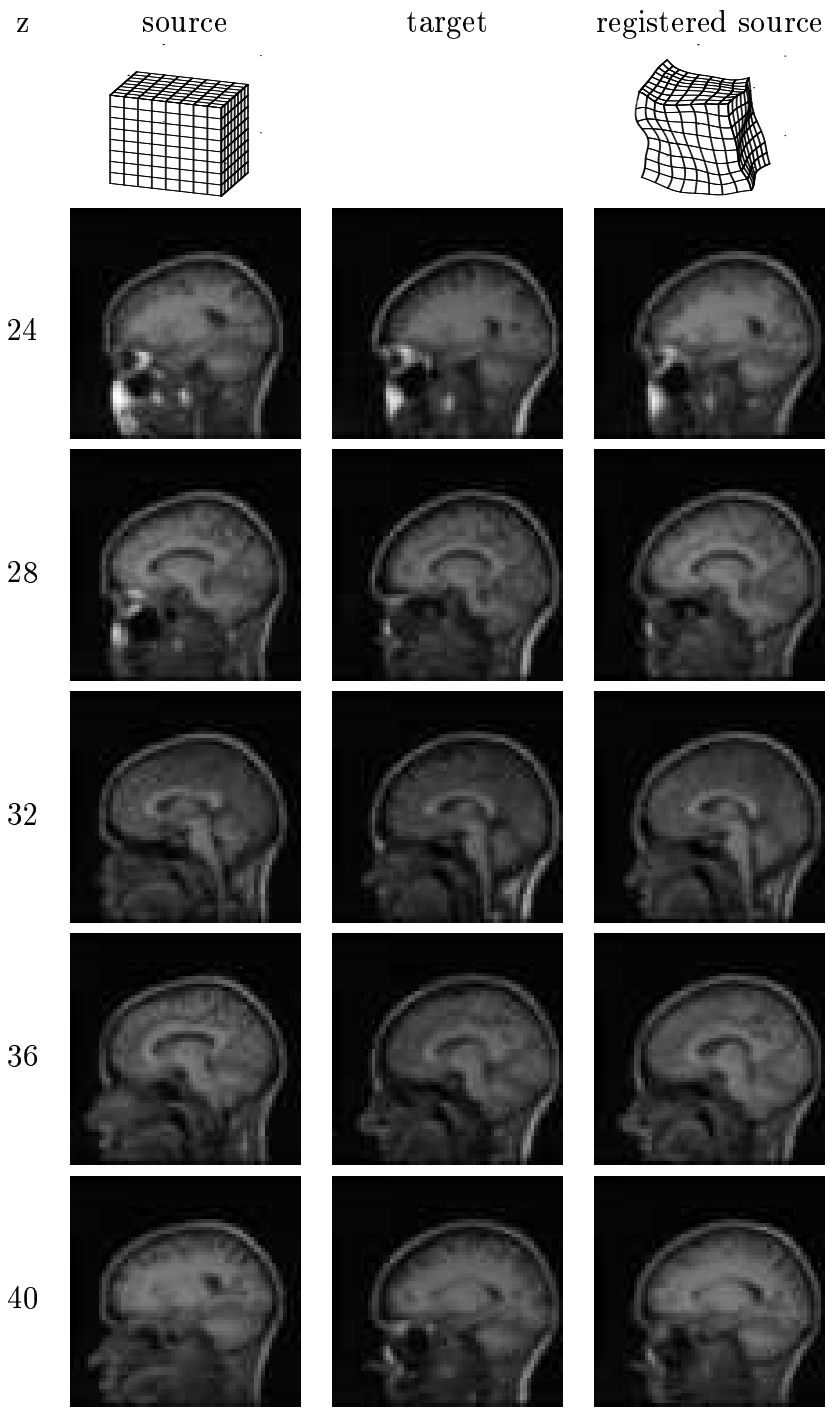


Figure 3.33: 3-D clinical result: MR brain volumes (4). See also Figure 3.37.

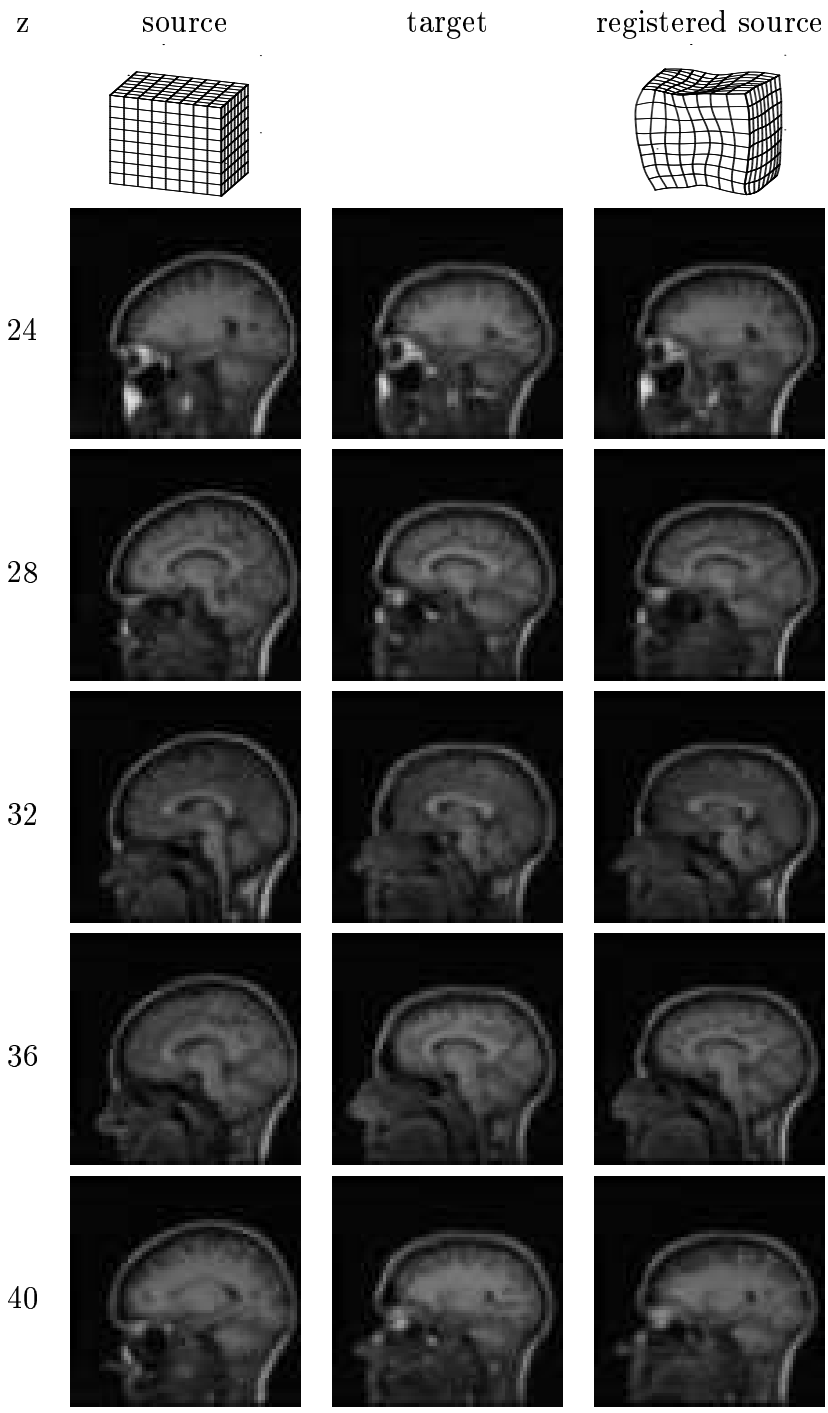


Figure 3.34: 3-D clinical result: MR brain volumes (5). See also Figure 3.37.

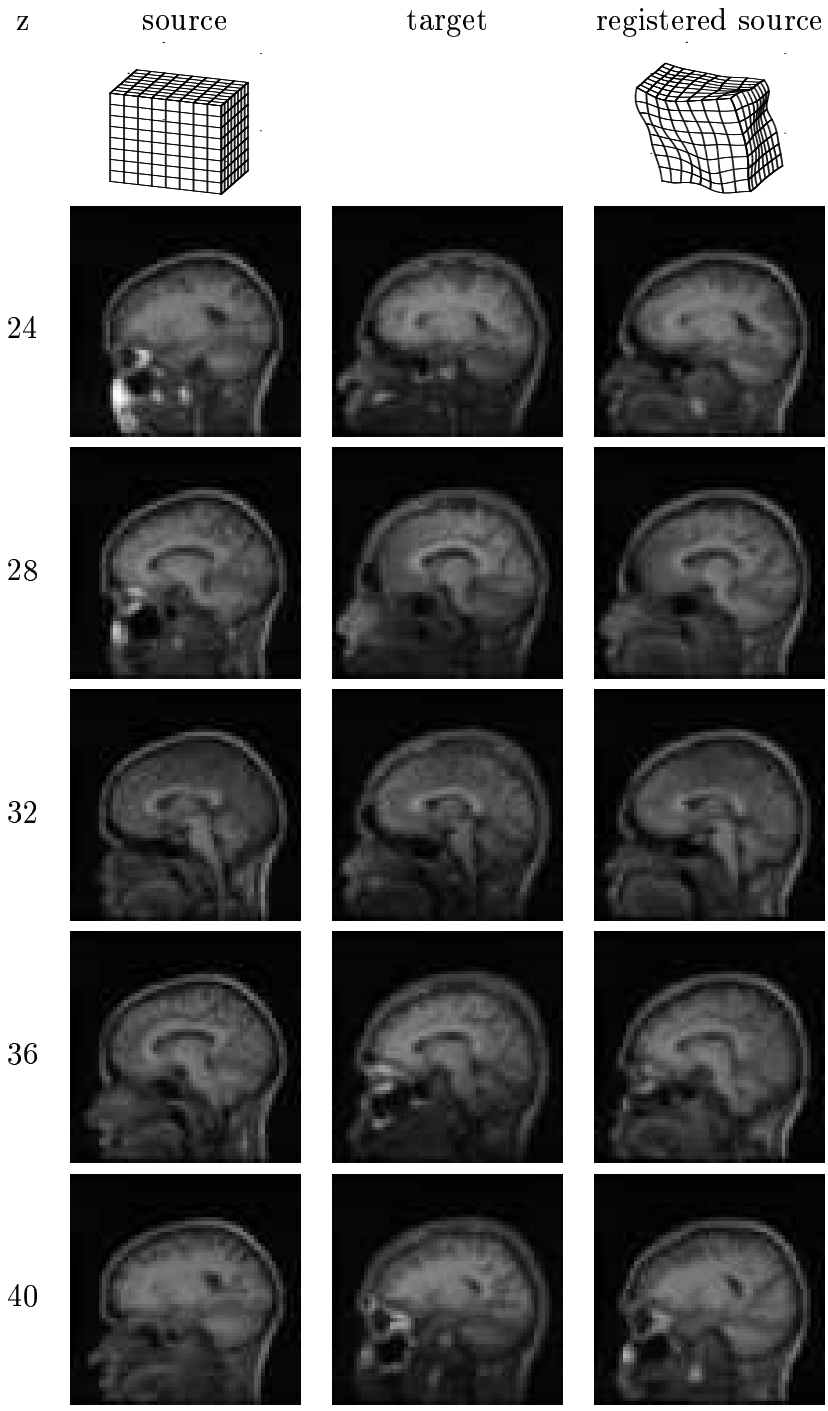


Figure 3.35: 3-D clinical result: MR brain volumes (6). See also Figure 3.37.

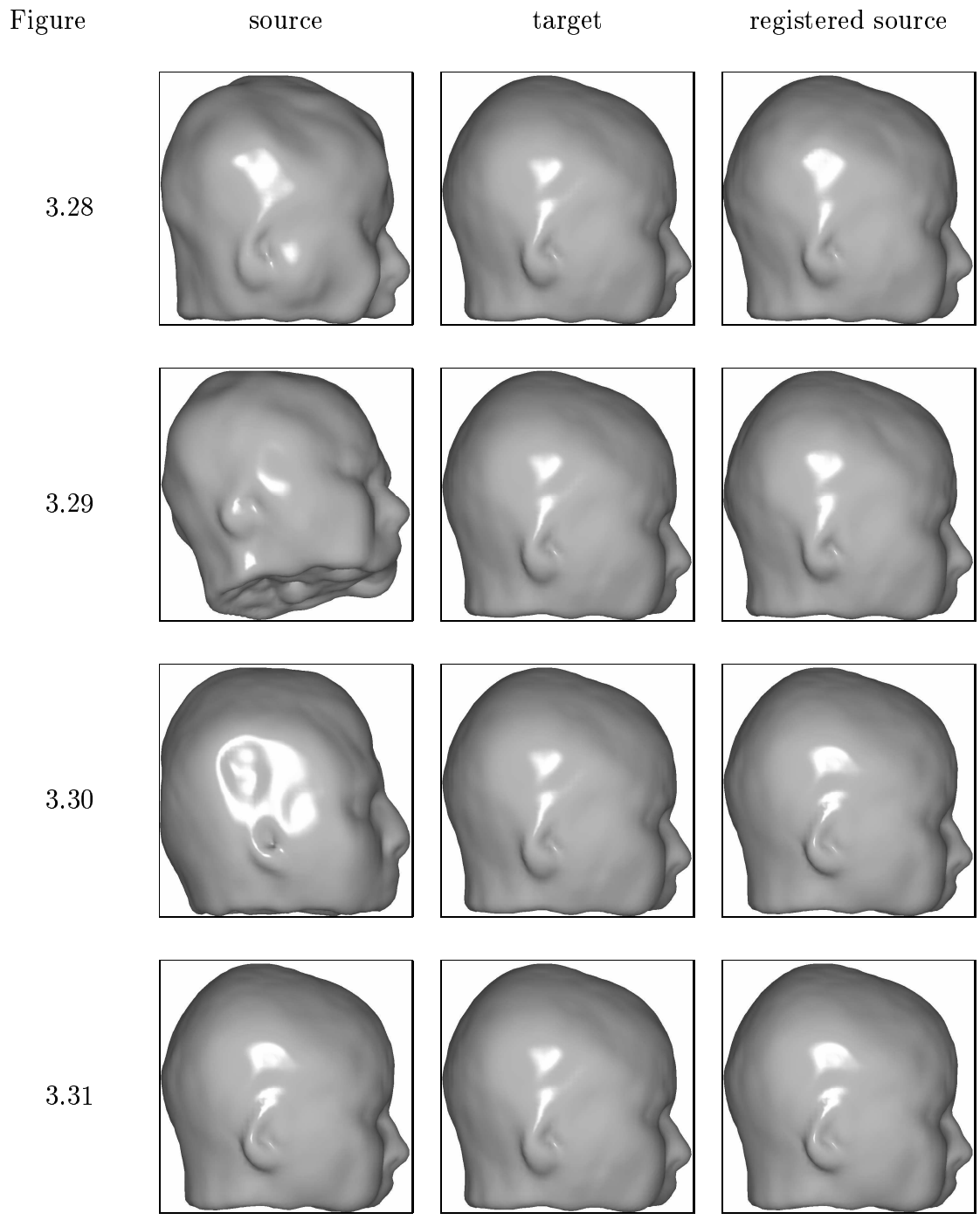


Figure 3.36: 3-D results (iso-surface view). Figure numbers in the first column refer to the corresponding slice views.

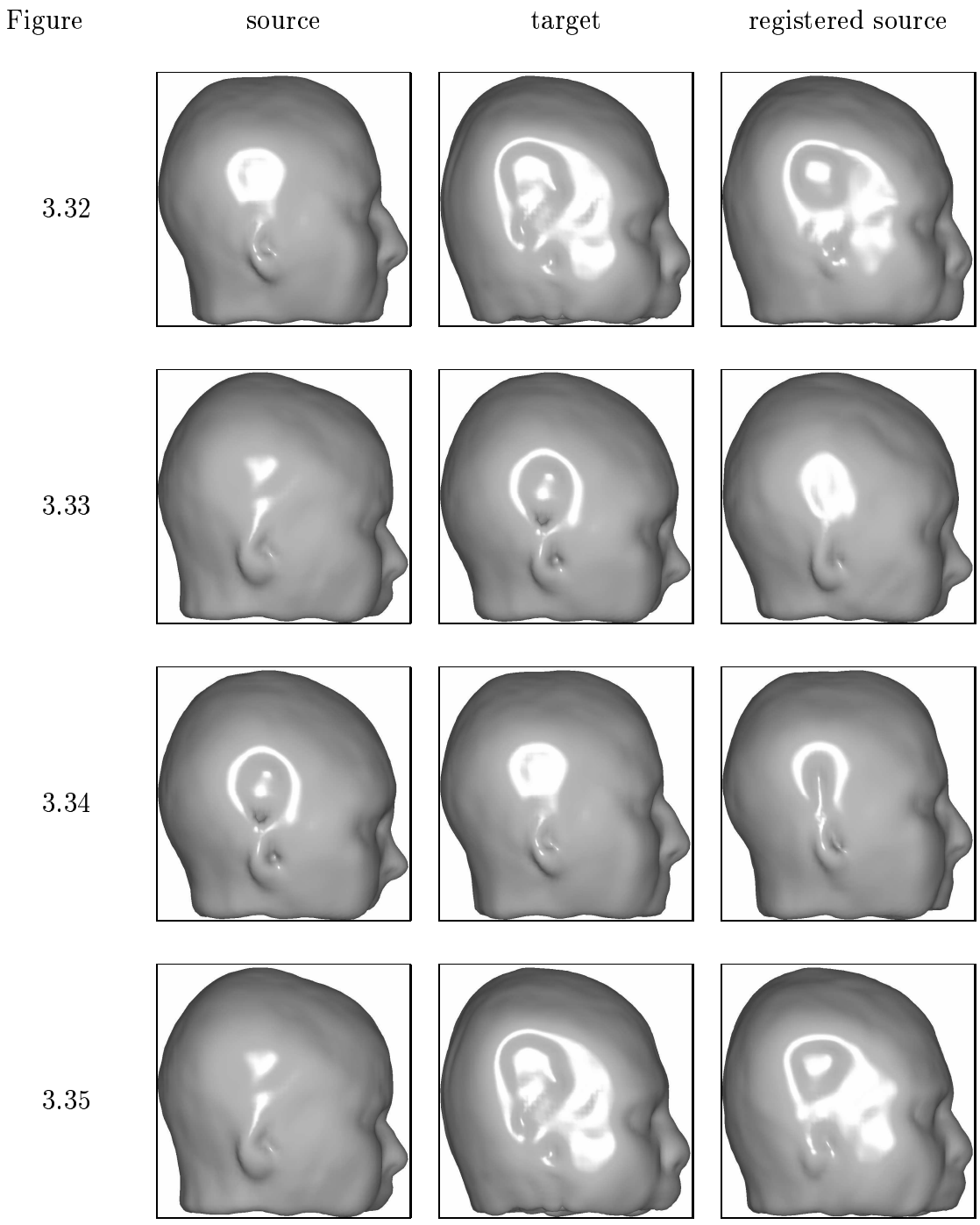
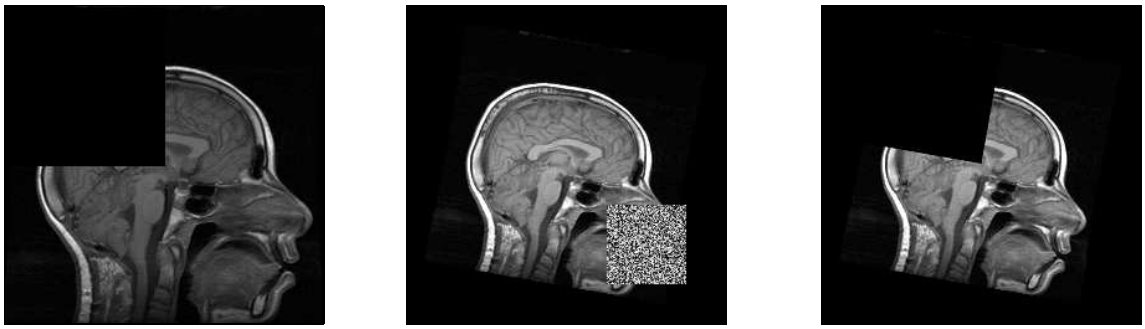
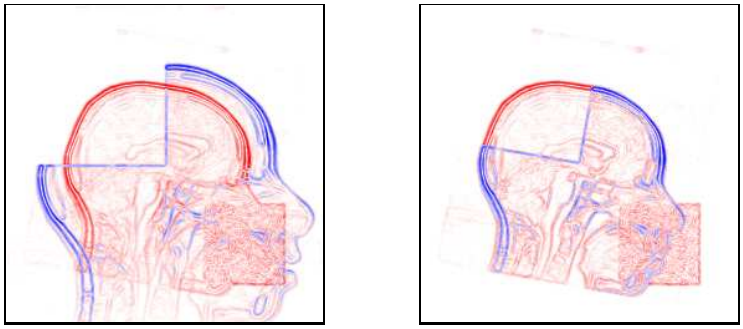


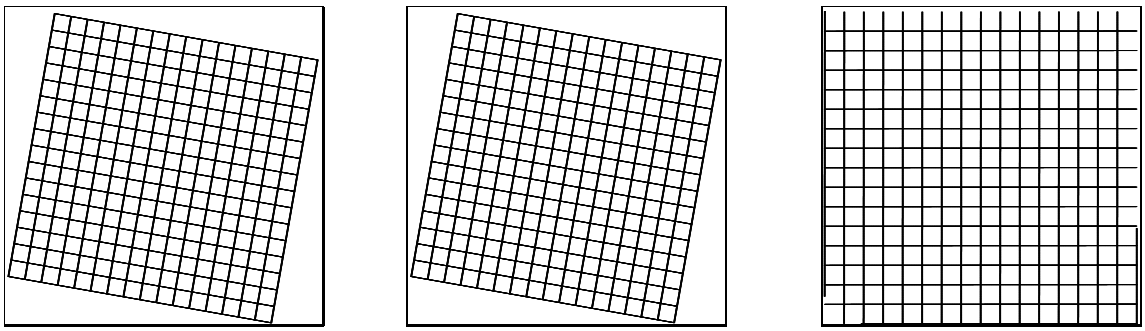
Figure 3.37: 3-D results (iso-surface view). Figure numbers in the first column refer to the corresponding slice views.



(a) source (b) target (c) registered source

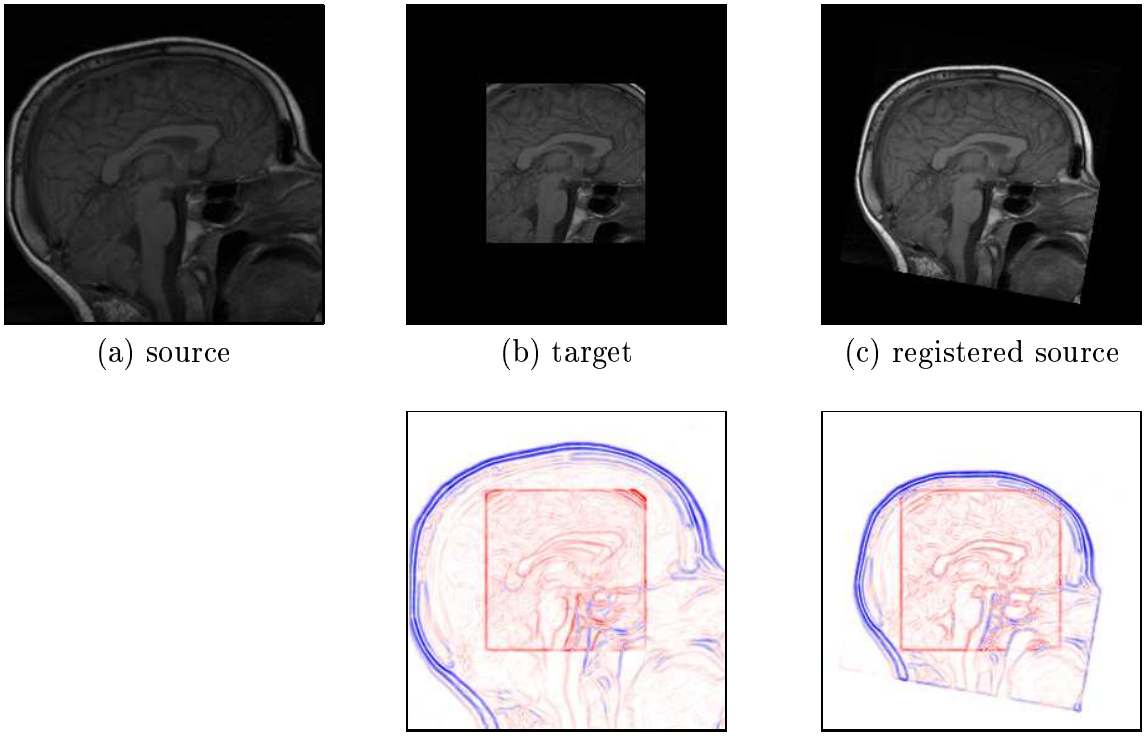


(d) error before registration (e) error after registration



(f) appled geometric map (g) estimated geometric map (h) rectified geometric map

Figure 3.38: EM Synthetic global result: Sagital images with missing data and noise added.

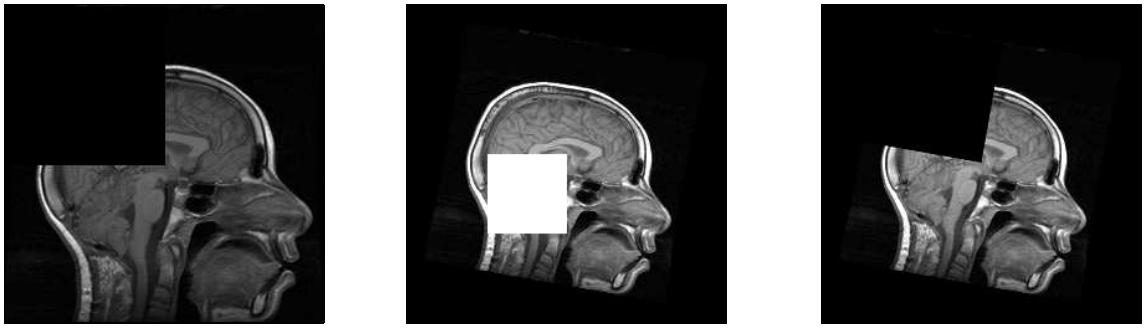


(a) source (b) target (c) registered source

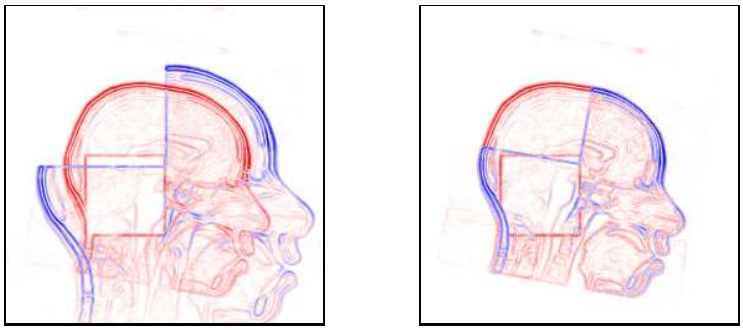
(d) error before registration (e) error after registration

(f) appled geometric map (g) estimated geometric map (h) rectified geometric map

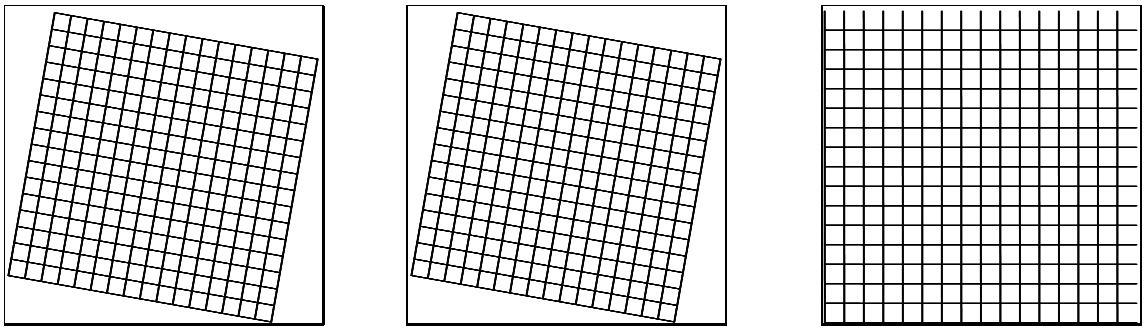
Figure 3.39: EM Synthetic global result: Partial sagital image.



(a) source (b) target (c) registered source

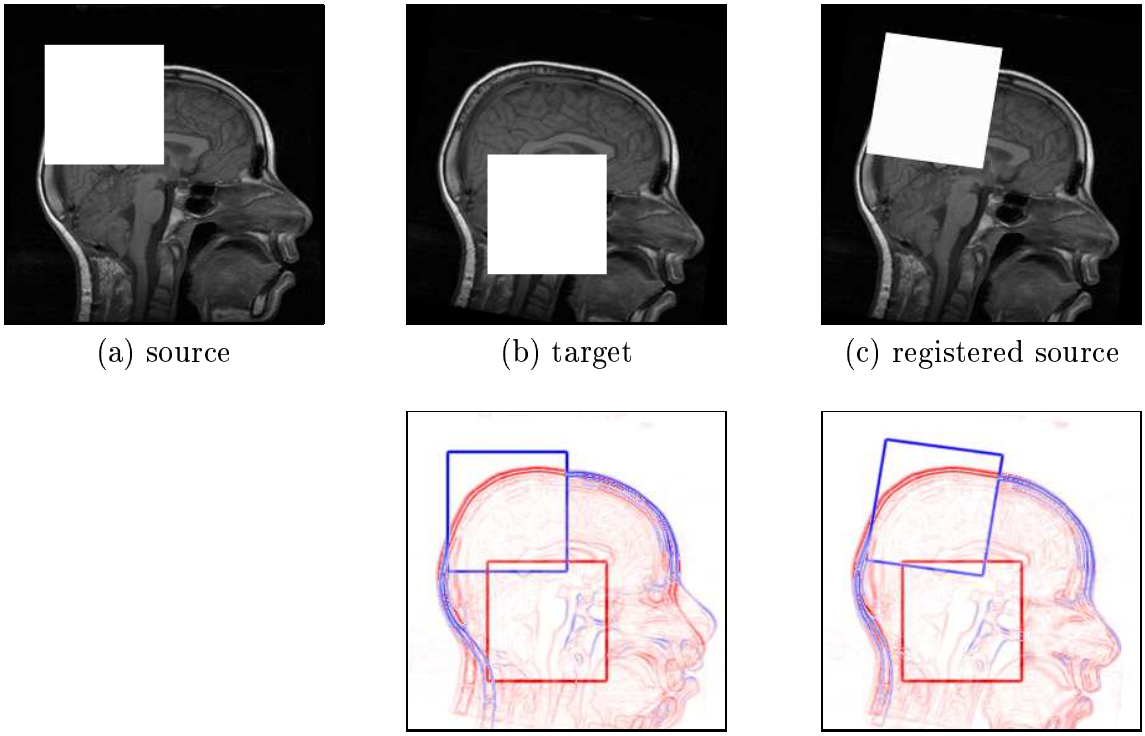


(d) error before registration (e) error after registration



(f) appled geometric map (g) estimated geometric map (h) rectified geometric map

Figure 3.40: EM Synthetic global result: Sagital images with white and black squares.

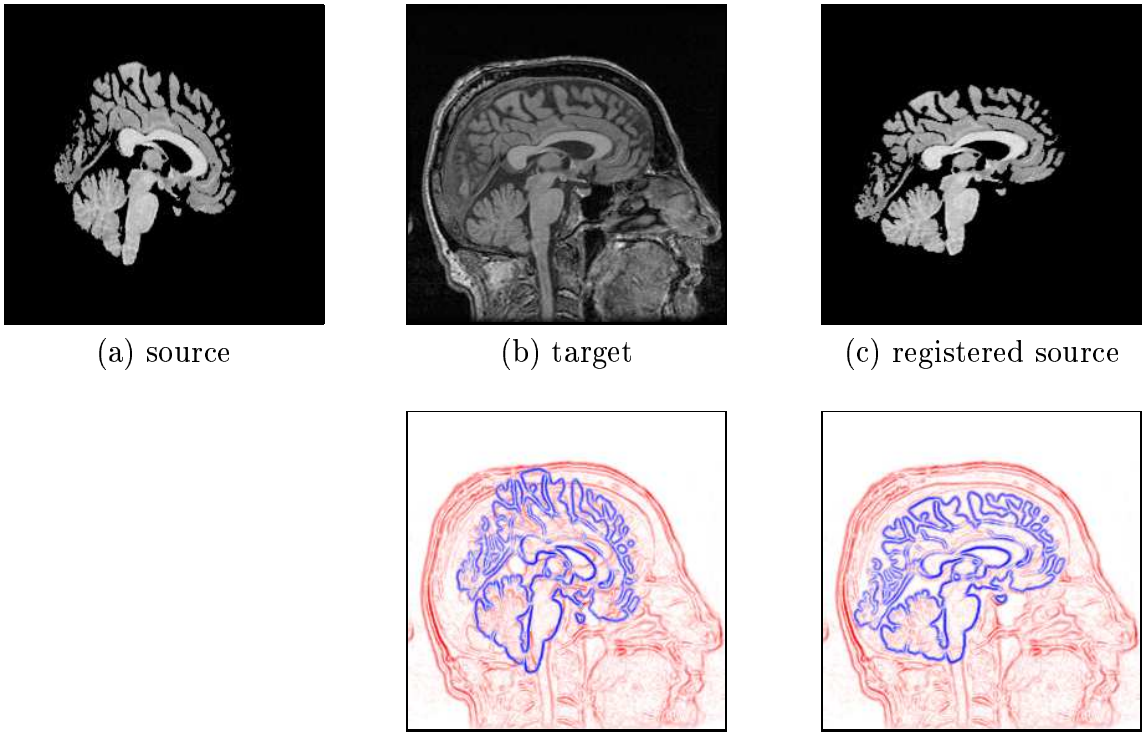


(a) source (b) target (c) registered source

(d) error before registration (e) error after registration

(f) appled geometric map (g) estimated geometric map (h) rectified geometric map

Figure 3.41: EM Synthetic global result: Sagital imges with white squares.

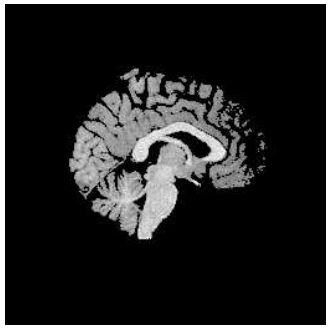


(a) source (b) target (c) registered source

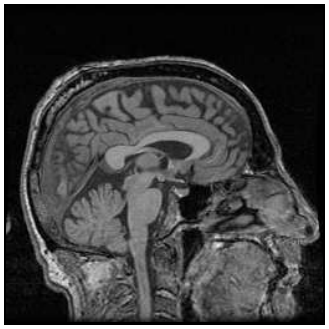
(d) error before registration (e) error after registration

(f) appled geometric map (g) estimated geometric map (h) rectified geometric map

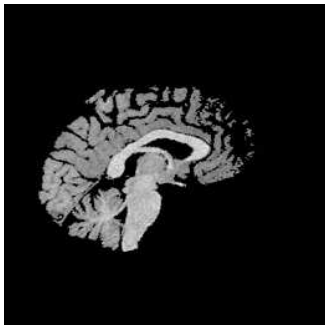
Figure 3.42: EM Synthetic global result: Segmented MR, same subject.



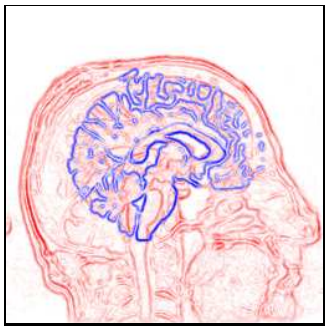
(a) source



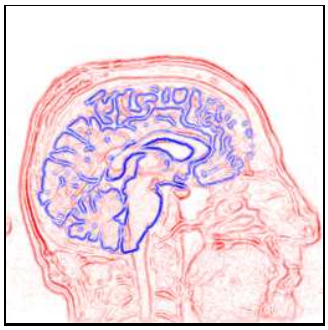
(b) target



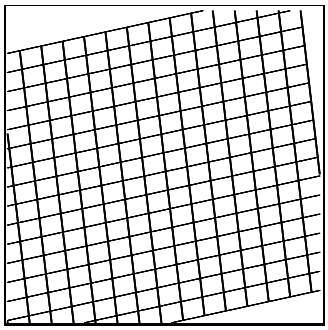
(c) registered source



(d) error before registration

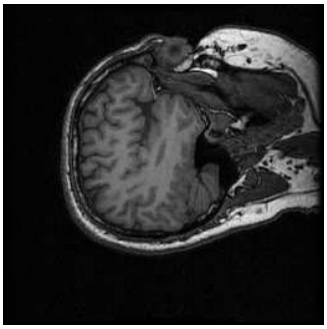


(e) error after registration

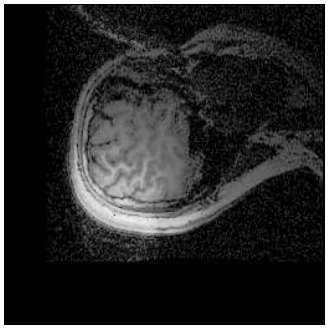


(f) estimated registration map

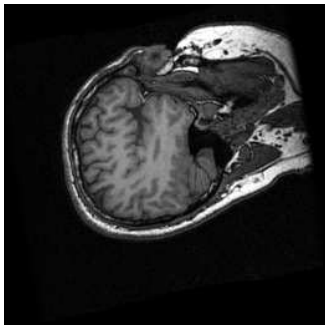
Figure 3.43: EM Clinical global result: Segmented MR, different subjects.



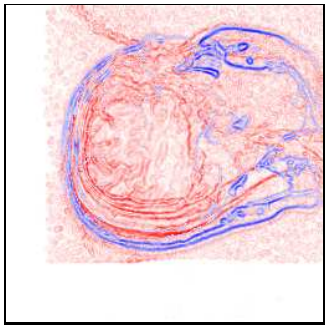
(a) source



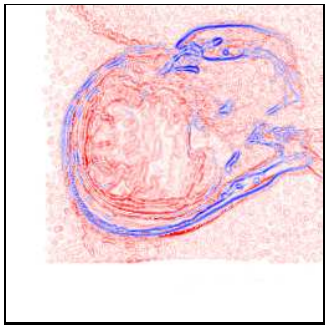
(b) target



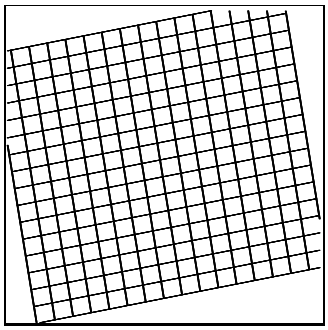
(c) registered source



(d) error before registration

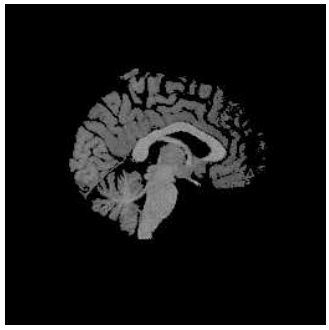


(e) error after registration

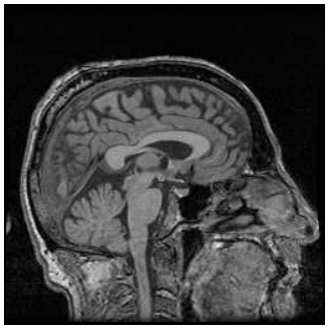


(f) estimated registration map

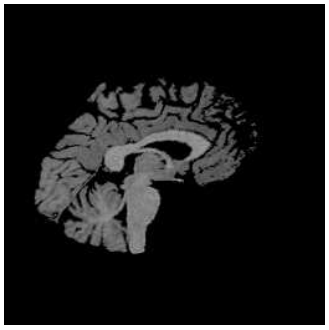
Figure 3.44: EM Clinical global result: MR with MR coil, sagitals.



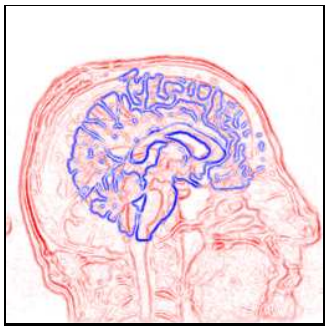
(a) source



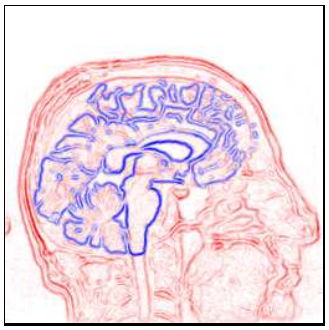
(b) target



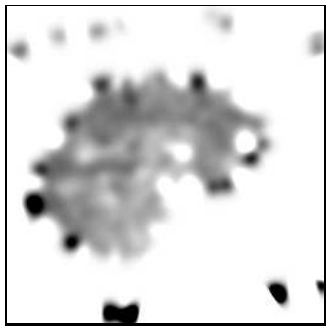
(c) registered source



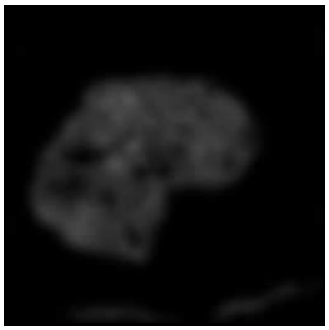
(d) error before registration



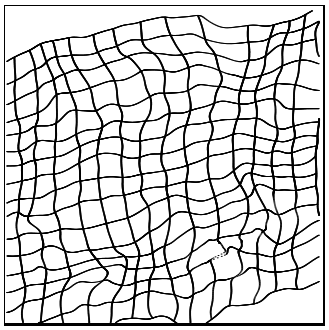
(e) error after registration



(f) estimated contrast

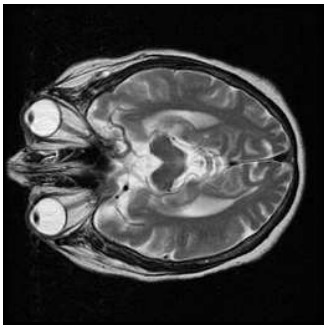


(g) estimated brightness

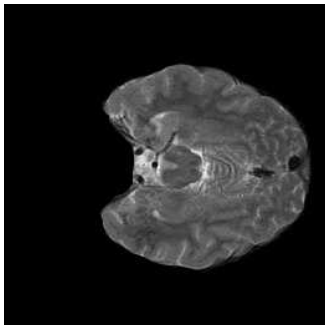


(h) estimated geometric map

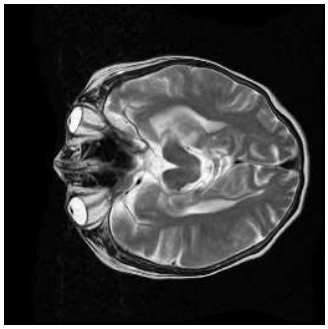
Figure 3.45: EM Clinical result: MRI sagitals, stripped skull.



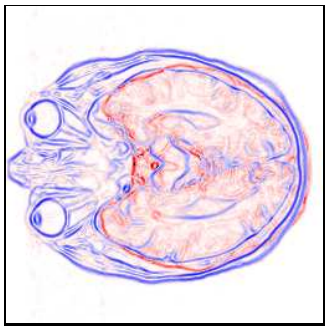
(a) source



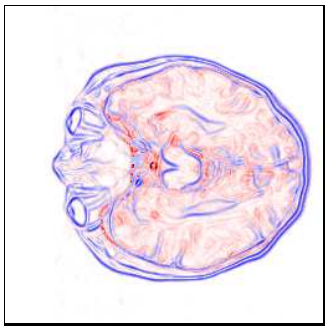
(b) target



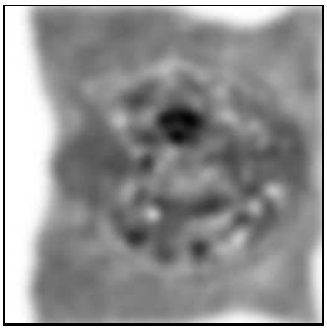
(c) registered source



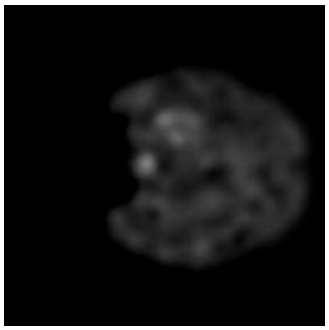
(d) error before registration



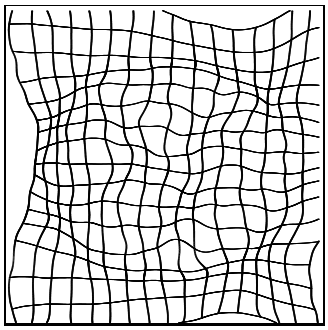
(e) error after registration



(f) estimated contrast

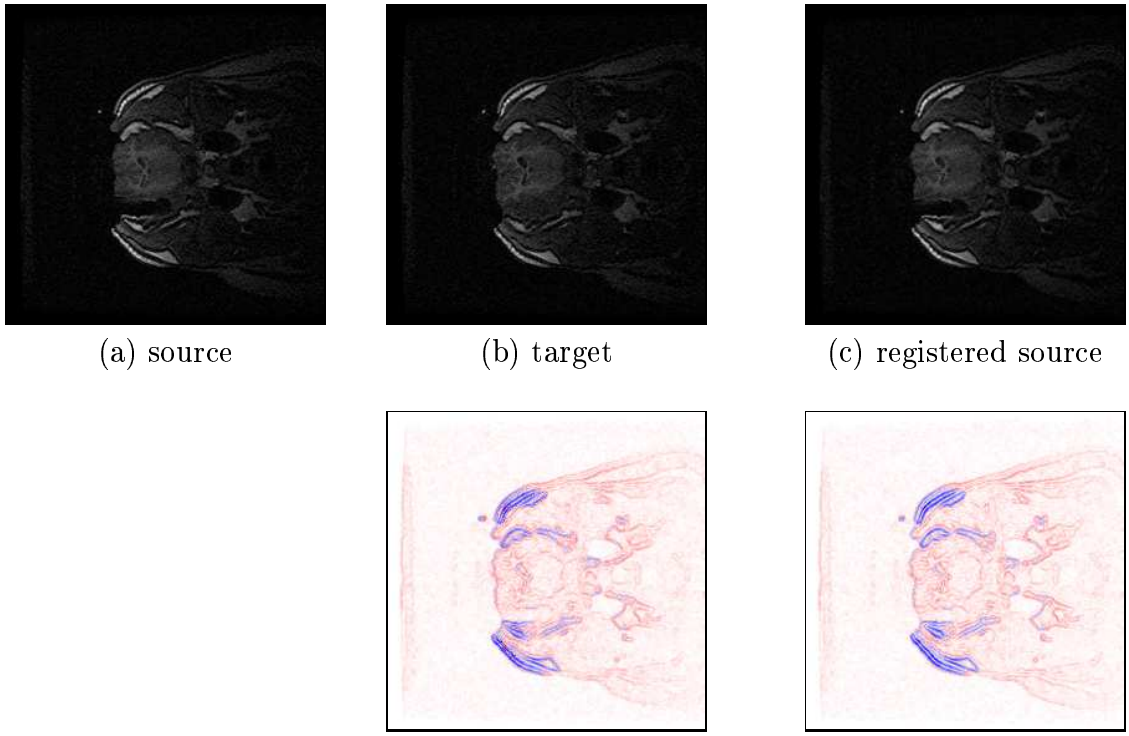


(g) estimated brightness



(h) estimated geometric map

Figure 3.46: EM Clinical result: MRI axials, stripped skull.

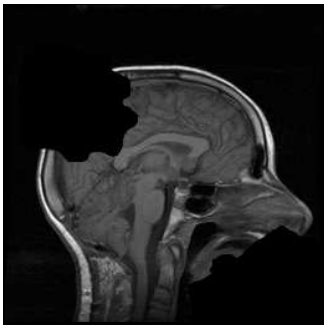


(a) source (b) target (c) registered source

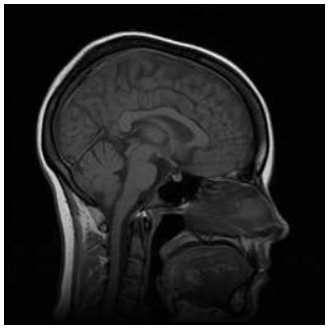
(d) error before registration (e) error after registration

(f) estimated contrast (g) estimated brightness (h) estimated geometric map

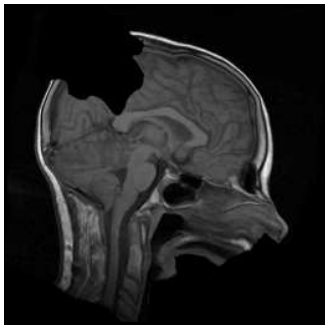
Figure 3.47: EM Clinical result: MRI coil with pig experiment.



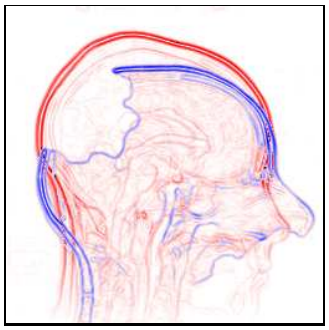
(a) source



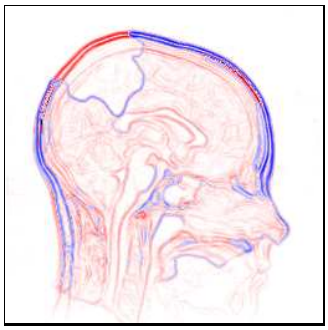
(b) target



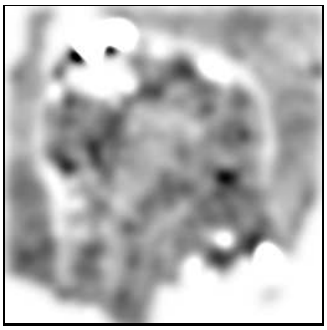
(c) registered source



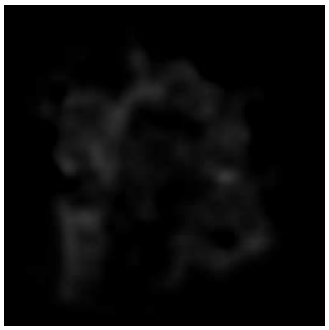
(d) error before registration



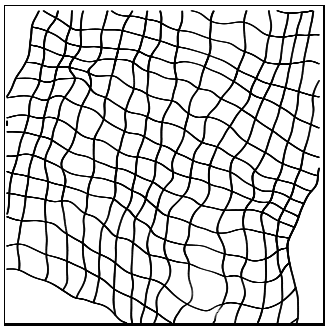
(e) error after registration



(f) estimated contrast

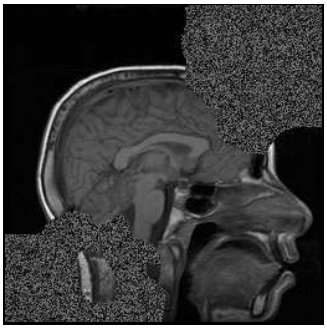


(g) estimated brightness

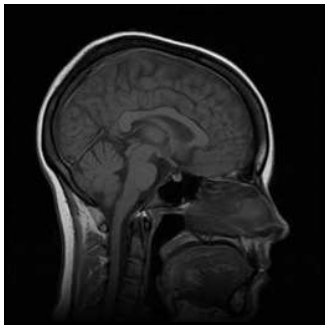


(h) estimated geometric map

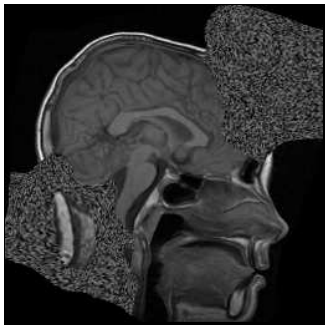
Figure 3.48: EM Clinical result: MRI sagitals, missing data.



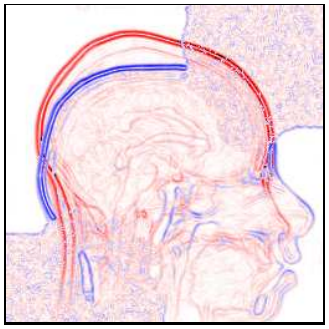
(a) source



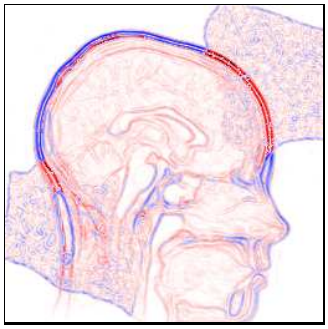
(b) target



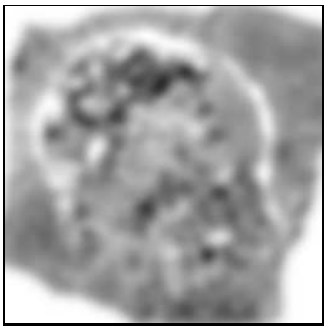
(c) registered source



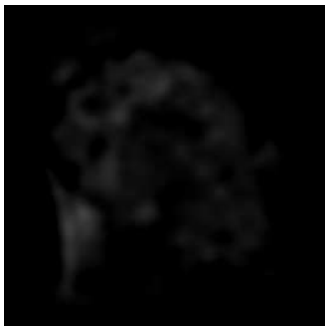
(d) error before registration



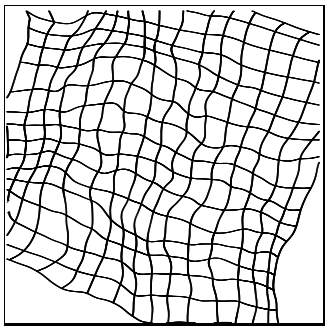
(e) error after registration



(f) estimated contrast

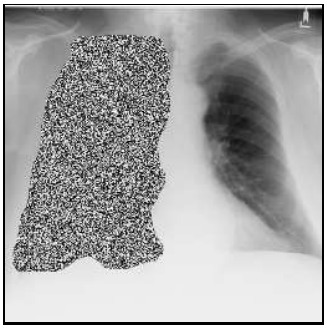


(g) estimated brightness



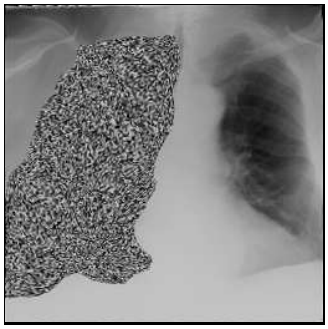
(h) estimated geometric map

Figure 3.49: EM Clinical result: MRI sagitals, with noise.

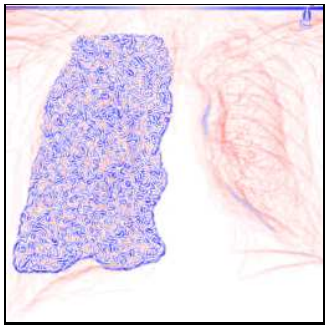


(a) source

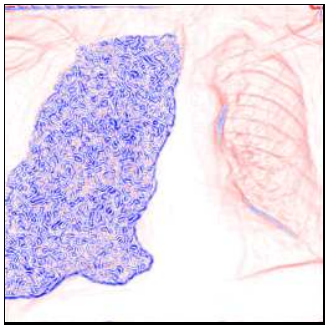
(b) target



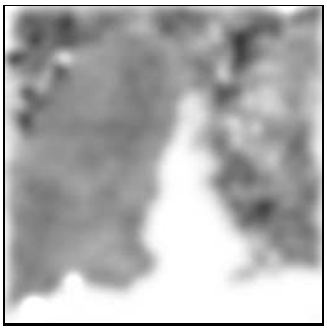
(c) registered source



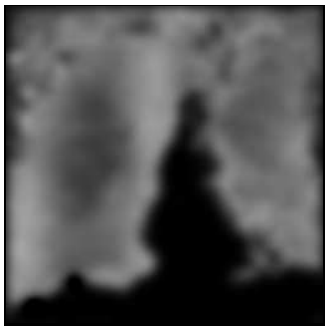
(d) error before registration



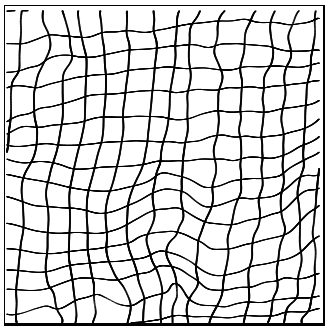
(e) error after registration



(f) estimated contrast

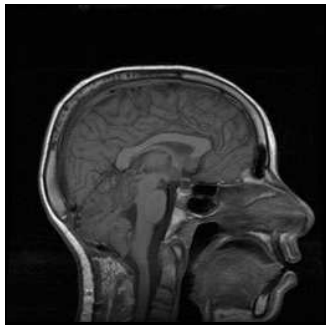


(g) estimated brightness

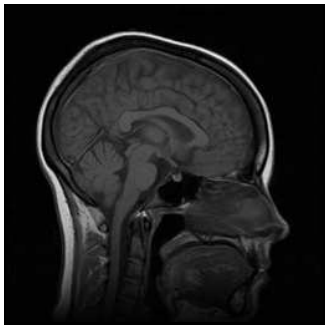


(h) estimated geometric map

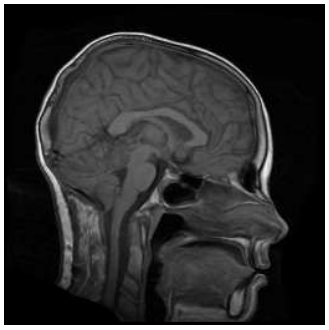
Figure 3.50: EM Clinical result: Chest X-rays, with noise.



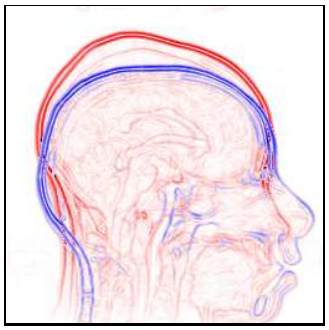
(a) source



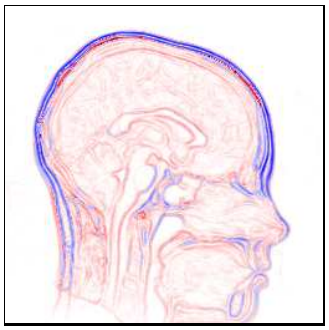
(b) target



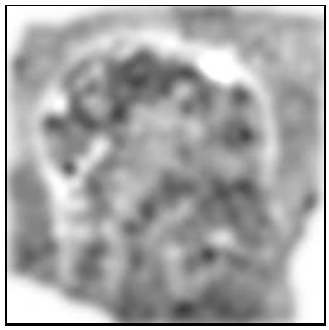
(c) registered source



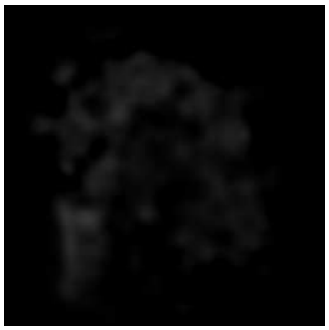
(d) error before registration



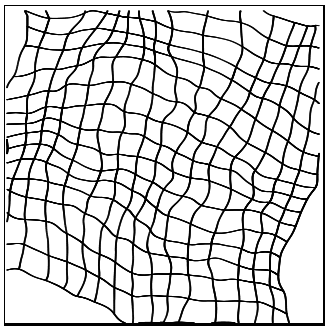
(e) error after registration



(f) estimated contrast

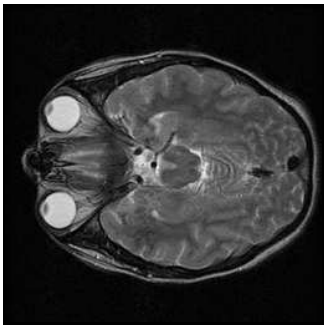


(g) estimated brightness

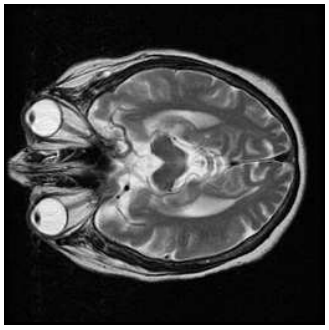


(h) estimated geometric map

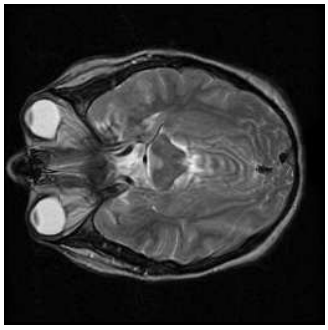
Figure 3.51: EM Clinical result: MRI sagitals, different subjects.



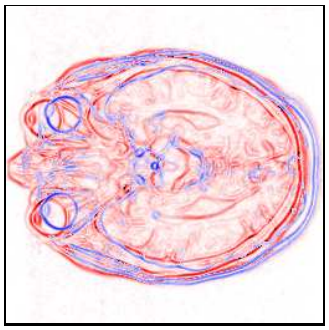
(a) source



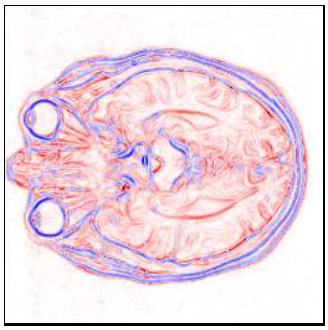
(b) target



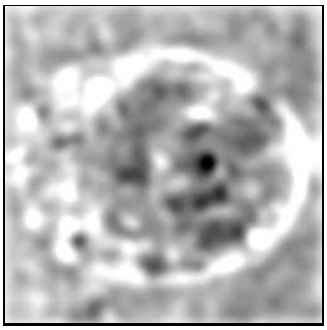
(c) registered source



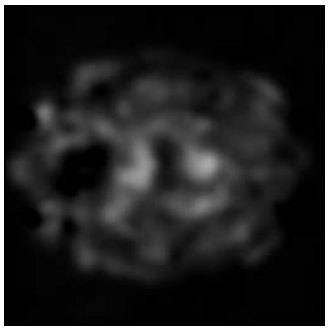
(d) error before registration



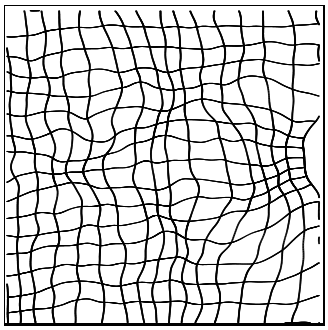
(e) error after registration



(f) estimated contrast



(g) estimated brightness



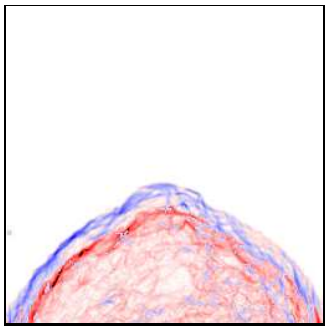
(h) estimated geometric map

Figure 3.52: EM Clinical result: MRI axials, different subjects.

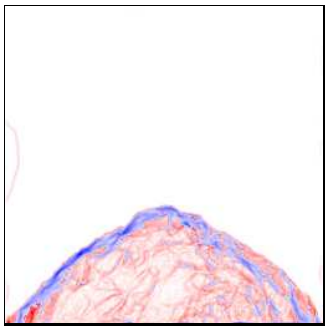
(a) source

(b) target

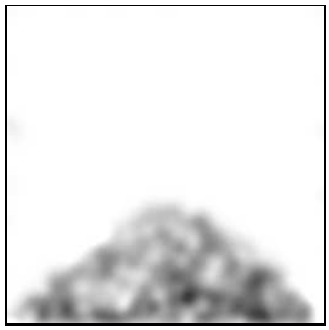
(c) registered source



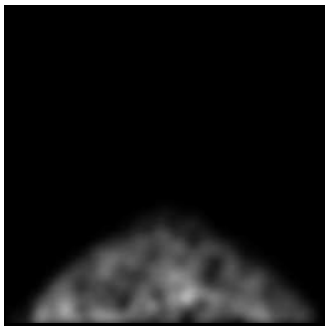
(d) error before registration



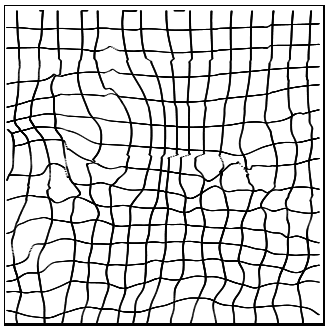
(e) error after registration



(f) estimated contrast



(g) estimated brightness



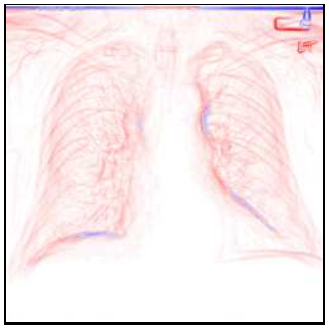
(h) estimated geometric map

Figure 3.53: EM Clinical result: Mammograms.

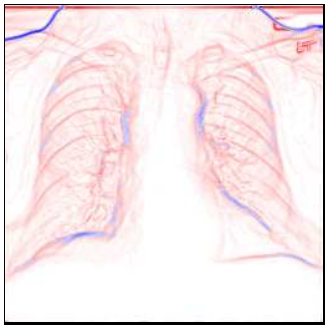
(a) source

(b) target

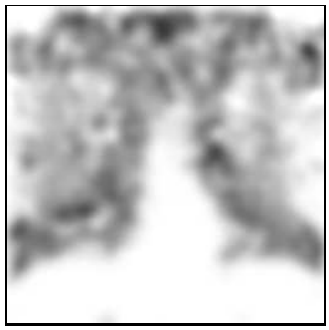
(c) registered source



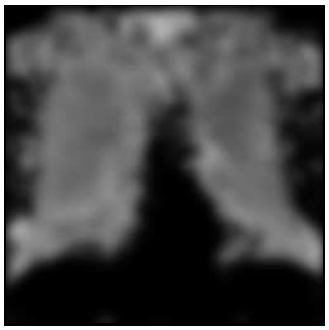
(d) error before registration



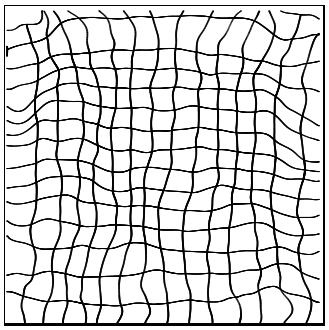
(e) error after registration



(f) estimated contrast



(g) estimated brightness



(h) estimated geometric map

Figure 3.54: EM Clinical result: Chest X-rays.

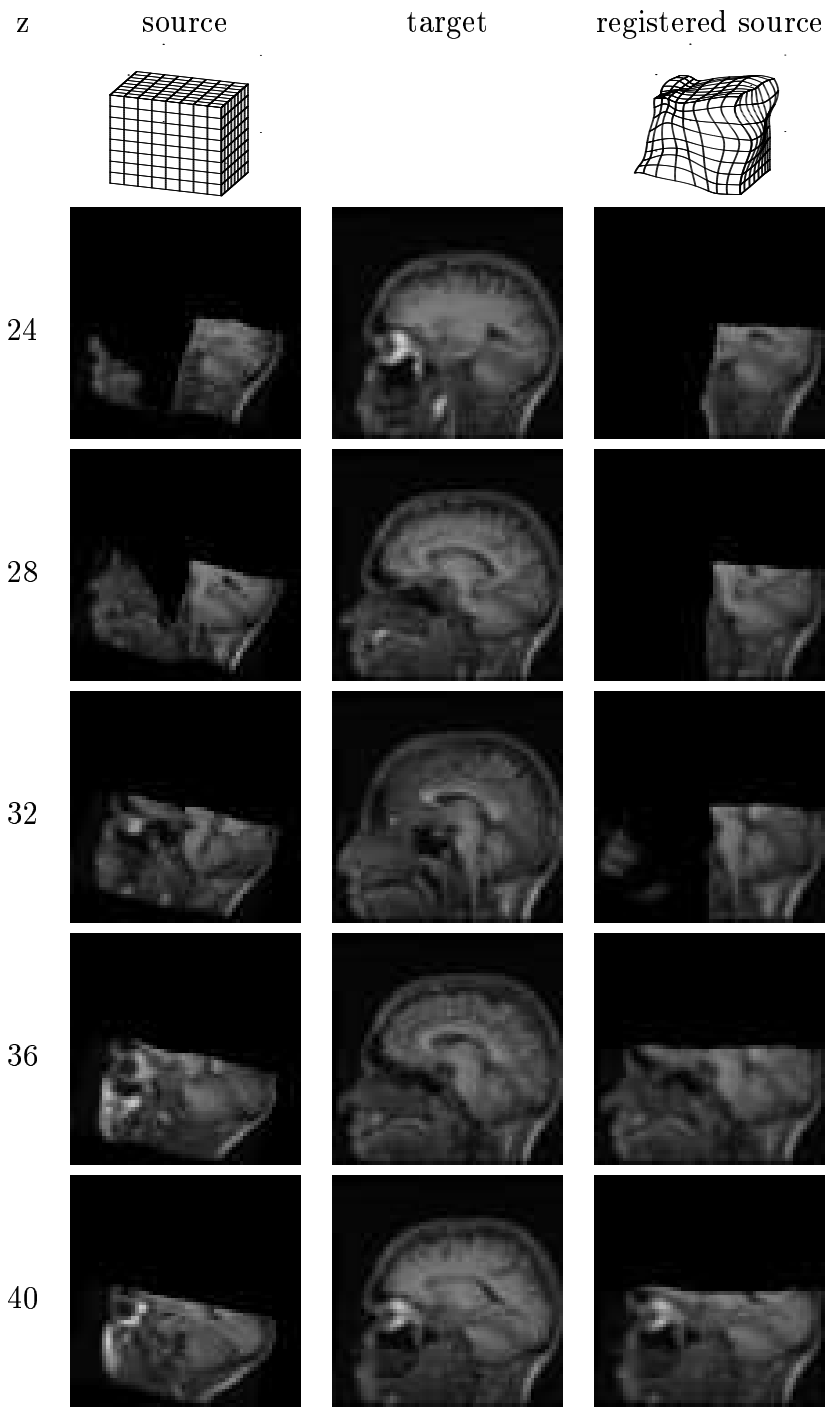


Figure 3.55: 3-D EM result: MR partial brain (1). See also Figure 3.59.

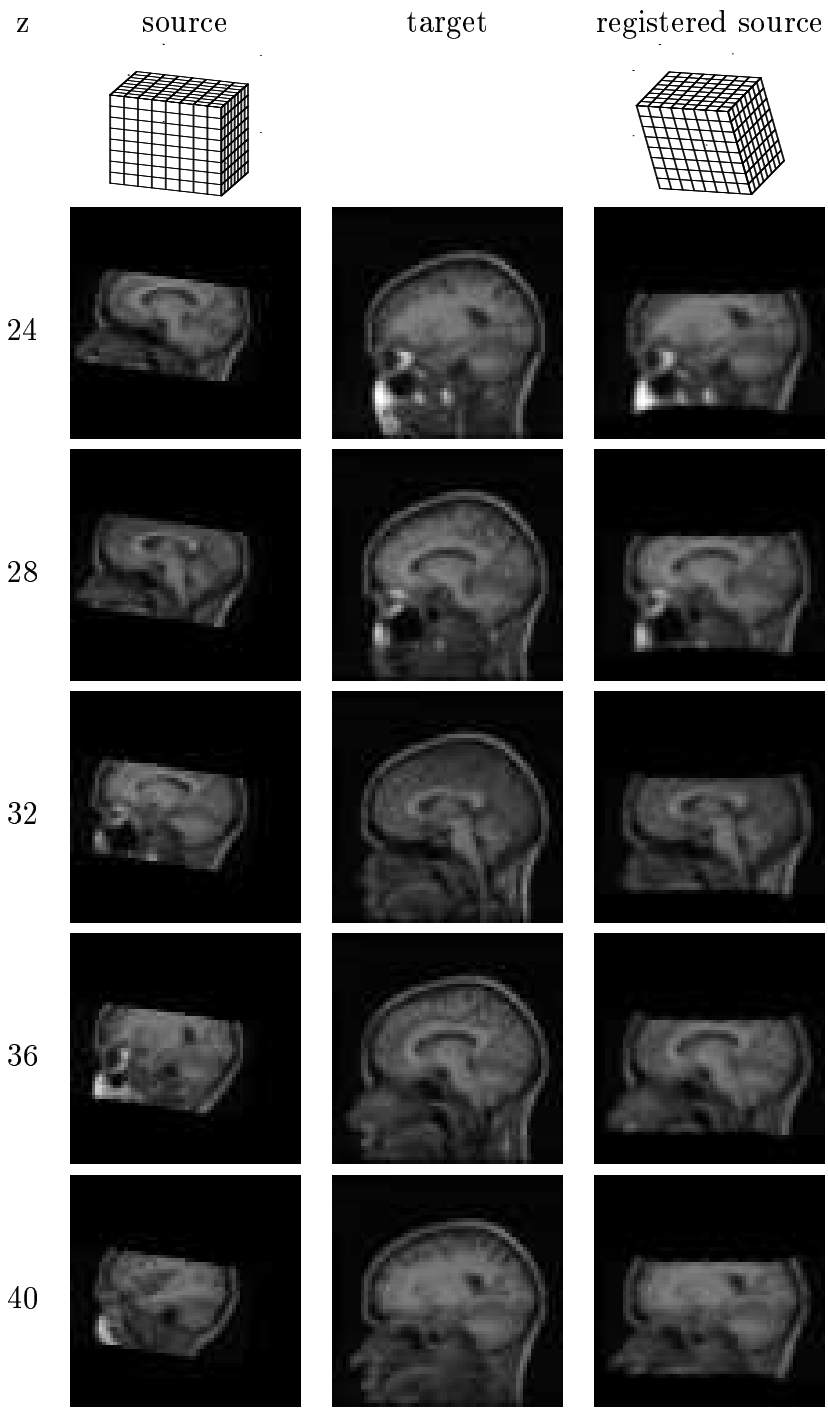


Figure 3.56: 3-D EM result: MR partial brain (2). See also Figure 3.59.

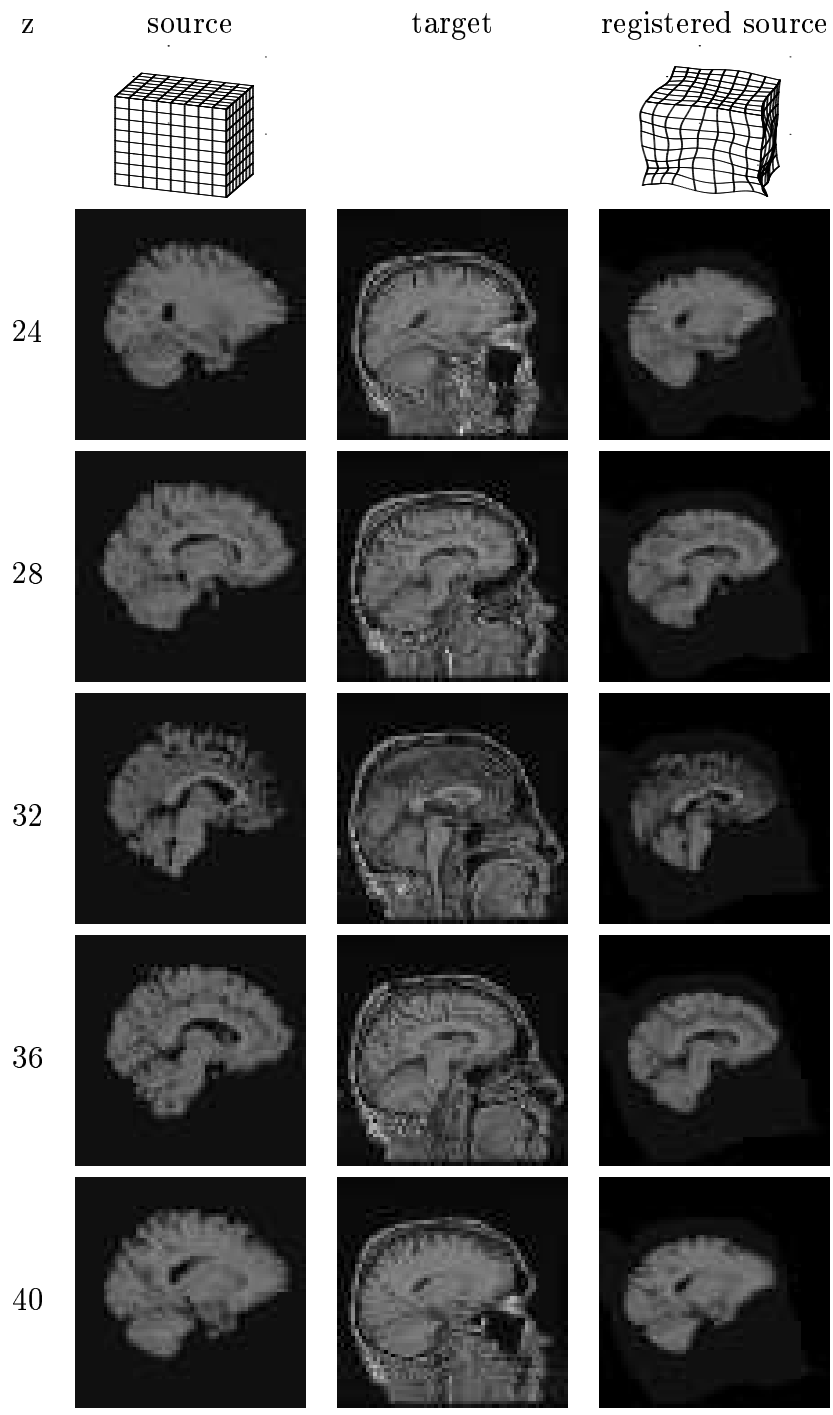


Figure 3.57: 3-D EM result: MR segmented brain. See also Figure 3.59.

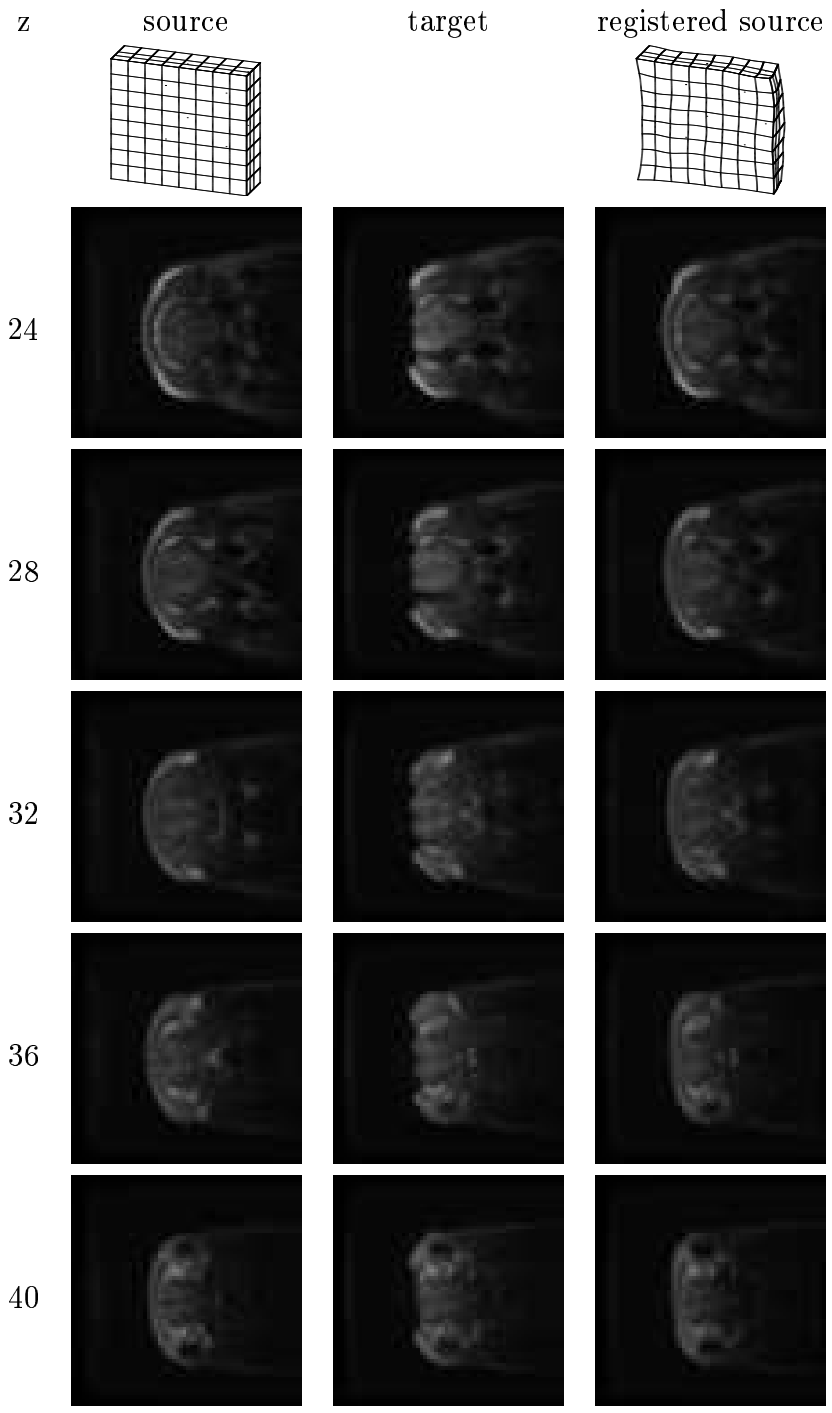


Figure 3.58: 3-D EM result: MR coil, pig experiment. See also Figure 3.59.

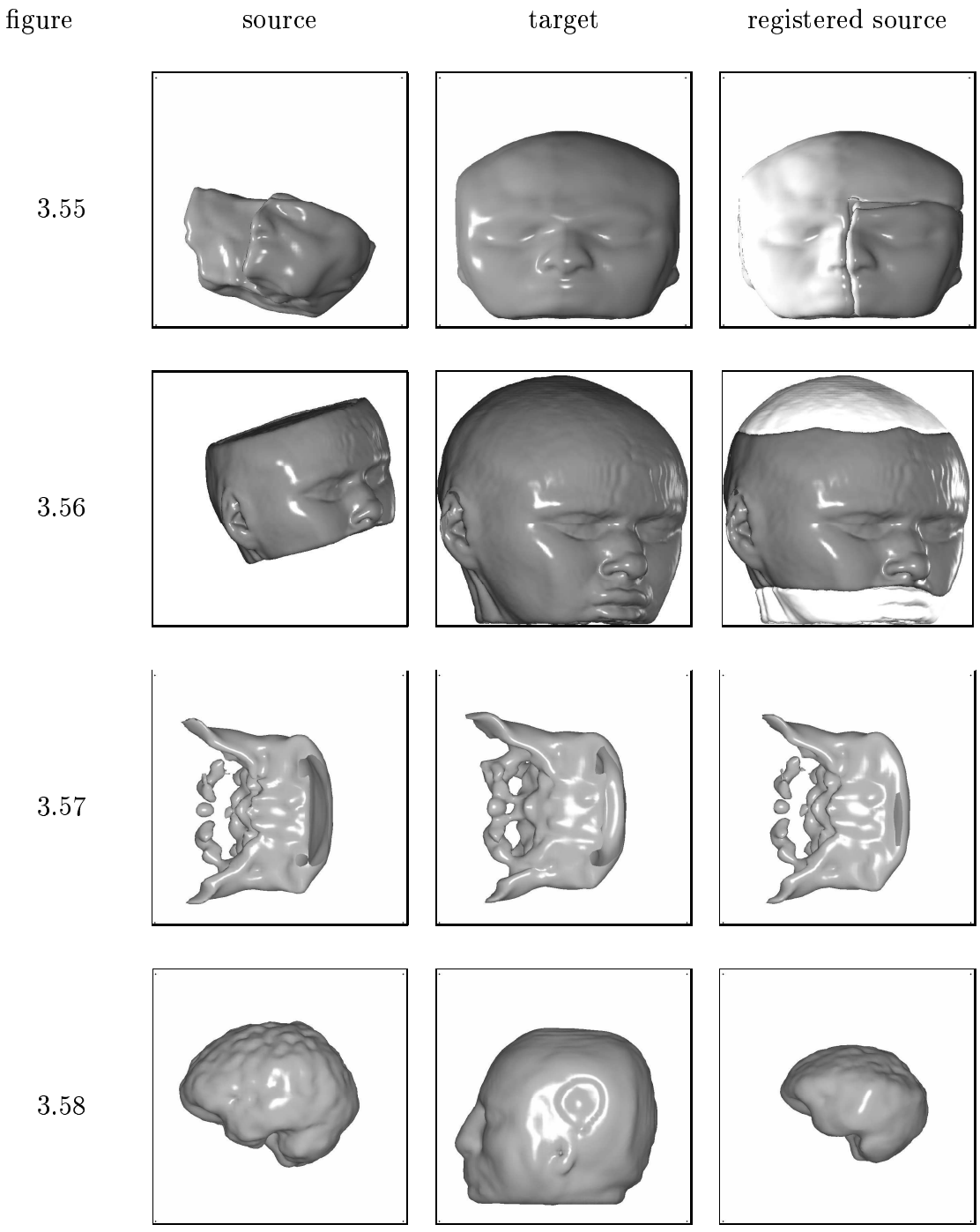
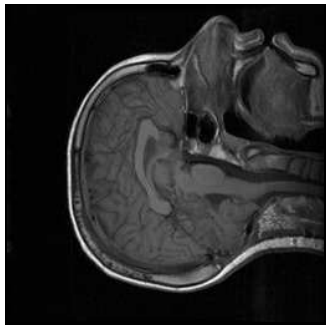
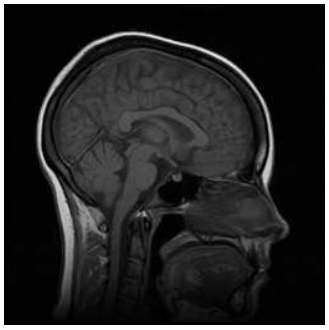


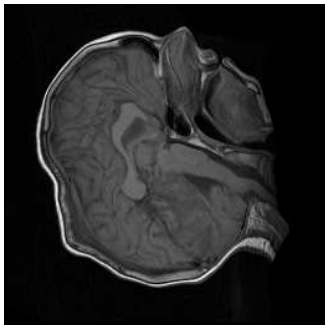
Figure 3.59: 3-D EM results (iso-surface view). Figure numbers in the first column refer to the corresponding slice views. In the first and second rows, the brighter regions shown with the registered source are the portions of the target that are missing in the source - these regions are superimposed to show the accuracy of the registration.



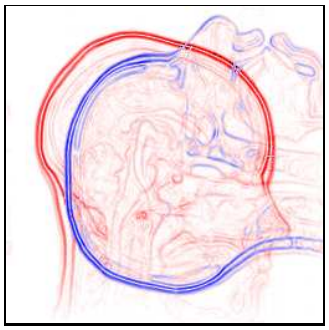
(a) source



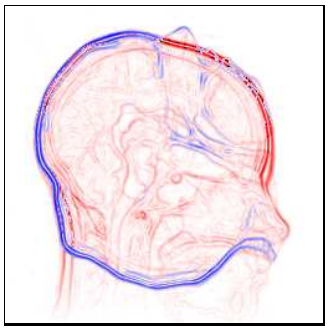
(b) target



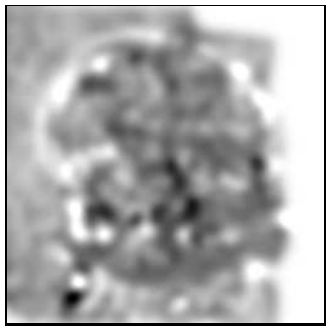
(c) registered source



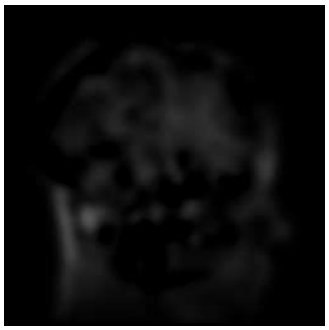
(d) error before registration



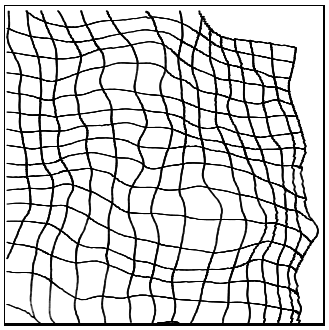
(e) error after registration



(f) estimated contrast

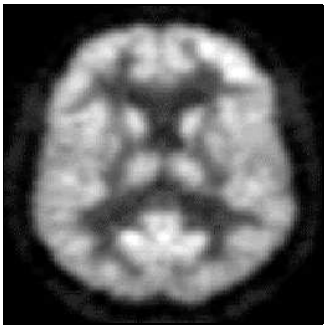


(g) estimated brightness

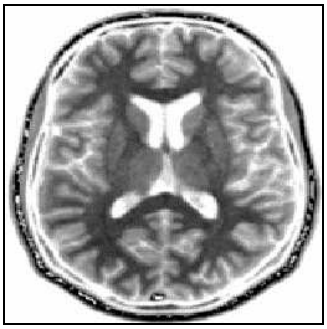


(h) estimated geometric map

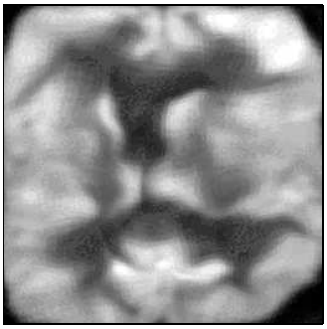
Figure 3.60: Failure 1: MRI sagitals, large 90 degree rotation



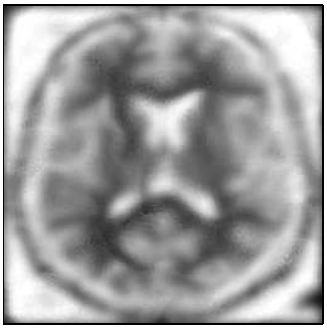
(a) source



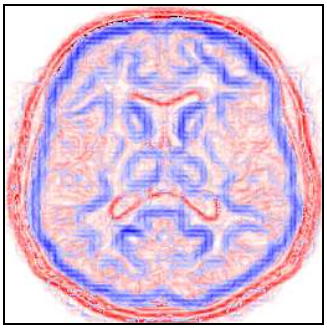
(b) target



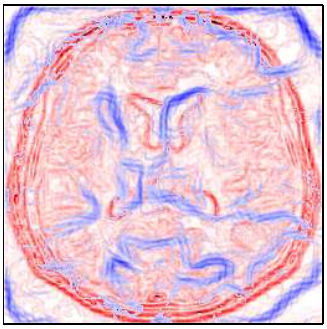
(c) registered source



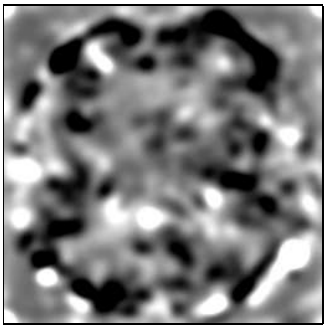
(d) registered source, with b/c



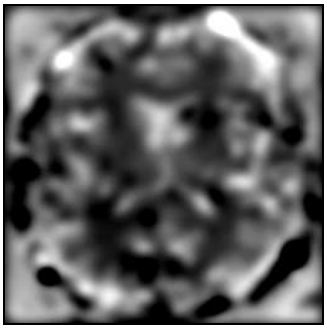
(e) error before registration



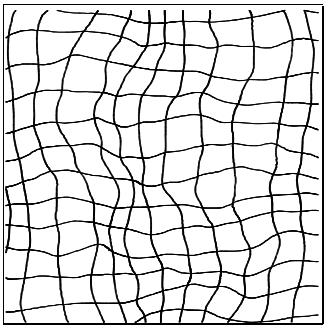
(f) error after registration



(g) estimated contrast

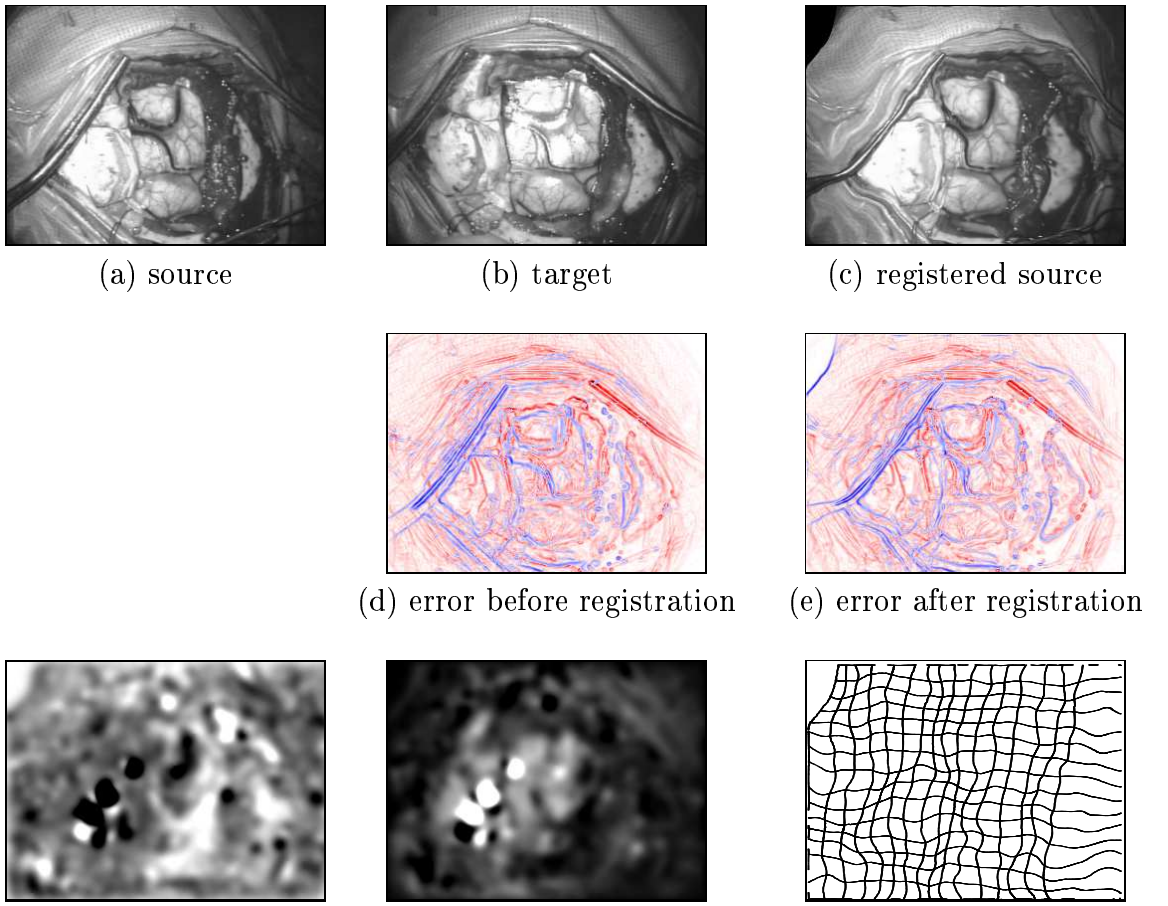


(h) estimated brightness



(i) estimated geometric map

Figure 3.61: Failure 2: PET and MRI. Shown in panel (d) is the result of applying the estimated contrast (panel (g)) and brightness (panel (h)) maps to the registered source in panel (c).



(a) source (b) target (c) registered source

(d) error before registration (e) error after registration

(f) estimated contrast (g) estimated brightness (h) estimated geometric map

Figure 3.62: Failure 3: Photos of brain during operation

Chapter 4

Simulations and analysis

4.1 Introduction

The previous section illustrated results of the registration algorithm on many synthetic and clinical images, demonstrating the efficacy of the algorithm. This section contains a sensitivity analysis of the algorithm with respect to the design assumptions. The extent to which distortions can be recovered is analyzed and quantified, along with a study of the various trade-offs that occur during the registration. This analysis will help to better understand the system as a whole, especially its limitations. We begin with a description of how the synthetic images and registration maps used in all the experiments are generated.

4.2 Synthetic random data

Synthetic images and registration maps are used for all the simulations in the analysis. While these images/maps capture only certain aspects of medical images, they allow for a large testbed, and avoid biasing the results towards specific images/maps.

Synthetic images

Consistent with a simple model for natural images [54], the images synthesized for the simulations are fractal in nature and modeled with a $\left(\frac{1}{\vec{\omega}}\right)^{\alpha}$ power spectrum and random phase. The value of α is determined as follows. For a set of sample images, the 2-D power spectrum is first computed. For each discrete frequency ω in the 2-D power spectrum, the median value is computed, thus reducing the 2-D spectrum to 1-D. This 1-D spectrum is then modeled as $\left(\frac{1}{\vec{\omega}}\right)^{\alpha}$, and α is estimated using a simple least-squares estimation. Note that in this process, directional information is lost (since we use the median value in each frequency), along with the phase correlations (since the model uses only the power spectrum). The estimated value of α is 1.4, averaged over a set of ten clinical images (MRI, chest X-rays, mammograms, and CT).

Given the value of α, each synthetic image $s(x,y)$ is generated in the Fourier domain as follows. Let $\Im\{\cdot\}$ denote the Fourier operator, and $\Im^{-1}\{\cdot\}$ the inverse Fourier operator. Then:

$$s(x,y) = \Im^{-1}\left\{\Im\left\{r(x,y)\right\} \cdot H(\omega_x, \omega_y)\right\}, \tag{4.1}$$

where $r(x,y)$ is a random image of size 128×128, with pixel values chosen from a normal distribution with zero mean and unit variance. The response $H(\omega_x, \omega_y)$ corresponds to the model of the desired power spectrum in 2-D, given by:

$$H(\omega_x, \omega_y) = \left(\frac{1}{\sqrt{\omega_x^2 + \omega_y^2}}\right)^{\alpha}. \tag{4.2}$$

The intensity values of the image $s(x,y)$ are scaled into the range $[0,1]$. A sample synthetic image is shown in Figure 4.1(a).

Synthetic registration map

A registration map consists of a displacement vector $\vec{v}(x,y) = (v_x(x,y), v_y(x,y))$ at each pixel location (x,y). The displacement fields $v_x(\cdot)$ and $v_y(\cdot)$ are modeled independently as two Gaussian distributions, and are synthesized independently as follows. All parameter values used were estimated from known registration maps, as in the previous section. First, two random images of size 32×32 are generated, with pixel values drawn from a normal distribution with zero mean and a standard deviation of 5.0 pixels. Each image is then up-sampled to a size of 128×128, after which a 3×3 low-pass filter with coefficients $[1\,2\,1; 2\,4\,2; 1\,2\,1]/16$ is repeatedly applied, until a smoothness value of 0.3 is obtained. Smoothness is computed as follows:

$$\left(\frac{1}{N^2}\right)\left(\sqrt{\sum_{x,y\in\Omega}\left(\left(\frac{\partial v_x(x,y)}{\partial x}\right)^2+\left(\frac{\partial v_x(x,y)}{\partial y}\right)^2\right)}+\sqrt{\sum_{x,y\in\Omega}\left(\left(\frac{\partial v_y(x,y)}{\partial x}\right)^2+\left(\frac{\partial v_y(x,y)}{\partial y}\right)^2\right)}\right), \tag{4.3}$$

where Ω is over the entire image, and N is the total number of pixels. Finally, a global affine $[m_1\ m_2;\ m_3\ m_4]$ and translation $[m_5;\ m_6]$ map is applied to each of the images. The affine parameters vary with a uniform distribution; parameters m_1 and m_4 (the scale parameters) vary from 0.9 to 1.1, parameters m_2 and m_3 (the shear parameters) vary from 0 to 0.2, and parameters m_5 and m_6 (the translation parameters) vary from -10 to $+10$. A sample synthetic registration map is shown in Figure 4.1(b).

4.3 Simulations

The simulations are designed to determine the range of geometric and intensity distortions that can be recovered, as well as analyze the algorithm with respect to the design assumptions. Geometric distortions include both global distortions (translation, rotation and scale changes), as well as local distortions. Intensity distortions include variations in contrast, brightness, and noise levels. All these distortions (ex-

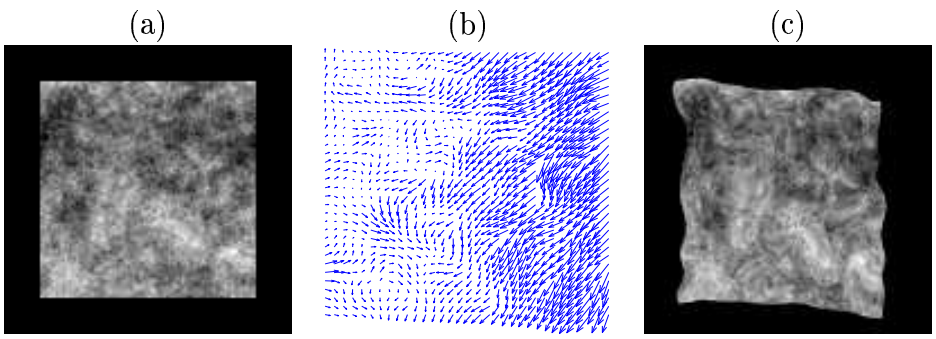


Figure 4.1: (a) Sample random synthetic image; (b) sample random synthetic registration map ; (c) result of applying the synthetic registration map to the synthetic image.

cluding noise) are assumed by the algorithm to vary smoothly. The effects of differing resolutions are also analyzed. Since it is impractical to study the combined effect of all these distortions, each distortion is analyzed independently.

4.3.1 Global geometric distortions

The ability of differential techniques to handle large distortions is generally a concern. Therefore, in the first set of simulations, the extent to which global translations, rotations and scale changes can be recovered is analyzed. In these simulations, the target image consists of a random synthetic image, while the source image consists of the target image with a random global distortion applied.

In the first simulation, the global distortions consists of only translations, varied from 0 to 30 pixels. Sample source and target images are shown in Figure 4.3. Shown in Figure 4.2 are the errors in both intensity and motion, where the dotted lines depict the errors before registration, and the solid lines depict the errors after registration. Each data point is the result of averaging over 100 independent trials. Note that the algorithm is successful up to a translation of 24 pixels, and then fails abruptly. The sudden failure is due to the finite-length derivative filters used in estimating

the spatio-temporal derivatives. These filters are of size 3×3, limiting the range of motion that can be found to 3 pixels. However, the multiscale implementation of the algorithm allows us to capture a larger range, essentially doubling the range at each scale. Thus, a three-level pyramid would allow us to capture a range of 12 pixels. Another important factor that helps improve the range (as well as accuracy) is the Taylor-series iterations. Recall that the registration is performed iteratively at each level of the pyramid, and at each iteration the source is warped according to the previously estimated registration map, bringing the source closer to the target. These two factors contribute together to allow us to capture a range of 24 pixels.

In the second set of simulations, the global distortions consists of only rotations, applied from 0 to 90 degrees. Sample source and target images are shown in Figure 4.5. Shown in Figure 4.4 are the errors in both intensity and motion, where the dotted lines depict the errors before registration, and the solid lines depict the errors after registration. Each data point is the result of averaging over 100 independent trials. Note that there is first a jump in errors from 0 to 5 degrees. This is due to effects of interpolation (this is not an issue with translations because the integer translations do not require interpolation). The errors remain constant until about 45 degrees, after which the algorithm fails, and the errors jump. At this rotation, the average translation is 24 pixels, which is also the limit in the global translation simulation. Beyond 45 degrees, the errors are large, but remain fairly constant. The differential filters fail to find the correct rotation at this point, and instead simply find the closest rotation to 0 degrees, maximizing the overlap between the images. Thus, beyond 45 degrees, the error values are equivalent to the errors between two different random fractal images.

In the final set of global simulations, the global distortions consists of only scale changes, varying from a factor of 1.0 to 2.0. Sample source and target images are

shown in Figure 4.7. Shown in Figure 4.6 are the errors in both intensity and motion, where the dotted lines depict the errors before registration, and the solid lines depict the errors after registration. Each data point is the result of averaging over 100 independent trials. Note that there is once again first a jump in errors from a scale change of 1.0 to 1.1, caused by interpolation effects. The errors then increase gradually up to a scale change of 1.6, after which the algorithm fails completely. This behavior is different from the behavior of the rotation simulation, in which the errors were constant for a range of rotations before increasing. At a scale change of 1.6, the computed average translation is 19 pixels, with a maximum translation of 38 pixels. From the simulation with global translations, we know that an average translation of 24 pixels can be accommodated, so it reasonable for this scale change to work. At a scale change of 1.7, the computed average translation is 22 pixels, with a maximum translation (at the boundaries) of 44 pixels. In this case, the large maximum translations begins to dominate the results, causing the algorithm to fail.

4.3.2 Smoothness of local geometric distortions

In this simulation, the ability of the registration algorithm to recover local geometric distortions of varying smoothness is analyzed. The target image consists of a synthetic fractal image, while the source image consists of the target image, with a known synthetic geometric distortion applied to it. The smoothness of this distortion field is varied from 0.1 to 0.6, rounded to one decimal place, and is generated similar to the synthetic registration map, as described in Section 4.2; the only two differences being that 1) there is no global distortion applied to the registration map, and 2) the smoothness of the registration map is varied. Sample target images with the local geometric maps applied are shown in Figure 4.9. Shown in Figure 4.8 are the errors in

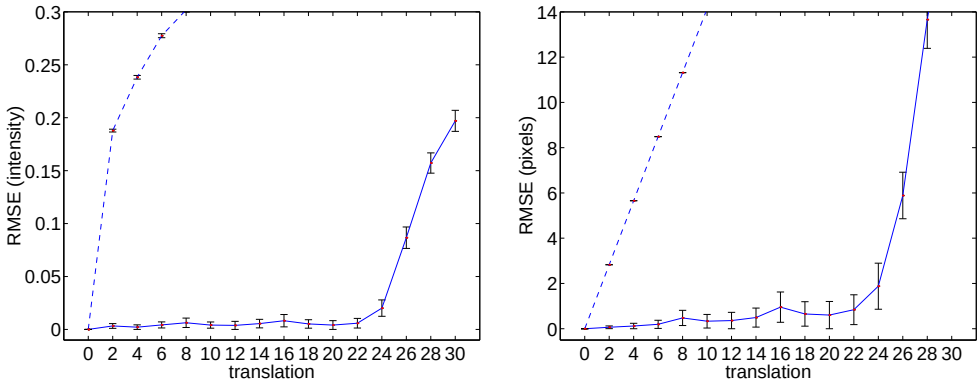


Figure 4.2: Global translation vs. registration error. Shown are errors in intensity (left) and motion (right). The dotted lines depict the errors before registration, and the solid lines depict the errors after registration. See also Figure 4.3.

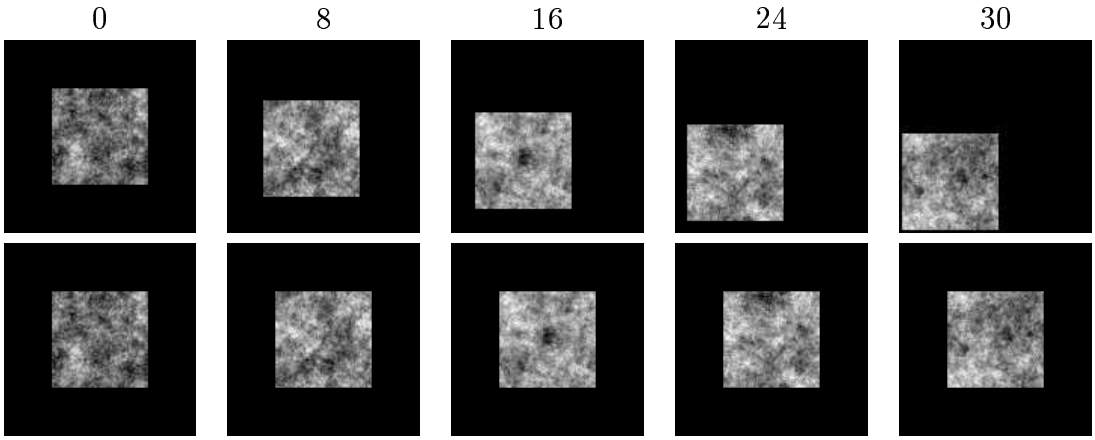


Figure 4.3: Sample source and target images, with a global translation applied. The top label is the amount of translation. The first and second rows show the source and target images, respectively. See also Figure 4.2.

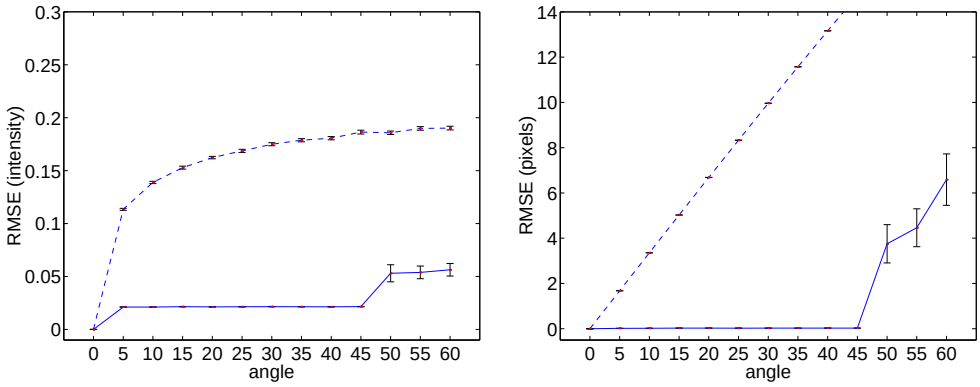


Figure 4.4: Global rotation vs. registration error. Shown are errors in intensity (left) and motion (right). The dotted lines depict the errors before registration, and the solid lines depict the errors after registration. See also Figure 4.5.

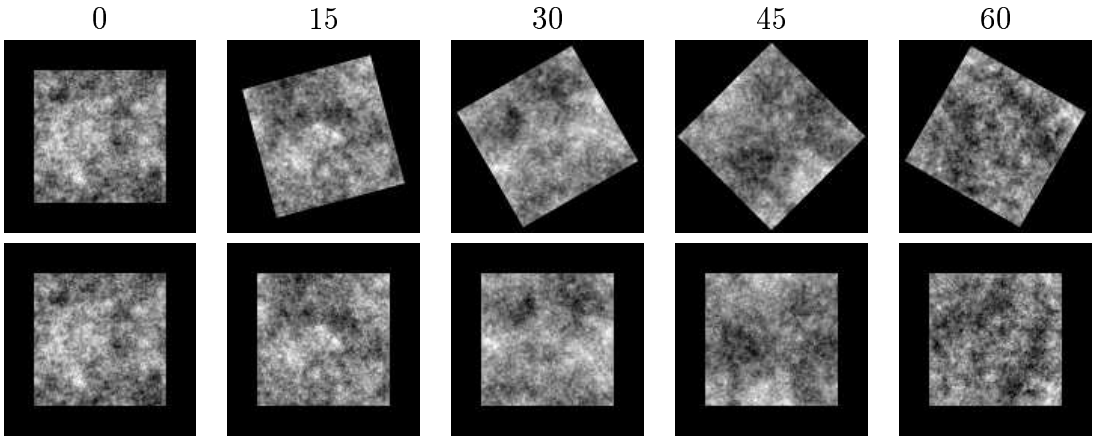


Figure 4.5: Sample source and target images, with a global rotation applied. The top label is the amount of rotation. The first and second rows show the source and target images, respectively. See also Figure 4.4.

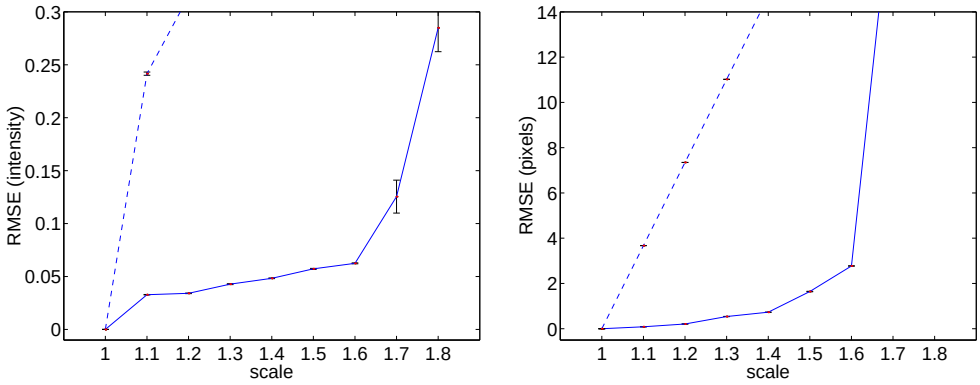


Figure 4.6: Global scale change vs. registration error. Shown are errors in intensity (left) and motion (right). The dotted lines depict the errors before registration, and the solid lines depict the errors after registration. See also Figure 4.7.

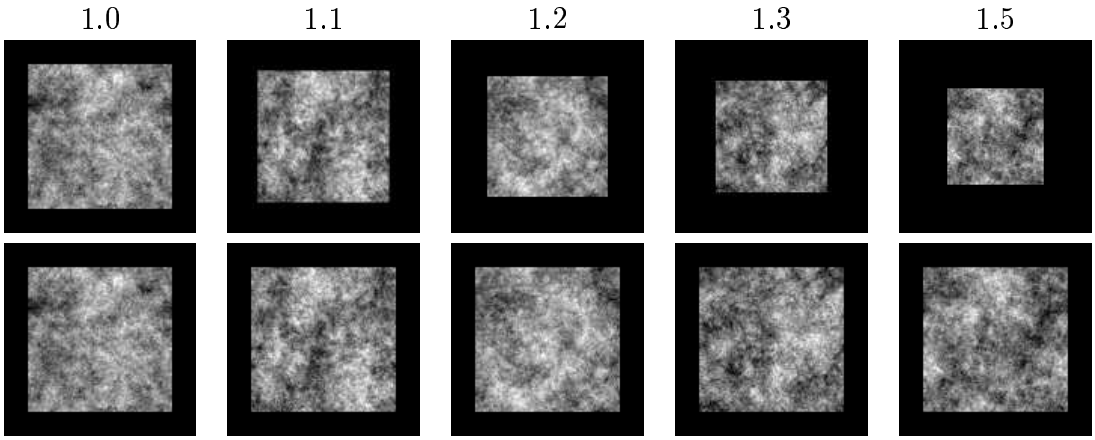


Figure 4.7: Sample source and target images, with a global scale change applied. The top label is the amount of scale change. The first and second rows show the source and target images, respectively. See also Figure 4.6.

both intensity and motion, where the dotted lines depict the errors before registration, and the solid lines depict the errors after registration. Each data point is the result of averaging over 100 independent trials. Note that a smoothness distortion of less than or equal to 0.3 corresponds to errors less than 0.05 in intensity, indicating the registration algorithm is successful at this smoothness value (and below). Note also that this is the chosen smoothness value for the synthetic registration map, estimated over known registration maps obtained using clinical images.

4.3.3 Brightness sensitivity

In this simulation, the sensitivity of the registration algorithm to brightness variations is analyzed. The source image consists of a synthetic fractal image (with intensities in the range $[0, 1]$), with a synthetic registration map applied. The target image consists of the same synthetic image used in the source, with a random fractal brightness map added to it, so that the intensities range from $[0, 1 + b]$. Examples of these source and target images for different values of b are shown in Figure 4.11. Shown in Figure 4.10 are the errors in both intensity and motion, where the dotted lines depict the errors before registration, and the solid lines depict the errors after registration. Each data point is the result of averaging over 100 independent trials. Note that the error increases with brightness as expected. Reasonable errors are observed below a brightness value of 0.5. The small errors observed in this range are largely due to interpolation.

4.3.4 Contrast sensitivity

In this simulation, the sensitivity of the registration algorithm to contrast variations is analyzed. The source image consists of a synthetic fractal image (with intensities in

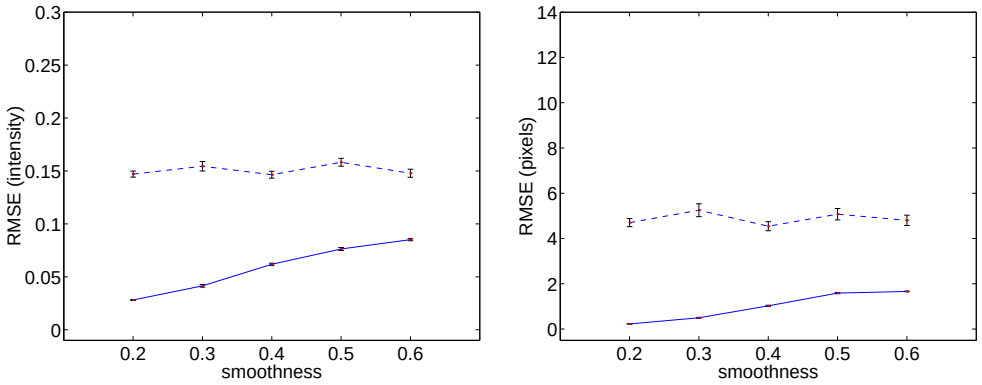


Figure 4.8: Smoothness of geometric registration map vs. registration error. Shown are errors in intensity (left) and motion (right). The dotted lines depict the errors before registration, and the solid lines depict the errors after registration. See also Figure 4.9.

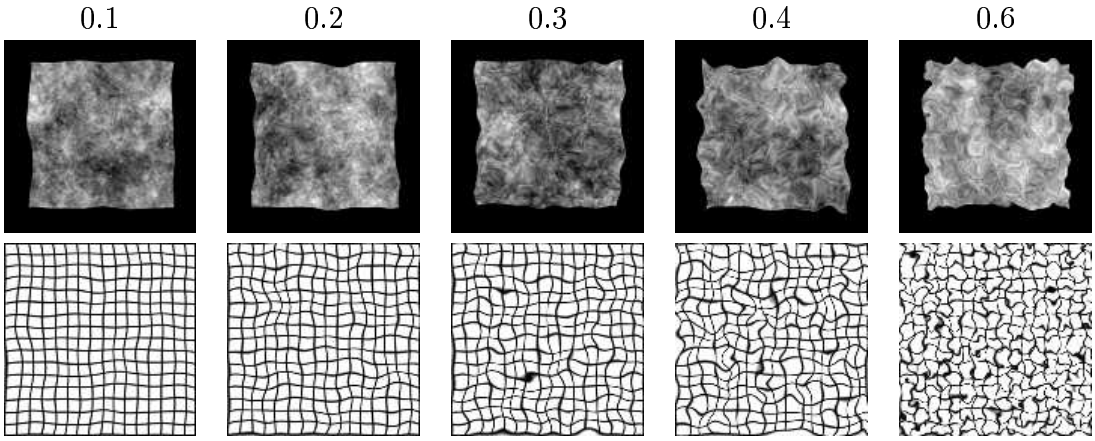


Figure 4.9: Sample source and target images, with a geometric distortion of varying smoothness applied. The top label is the smoothness value. The first and second rows show the source and target images, respectively. See also Figure 4.8.

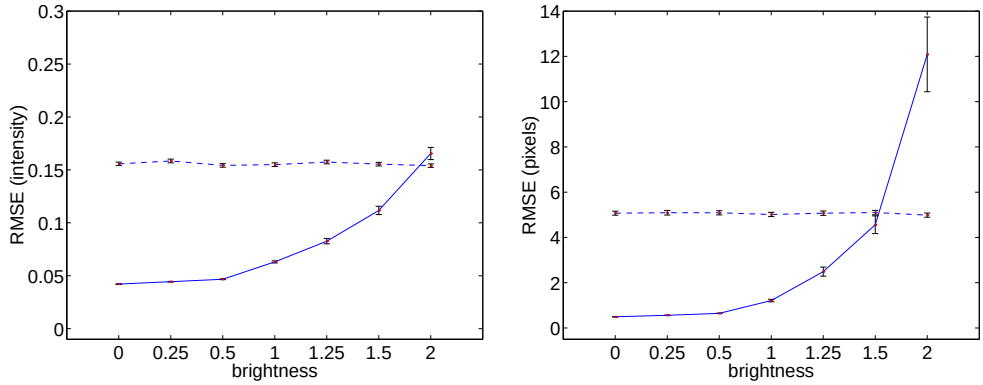


Figure 4.10: Brightness vs. registration error. Shown are errors in intensity (left) and motion (right). The dotted lines depict the errors before registration, and the solid lines depict the errors after registration. See also Figure 4.11.

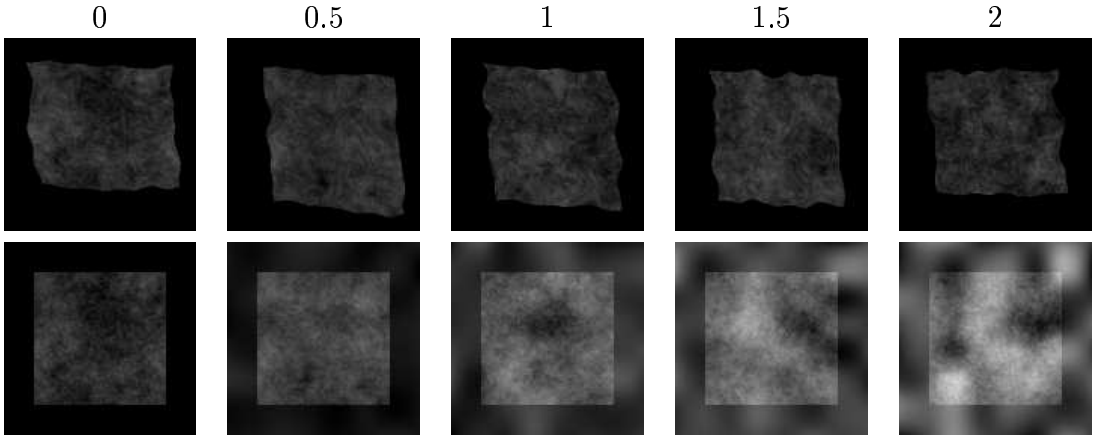


Figure 4.11: Sample source and target images, with a brightness map applied. The top label is the brightness value. The first and second rows show the source and target images, respectively. See also Figure 4.10.

the range $[0, 1]$), with a synthetic registration map applied. The target image consists of the same synthetic image used in the source, with a random fractal contrast map multiplied to it, so that the intensities range from $[c, 1]$. Examples of these source and target images for different values of c are shown in Figure 4.13. Shown in Figure 4.12 are the errors in both intensity and motion, where the dotted lines depict the errors before registration, and the solid lines depict the errors after registration. Each data point is the result of averaging over 100 independent trials. Note that the contrast value c is varied first from 0.01 to 0.1 in steps of 0.01 (up to the vertical dashed line), and then from 0.1 to 1.0 in steps of 0.1. The errors are reasonable for contrast values greater than 0.2, after which the errors become constant. The small errors observed in this range are largely due to interpolation.

4.3.5 Noise sensitivity

In this simulation, the ability of the registration algorithm to perform in the presence of additive uniform noise is analyzed. The target image consists of a synthetic fractal image, and the source image consists of the target image with uniformly distributed noise added to it. The intensity of the noise is modulated within the range $[0, n]$, where n varies from 0 (no noise) to 1.0 (100% noise). Sample source and target images for different values of n are shown in Figure 4.15. Shown in Figure 4.14 are the errors in both intensity and motion, where the dotted lines depict the errors before registration, and the dark solid lines depict the errors after registration. Each data point is the result of averaging over 100 independent trials. Note that reasonable errors are obtained for noise modulations greater than or equal to 3dB. It is hypothesized that the registration is successful in this range because the estimated contrast and brightness terms absorb the errors caused by the noise. This hypothesis was tested

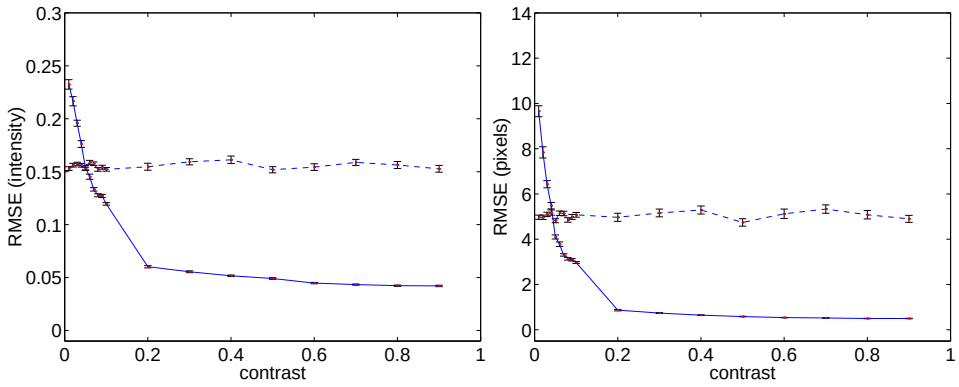


Figure 4.12: Contrast vs. registration error. Shown are errors in intensity (left) and motion (right). The dotted lines depict the errors before registration, and the solid lines depict the errors after registration. See also Figure 4.13.

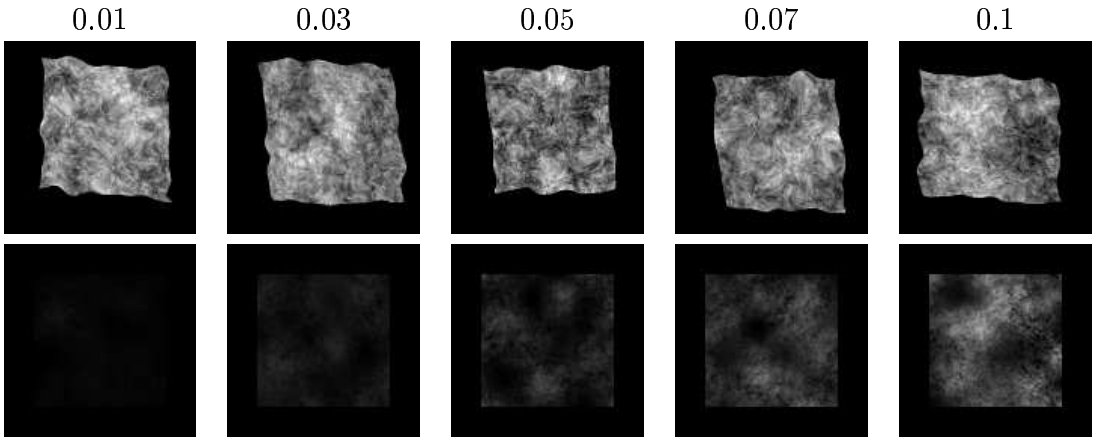


Figure 4.13: Sample source and target images, with a contrast map applied. The top label is the brightness value. The first and second rows show the source and target images, respectively. See also Figure 4.12.

by repeating the simulation, estimating only the geometric terms in the registration map. The results of this simulation are shown in Figure 4.14 using the solid lines with squares. Note that in this case, the registration is mostly unsuccessful, validating the hypothesis.

4.3.6 Resolution sensitivity

In this simulation, the ability of the registration algorithm to perform in the presence of differing resolutions is analyzed. The target image consists of a synthetic fractal image, while the source image consists of the target with a synthetic registration map applied, along with a change in resolution. The resolution of the source is varied by first applying a global scale change α, (reducing the image), followed by a global scale change of $\frac{1}{\alpha}$ (enarging the image), for α varying from 1 to 3. Sample source and target images for different values of α are shown in Figure 4.17. Shown in Figure 4.16 are the errors in both intensity and motion, where the dotted lines depict the errors before registration, and the dark solid lines depict the errors after registration. The resolution ratio is defined as the ratio of the source resolution to the target resolution. Each data point is the result of averaging over 100 independent trials. Note that errors increase linearly with respect to the resolution ratio.

4.3.7 Smoothness of contrast/brightness maps

In all the previous simulations, a particular distortion (e.g., global translation) was analyzed independent of all other distortions. In this simulation, a combination of geometric, contrast and brightness distortions is applied, over a range of different contrast/brightness smoothness values, and the results analyzed. The source image consists of a synthetic fractal image with a synthetic registration map applied to it.

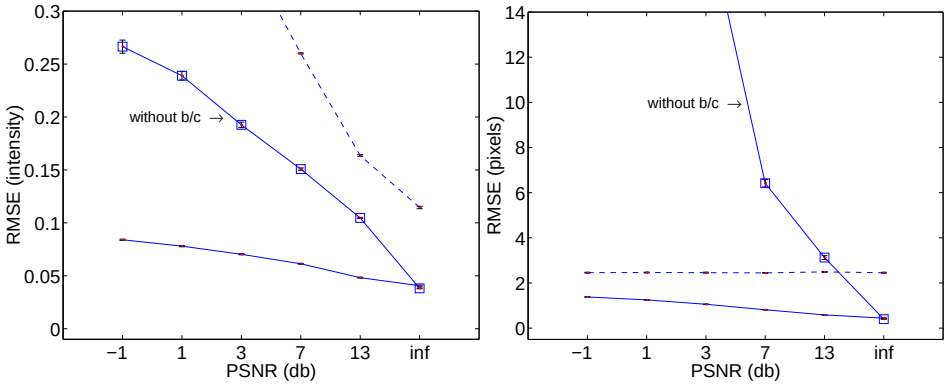


Figure 4.14: Additive uniform noise vs. registration error. Shown are errors in intensity (left) and motion (right). The dotted lines depict the errors before registration, and the solid lines depict the errors after registration. The solid line with the squares depict the errors after registration when using a geometry-only model (i.e., without estimating the brightness/contrast terms). See also Figure 4.15.

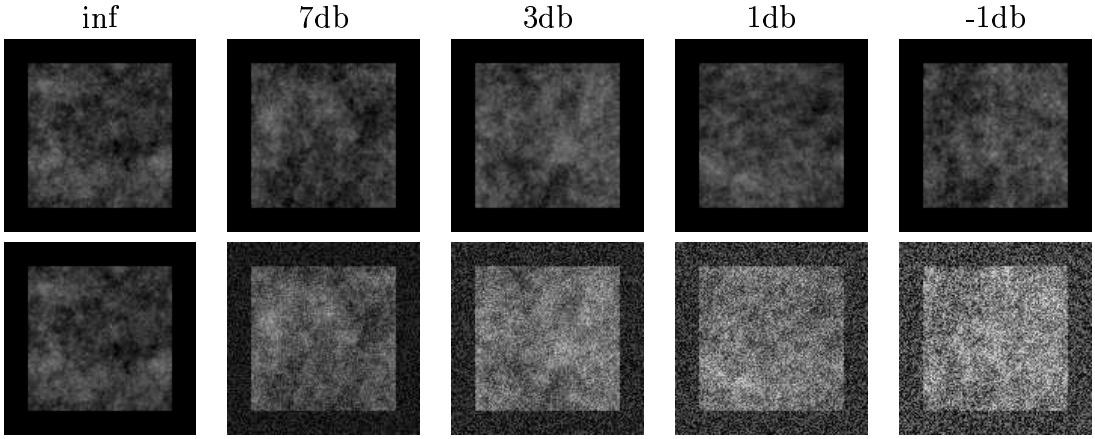


Figure 4.15: Sample source and target images, with uniform noise added. The top label is the brightness value. The first and second rows show the source and target images, respectively. See also Figure 4.14.

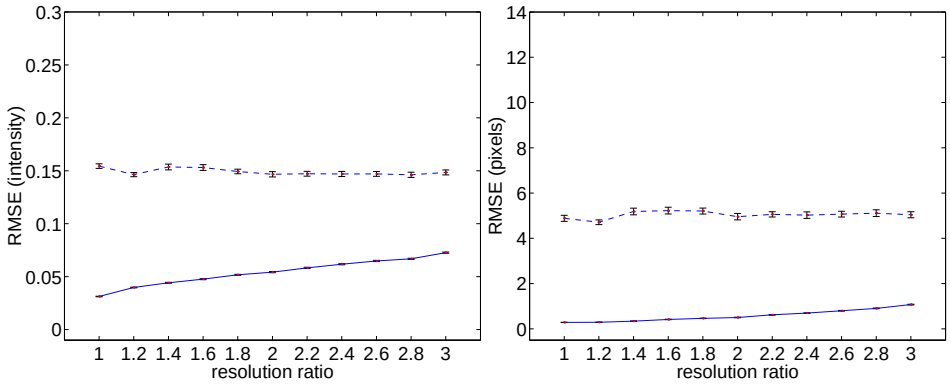


Figure 4.16: Resolution ratio vs. registration error. Shown are errors in intensity (left) and motion (right). The dotted lines depict the errors before registration, and the solid lines depict the errors after registration. See also Figure 4.17.

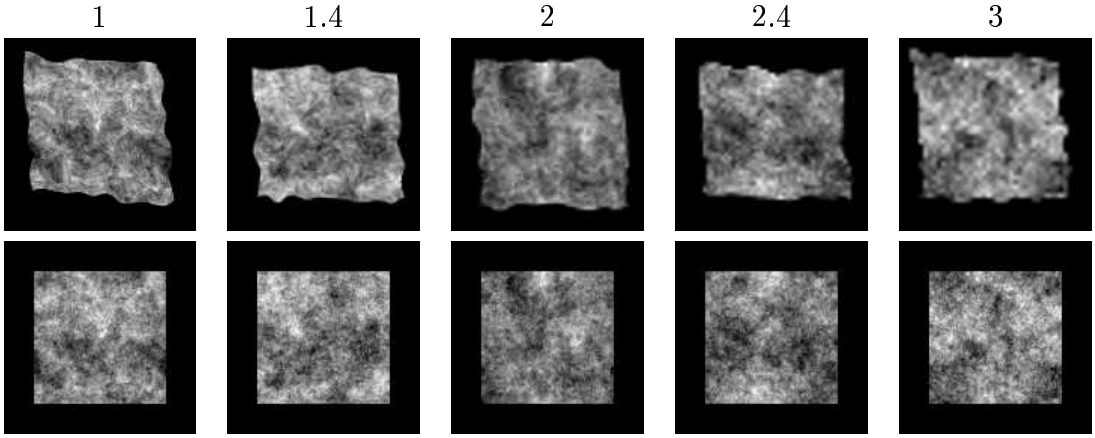


Figure 4.17: Sample source and target images, with differing resolutions. The top label is the resolution ratio (defined as the ratio of the source resolution to the target resolution). The first and second rows show the source and target images, respectively. See also Figure 4.16.

The target image consists of the the same synthetic fractal image used in the source, with a contrast and brightness distortion both applied to it. These maps are generated as described in Sections 4.3.3 and 4.3.4, with a fixed brightness of 0.2, and a fixed contrast of 0.8. The smoothness of both the maps is then varied together from 0.1 to 0.9 for each experiment in the simulation. Sample source and target images are shown in Figure 4.19. Shown in Figure 4.18 are the errors in both intensity and motion, where the dotted lines depict the errors before registration, and the solid lines depict the errors after registration. Each data point is the result of averaging over 100 independent trials. Note that the errors are relatively constant, and independent of the smoothness values of the intensity distortion. Similar results were observed for different values of the brightness and contrast modulation, indicating that the algorithm is indeed insensitive to the smoothness of the contrast/brightness modulations.

4.4 Summary

In this section, the registration algorithm was analyzed with respect to various geometric and intensity distortions. Distortions studied include global geometric distortions (translation, rotation and scale changes), local geometric distortions of varying smoothness, smoothly varying brightness and contrast modulations, as well as noise modulations. While these distortions are only a subset of possible distortions, they form a reasonable and interesting subset, and help test the design assumptions of the algorithm. In most of these simulations, a single distortion was applied and varied until the algorithm failed. In most cases, the failure is gradual (and not exponential, which is undesirable). An exception occurs when the failure is caused by the inability of the differential filters to see the motion, as in the case of large global changes. In

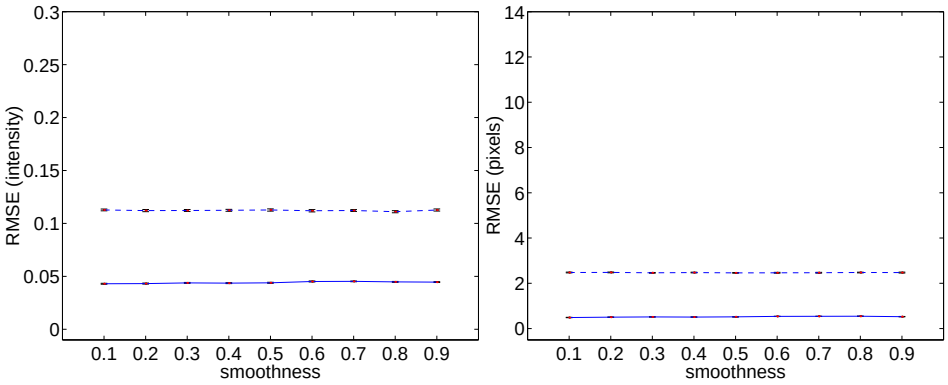


Figure 4.18: Smoothness of contrast/brightness maps vs. registration error. Shown are errors in intensity (left) and motion (right). The dotted lines depict the errors before registration, and the solid lines depict the errors after registration. See also Figure 4.19.

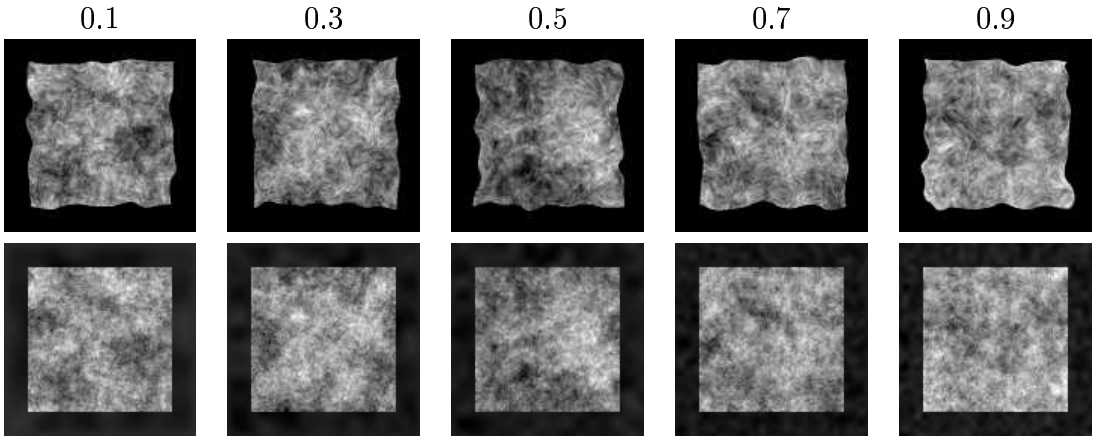


Figure 4.19: Sample source and target images, with smoothness of contrast/brightness maps varied. The top label is the smoothness value. The first and second rows show the source and target images, respectively. See also Figure 4.18.

Distortion	Range
Global translation	up to 24 pixels
Global rotation	up to 45 degrees
Global scale change	up to a factor of 1.6
Smoothness of geometric distortion	up to 0.3
Brightness sensitivity	up to 50%
Contrast modulation	greater than 50%
Additive uniform noise	greater than 4dB

Table 4.1: Ranges of various distortions which can be recovered by the algorithm.

these cases, the failures are abrupt, as expected. In the last simulation, a combination of geometric, contrast and brightness distortions was applied, and the algorithm analyzed with varying levels of contrast/brightness smoothness. It is shown that the registration remains unaffected in general to the amount of contrast/brightness smoothness. Table 4.1 shows a summary of the range of distortions which can be recovered by the algorithm.

Note that empirically, the distortions encountered while registering clinical images (results shown in Chapter 3) fall within the range of recoverable distortions, allowing for successful registration.

While the emphasis of this algorithm is on medical images, it is by no means limited to these types of images. In the next section, we explore possible applications to both medical and non-medical image registration. We also explore possible extensions to the algorithm, such as an extension to deal with differing resolutions.

Chapter 5

Conclusions

5.1 Introduction

In the preceding chapters, we described the registration algorithm in detail and demonstrated its robustness over numerous synthetic and clinical examples. In this final chapter, we briefly explore possible extensions to the algorithm, and describe some potential applications beyond medical imaging.

5.1.1 Quantifying tumor growth

Given two images of a tumor taken at different times, it is possible to quantify the change in area, thus providing valuable information about its growth. For example, consider the images shown in Figure 5.1. The source and target images (panels (a) and (b)) are CT images of a tumor, taken at two different points in time. Upon registration of the images, the change in the region of the tumor becomes apparent in the estimated geometric map (panel (e)). This map can be used to quantify the change in area as follows. Consider a 2×2 square neighborhood of pixels located in the source image at position (i, j). After registration, let this neighborhood correspond

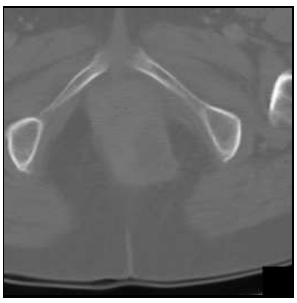
to a polygon with coordinates $(x_i, y_j), (x_{i+1}, y_j), (x_i, y_{j+1}), (x_{i+1}, y_{j+1})$ in the target image. The area of this polygon is given by the sum of the area of the two triangles formed by $(x_i, y_j), (x_{i+1}, y_j), (x_i, y_{j+1})$ and $(x_{i+1}, y_j), (x_i, y_{j+1}), (x_{i+1}, y_{j+1})$, given by:

$$\frac{1}{2}\begin{vmatrix} x_i & y_j & 1 \\ x_{i+1} & y_j & 1 \\ x_i & y_{j+1} & 1 \end{vmatrix} + \frac{1}{2}\begin{vmatrix} x_{i+1} & y_j & 1 \\ x_i & y_{j+1} & 1 \\ x_{i+1} & y_{j+1} & 1 \end{vmatrix}, \tag{5.1}$$

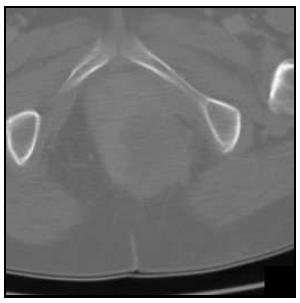
where $|\cdot|$ denotes the determinant operator. This measure is computed for every 2×2 region in the source image. The results of this computation for the CT tumor source and target images are shown in panels (f) and (g) respectively. Panel (f) depicts regions that have decreased in area, while panel (g) depicts regions that have increased in area. The images are color-coded such that the intensity is proportional to the change in area; in panel (f), the brighter the intensity, the larger the decrease in area, while in panel (g), the brighter the intensity, the larger the increase in area. Note that we are computing the change in area locally for the entire image; in order for us to find the change in area in the region of the tumor, the tumor must first be outlined, and the changes in that region summed.

5.1.2 Super-temporal resolution

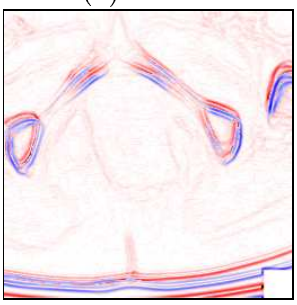
Super-temporal resolution refers to the process of increasing the temporal resolution in video sequences by introducing additional intermediate frames. A simple way to introduce an intermediate frame is to interpolate pixel intensities between the frames. In this approach, if the video sequence consists of, for example, a translating square, the intermediate frames would look like two overlapped squares. Instead, we would like the intermediate frame to have the square at an intermediate position. This can be achieved using image registration. In particular, consider two sequential frames, f_t and f_{t+1}. First, the two frames are registered, and a registration map $\mathcal{M}_t$ is obtained.



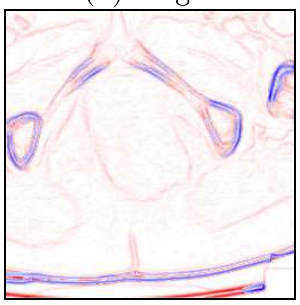
(a) source



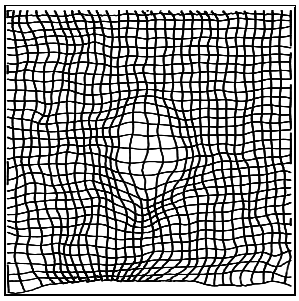
(b) target



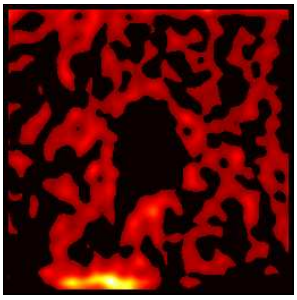
(c) errors before registration



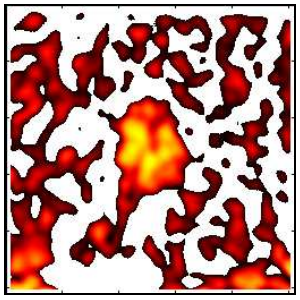
(d) errors after registration



(e) estimated geometric map



(f) regions of contraction



(g) regions of expansion

Figure 5.1: Quantifying tumor growth. Shown are the results of registering CT scans of a pelvic tumor taken over a period of time.

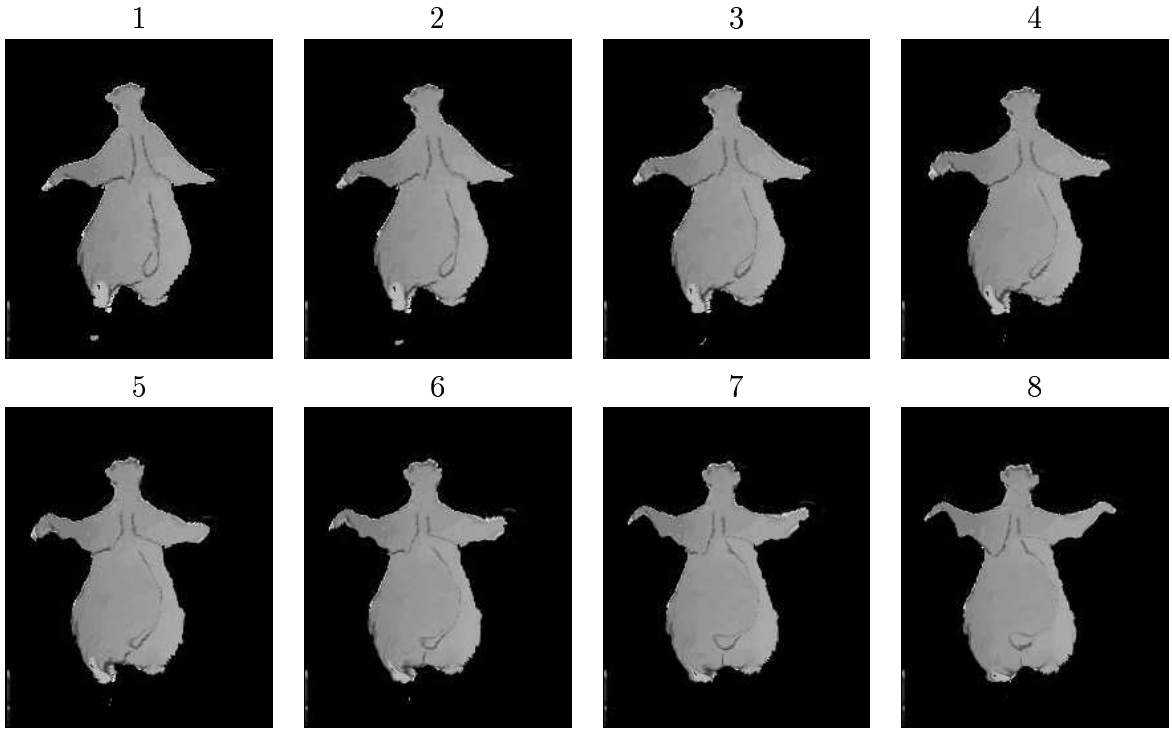


Figure 5.2: An application of super-temporal resolution - filling in of key frames in a cartoon animation. Frames 1 and 8 are the key frames, while frames 2-7 are frames generated using the super-temporal resolution technique.

In order to generate an intermediate frame $f_{t+\delta}$, where δ is a scalar between 0 and 1, all the vectors in $\mathcal{M}_t$ are scaled linearly by δ, and then applied to f_t.

One application of super-temporal resolution is the filling in of key frames in a cartoon animation. Currently, animators first establish key frames in the cartoon, and later work on the frames that go between them. Generating these in-between frames automatically (using the super-temporal resolution technique) would greatly reduce their workload. Shown in Figure 5.2 is an example of this; frames 1 and 8 are the key frames, while frames 2-7 have been synthetically generated using the super-temporal resolution technique described.

5.1.3 Digital aging

Another interesting application for image registration is digital aging. Given an image of a person's face, can we predict what it might look like after 10 years? One approach to this problem is to start with a database, which consists of images of faces of people at various ages. By registering each young face to its corresponding older face in the database, we could determine an average face, and an average geometric registration map. Precisely how to obtain these averages is in itself a subject of research. Then, in order to age a face, we first register the average face to it, and use the resulting geometric map to warp the average registration map. The resulting warped map can then be applied to the face to age it.

Given images of a face taken at two different ages, we can also estimate what the face would have looked like at intermediate ages [18]. Shown in Figure 5.3 are the results of registering faces of the same person taken at two different ages[1]. The super-temporal resolution technique from the previous section (Section 5.1.2) can then be applied to get faces at intermediate ages. Shown in Figure 5.4 are six intermediate faces generated in this manner, using the faces from Figure 5.3.

The results shown in Figure 5.3 can also be extended to make a young face younger, or an old face older, as shown in Figure 5.5. In this figure, the top row depicts the original young and old faces. The bottom row depicts a younger and older face, generated as follows. First, the young and old faces are registered to obtain a registration map (Figure 5.3). The inverse of this registration map is then applied to the young face to obtain a younger face, and the registration map is applied to the old face to get an older face.

[1]Images provided courtesy of Dave Perrett and Michael Burt, School of Psychology, University of St. Andrews, UK.

(a) source

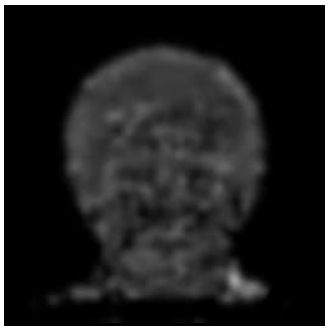
(b) target

(c) registered source

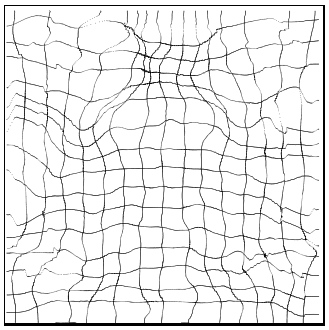
(d) error before registration

(e) error after registration

(f) estimated contrast



(g) estimated brightness



(h) estimated geometric map

Figure 5.3: Results of registering a young face to an older face.

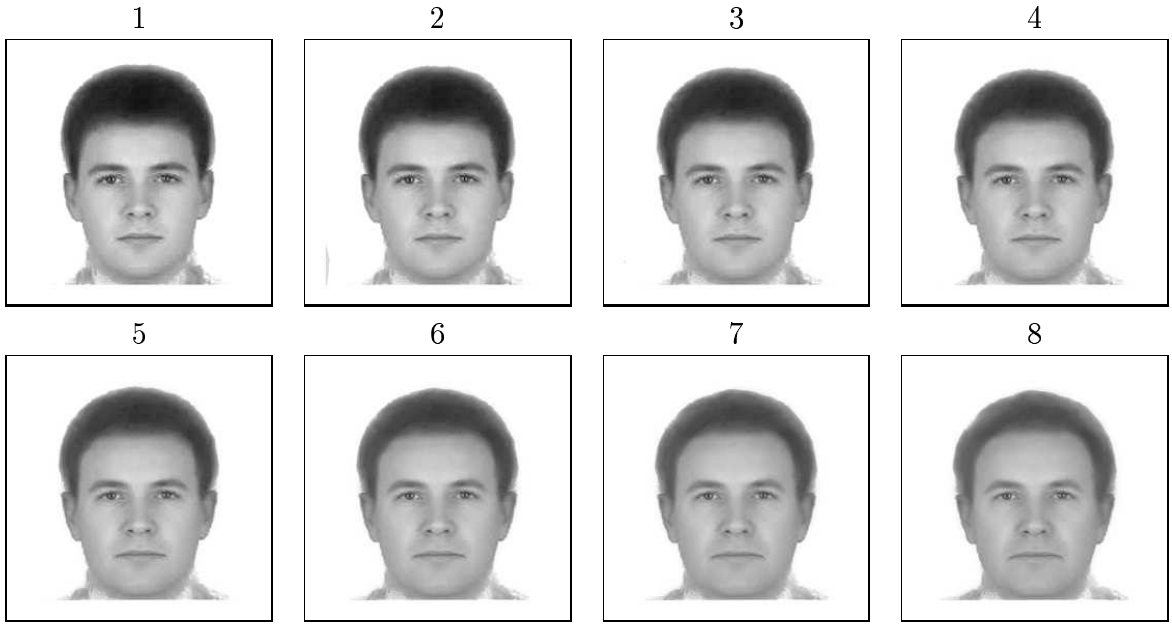


Figure 5.4: Faces with intermediate ages. Frames 1 and 8 are the key frames, while frames 2-8 are generated synthetically using the super-temporal resolution technique described in Section 5.1.2. The registration results from Figure 5.3 are used in this example.

Figure 5.5: Reversing the effects of aging, and aging a face further. The top row shows a younger face and an older face of the same person. The bottom row shows an even younger face and an even older face synthetically generated using the registration results shown in Figure 5.3.

5.1.4 Cartoon motion capture

Cartoon motion capturing [15] is a technique used to capture the motion of a cartoon character in an animation, which can then be re-targeted to other characters. This allows a complex sequence of motions artistically created by a master animator to be captured and transferred. We are currently exploring the use of registration to capture motion[2]. Shown in Figure 5.6 are sample results, in which the motion between each frame is determined using registration, and the results applied to a stick figure overlayed on the original sequence. The motion from the stick figure can now be transferred to another character.

Note that there are limitations to using image registration for this technique. When an object disappears and reappears into view, the motion captured using the registration algorithm is most likely incorrect. The ability of the algorithm to deal with missing data will help alleviate this problem, but it still does not guarantee a correct solution. Another problem that needs to be addressed is the re-targeting mechanism, as it will usually be the case that the original character and the re-targeted characters are completely different entities, thus requiring some manual process.

5.1.5 Validation/Improvement of physics based registration models

The non-rigid deformation of the brain during an operation can be modeled using finite element modeling [52, 46]. We are currently looking to validate and improve current brain deformation models by using the deformation field obtained by register-

[2]This is collaborative work with Lorie Loeb and Siwei Lyu from the Image Science Group at Dartmouth College.

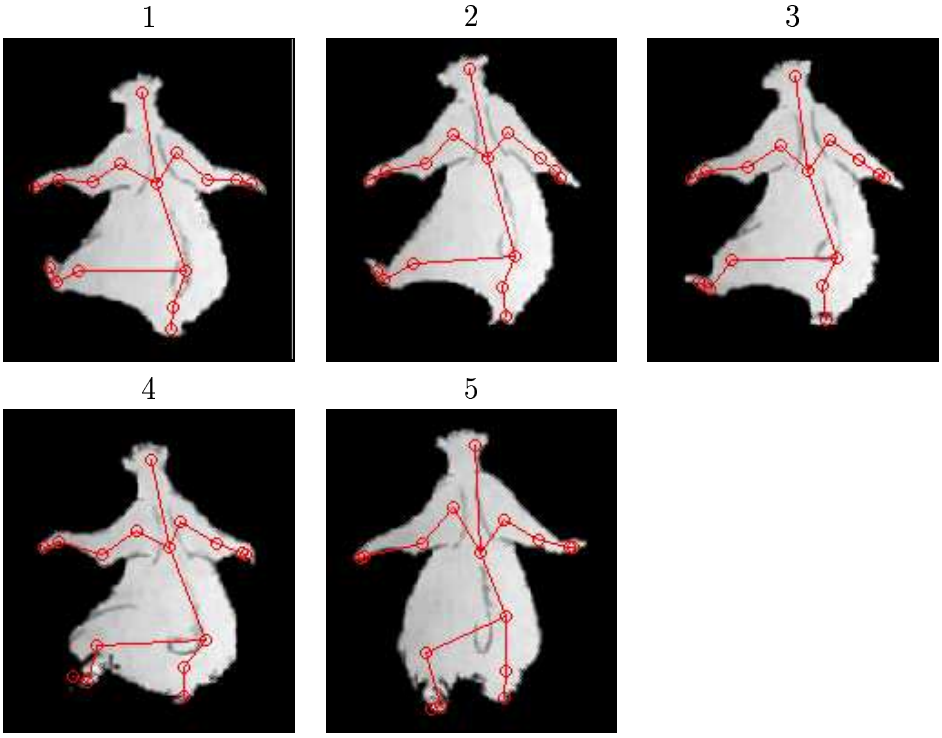


Figure 5.6: Cartoon motion capture. The motion captured is applied to a stick figure overlayed on the original sequence. This motion can now be re-targeted to another character.

ing volumes of a pig[3]. A balloon catheter is inserted between the skull and the brain of the pig, and the region is imaged before and after inflating the balloon, using an MR surface coil. The initial deformation model is built based on the coarse tracking of a few beads strategically placed in the brain region of the pig. Using the registration algorithm, a dense and accurate deformation field can be obtained, as shown in Figure 5.7. Note that the MR surface coil images produce an intensity gradient field, which is difficult to register directly using other standard techniques.

5.2 Optimizations for specific applications

The registration algorithm currently does not assume anything about the images being registered. However, given a large set of registered images of a specific type and modality, the resulting registration maps can be analyzed, and this additional information can be used to drive the registration. One approach is to use a prior on the smoothness of the registration maps, which can then be used to help constrain the possible solutions. These priors could also be used to drive a spatially varying smoothness constant. Priors can also be used directly on the motion vectors, choosing more likely solutions with respect to the existing registration results [64]. When registering images that have occlusions or partial data (using the EM extension), the constant parameter σ can also be estimated based on the existing registration maps. Note that the use of prior information (when available) should only strengthen the robustness of the algorithm.

[3]This is collaborative work with John West and Keith Paulsen, from the Biomedical Computation Group at the Thayer School of Engineering, Dartmouth College.

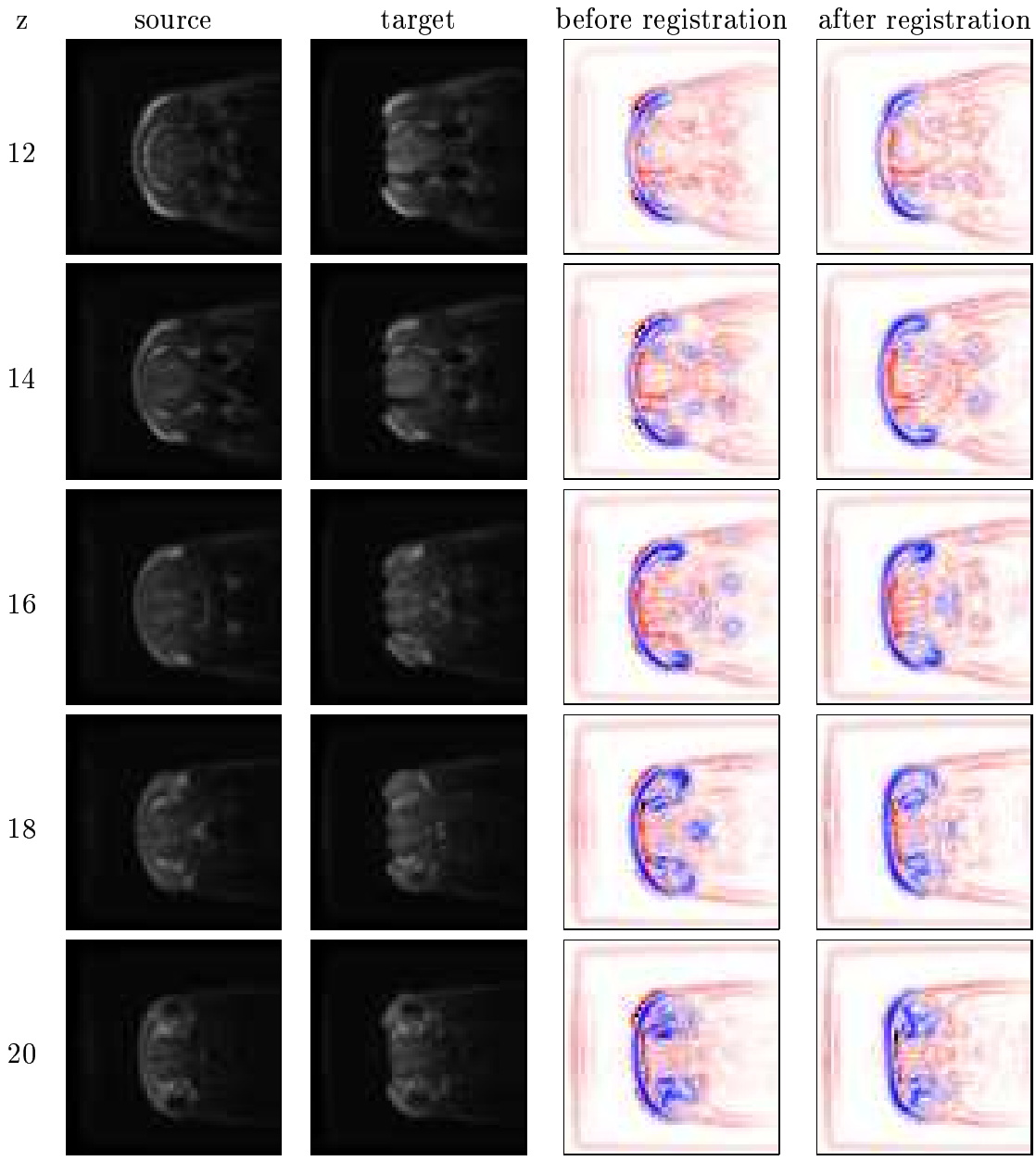


Figure 5.7: 3-D registration results of the pig experiment. Shown in columns 3 and 4 are the differences in the image edge-maps before and after registration.

5.3 Summary

In this chapter, a few sample applications are explored, some of them not related to medical imaging. The examples illustrate only a handful of potential applications for registration, and demonstrate the generality of the algorithm. Note that the algorithm has both its weaknesses and strengths. It is unable to deal with very large distortions, and is computationally intensive. However, it addresses a variety of clinical problems: it can find non-rigid distortions, explicitly deal with global and local intensity distortions, and also explicitly deal with occlusions or missing data. In Section 3 numerous results, both synthetic and clinical, have been presented, validating the assumptions made in the algorithm. In Section 4, the algorithm was analyzed using simulations to understand how it behaves with respect to the design assumptions. In summary, the algorithm is able to successfully capture a broad range of distortions over a broad range of imagery. This work is not meant to supplant the multitude of existing techniques; it merely provides an alternative approach and is meant to provide a basis for further research.

Bibliography

[1] E.H. Adelson, C.H. Anderson, J.R. Bergen, Peter J. Burt, and J.M. Ogden. Pyramid methods in image processing. *RCA Engineer*, 29(6), 1984.

[2] E.H. Adelson and J.R. Bergen. Spatiotemporal energy models for the perception of motion. *Journal of the Optical Society of America*, 2(2):284–299, Feb 1985.

[3] J. K. Aggarwal and N. Nandhakumar. On the computation of motion from sequences of images - a review. *Proceedings of the IEEE*, 76(8):917–935, Aug. 1988.

[4] N.M. Alpert, D. Berdichevsky, Z. Levin, E.D. Morris, and A.J. Fiscman. Improved methods for image registration. *Neuroimage*, 3:10–18, 1996.

[5] P. Anandan. A computational framework and an algorithm for the measurement of visual motion. *International Journal of Computer Vision*, 2(3):283–310, 1989.

[6] D.B. Rubin A.P. Dempster, N.M. Laird. Maximum lilelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 99(1):1–38, 1977.

[7] K.S. Arun, T.S. Huang, and S.D. Blostein. Least-squares fitting of two 3-d point sets. *IEEE Transactions on Pattern Analaysis and Machine Intelligence*, 9:698–700, 1997.

[8] J. Ashburner and K.J. Friston. Multimodal Image Coregistration and Partitioning - a Unified Framework. *NeuroImage*, 6(3):209–217, 1997.

[9] J. Ashburner and K.J. Friston. Nonlinear spatial normalization using basis functions. *Human Brain Mapping*, 7(4):254–266, 1999.

[10] J. Ashburner, P. Neelin, D.L. Collins, A.C. Evans, and K.J. Friston. Incorporating prior knowledge into image registration. *NeuroImage*, 6:344–352, 1997.

[11] R.K. Bajcsy and S. Kovacic. Multiresolution elastic matching. *Computer Vision, Graphics and Image Processing*, 46:1–21, 1989.

[12] J.L. Barron, D.J. Fleet, and S.S. Beauchemin. Performance of optical flow techniques. *International Journal of Computer Vision*, 12(1):43–77, Feb. 1994.

[13] F.L. Bookstein. Principal Warps: Thin-plate splines and the decomposition of deformations. *IEEE Transactions on Pattern Analaysis and Machine Intelligence*, 11(6):567–585, June 1989.

[14] F.L. Bookstein. Thin-plate splines and the atlas problem for biomedical images. *Information Processing in Medical Imaging*, pages 326–342, July 1991.

[15] C. Bregler, L. Loeb, E. Chuang, and H. Deshpande. Turning to the masters: Motion capturing cartoons. In *SIGGRAPH*, 2002.

[16] C. Broit. *Optimal Registration of Deformed Images.* PhD thesis, University of Pennsylvania, Pennsylvania, USA, 1981.

[17] L.G. Brown. A survey of image registration techniques. *ACM Computing Surveys*, 24(4):325–376, 1992.

[18] D.M. Burt and D.I. Perrett. Perception of age in adult caucasian male faces - computer graphic manipulation of shape and color information. In *Proceedings of the Royal Society of London, Series B-Biological Sciences*, volume 259, pages 137–143, 1995.

[19] T.M. Buzug, J. Weese, C. Fassnacht, and C. Lorenz. Using an entropy similarity measure to enhance the quality of DSA images with an algorithm based on template matching. In *Visualization in Biomedical Computing*, volume 1131, pages 235–240, Hamburg, Germany, Sept. 1996.

[20] E. De Castro and C. Morandi. Registration of translated and rotated images using finite Fourier transforms. *IEEE Transactions on Pattern Analaysis and Machine Intelligence*, 9(5):700–703, Sept. 1987.

[21] A. Collignon, F. Maes, D. Delaere, D. Vandermeulen, P. Suetens, and G. Marchal. *Information Processing in Medical Imaging*, chapter Automated multimodality image registration using information theory, pages 263–274. Kluwer, Dordrecht, 1995.

[22] T.M. Cover and J.A. Thomas. *Elements of Information Theory.* John Wiley & Sons, 1991.

[23] R.W. Cox and A. Jesmanowicz. Real-Time 3D Image Registration for Functional MRI. *Magnetic Resonance in Medicine*, 42:1014–1018, 1999.

[24] A.M. Dale, B. Fischl, and M.I. Sereno. Cortical surface-based analysis. I. Segmentation and surface reconstruction. *Neuroimage*, 9(2):179–194, Feb 1999.

[25] W.F. Eddy, M. Fitzgerald, and D.C. Noll. Improved image registration by using Fourier interpolation. *Magnetic Resonance in Medicine*, 36:923–931, 1996.

[26] A.C. Evans, S. Marrett, D.L. Collins, and T.M. Peters. Anatomical-functional correlative analysis of the human brain using three-dimensional imaging systems. In *Proceedings of the SPIE - The International Society for Optical Engineering*, volume 1092, pages 264–274, May 1989.

[27] H. Farid and E.P. Simoncelli. Optimally rotation-equivariant directional derivative kernels. In *International Conference on Computer Analysis of Images and Patterns*, pages 207–214, Berlin, Germany, Sept. 1997.

[28] B. Fischl, M.I. Sereno, and A.M. Dale. Cortical surface-based analysis. II: Inflation, flattening, and a surface-based coordinate system. *Neuroimage*, 9(2):195–207, Feb 1998.

[29] M.I. Miller G.E. Christensen, R.D. Rabbit. 3d brain mapping using a deformable neuroanatomy. *Physics in Medicine and Biology*, 39:609–618, march 1994.

[30] G.E. Christensen, R.D. Rabbit, M.I. Miller. A deformable neuroanatomy textbook based on viscous fluid mechanics. In *Proceedings of the 1993 Conference on Information Sciences and Systems*, pages 211–216, Johns Hopkins University, march 1995.

[31] J.C. Gee, M. Reivich, and R.K. Bajcsy. Elastically deforming an atlas to match anatomical brain images. *Journal of Computer Assisted Tomography*, 17(2):1–21, 1994.

[32] J. Hajnal, N. Saeed, E. Soar, A. Oatridge, I. Young, and G. Bydder. A registration and interpolation procedure for subvoxel matching of serially acquired MR images. *Journal of Computer Assisted Tomography*, 19:289–296, 1995.

[33] D.J. Heeger. Optical flow using spatio-temporal filters. *International Journal of Computer Vision*, 1(4):279–302, 1988.

[34] B.K.P. Horn. *Robot Vision*. MIT Press, Cambridge, MA, 1986.

[35] B.K.P Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America*, 4:629–642, 1987.

[36] B.K.P Horn, H.M. Hilden, and S. Negahdaripour. Closed-form solution of absolute orientation using orthonormal matrices. *Journal of the Optical Society of America*, 5:1127–1135, 1988.

[37] B.K.P. Horn and B. G. Schunck. Determining optical flow. *Artificial Intelligence*, 17(1-3):185–203, Aug 1981.

[38] W.S. Kerwin and C. Yuan. Active edge maps for medical image registration. In *Proceedings of the SPIE - The International Society for Optical Engineering*, pages 516–526, July 2001.

[39] K.J. Friston. Spatial normalization: A new approach. In *BrainMap*, volume 2, pages 165–189, San-Antonio, Texas, 1995.

[40] L. Lemieux, U.C. Wieshmann, N.F. Moran, D.R. Fish, and S.D. Shorvon. The detection and significance of subtle changes in mixed-signal brain lesions by serial MRI scan matching and spatial normalization. *Medical Image Analaysis*, 2(3):227–242, 1998.

[41] H. Lester and S.R. Arridge. A survey of hierarchical non-linear medical image registration. *Pattern Recognition*, 32(1):129–149, Jan 1999.

[42] B.D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *International Joint Conference on Artificial Intelligence*, pages 674–679, Vancouver, 1981.

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

[44] C.R. Maurer Jr. and J.M. Fitzpatrick. *Interactive Image-Guided Neurosurgery*, chapter A Review of Medical Image Registration. American Association of Neurological Surgeons, Park Ridge, IL., 1993.

[45] G. Medioni and R. Nevatia. Matching images using linear features. *IEEE Transactions on Pattern Analaysis and Machine Intelligence*, 6(6):675–685, Nov. 1984.

[46] M.I. Miga, K.D. Paulsen, P.J. Hoopes, F.E. Kennedy, A. Hartov, and D. W. Roberts. In vivo analysis and modeling of interstitial pressure in the brain under surgical loading. *Journal of Biomechanical Engineering*, 122(4):354–363, 2000.

[47] D. Mishra, A.K. Chan, and C.K. Chui. Histogram equalization, image registration, and data fusion for multispectral images. In *Proceedings of the SPIE - The International Society for Optical Engineering*, volume 2496, pages 1025–1031, June 1995.

[48] M.L Nack. Rectification and registration of digital images and the effect of cloud detection. In *Machine Processing of Remotely Sensed Data*, pages 12–23, West Lafayette, IN, June 1977.

[49] B.S. Nutter, S. Mitra, and T.F. Krile. Image registration algorithm for a pc-based system. In *Proceedings of the SPIE - The International Society for Optical Engineering*, volume 829, pages 214–221, 1987.

[50] O. Nestares and D.J. Heeger. Robust Multiresolution Alignment of MRI Brain Volumes. *Magnetic Resonance in Medicine*, 43:705–715, 2000.

[51] N. Paragios, M. Rousson, and V. Ramesh. Knowledge-based registration and segmentation of the left ventricle: a level set approach. In *IEEE workshop on Applications of Computer Vision*, pages 37–42, 2002.

[52] K. D. Paulsen, M. I. Miga, F. E. Kennedy, P. J. Hoopes, A. Hartov, and D. W. Roberts. A computational model for tracking subsurface tissue deformation during stereotactic neurosurgery. *IEEE Transactions on Biomedical Engineering*, 46(2):213–225, 1999.

[53] C.A. Pelizarri, G.T.Y. Chen, D.R. Spelbring, R.R. Weichselbaum, and C.T. Chen. Accurate three-dimensional registration of CT, PET and/or MR images of the brain. *Journal of Computer Assisted Tomography*, 13(1):20–26, 1989.

[54] A. P. Pentland. Fractal based description of natural scenes. *IEEE Transactions on Pattern and Machine Intelligence*, 6(6):661–674, 1984.

[55] P.J. Kostelec, J.B. Weaver, and D.M. Healy, Jr. Multiresolution elastic image registration. *Medical Physics*, 25(9):1593–604, 1998.

[56] P.J. Kostelec, J.B. Weaver, and D.M. Healy, Jr. Multiresolution elastic image registration. *Medical Physics*, 25(9):1593–604, 1998.

[57] A. Roche, G. Malandain, N. Ayache, and S. Prima. Towards a better comprehension of similarity measures used in medical image registration. In *Medical Image Computing and Computer Assisted Intervention*, volume 1679, pages 555–566, 1999.

[58] D. Rueckert, L.I. Sonoda, C. Hayes, D.L.G. Hill, M.O. Leach, and D.J. Hawkes. Nonrigid registration using free-form deformations: application to breast MR images. *IEEE Transactions on Medical Imaging*, 18(8):712–721, August 1999.

[59] S. Negahdaripour and C.-H. Yu. A generalized brightness change model for computing optical flow. In *International Conference of Computer Vision*, pages 2–11, Berlin, Germany, May 1993.

[60] S. Periaswamy, J.B. Weaver, D.M. Healy, Jr., and P.J. Kostelec. Automated multiscale elastic image registration using correlation. In *Proceedings of the SPIE - The International Society for Optical Engineering*, volume 3661, pages 828–38, 1999.

[61] P. H. Sconemann. A generalized solution of the orthogonal procrustes problem. *Pschometrika*, 31(1):1–10, 1966.

[62] J. Shi and C. Tomasi. Good features to track. In *Computer Vision and Pattern Recognition*, pages 593–600, Seatle, WA, USA, June 1994.

[63] W.S.V. Shih., W.C. Lin, and C.T.Chen. Contour-model-guided nonlinear deformation model for intersubject image registration. In *Proceedings of the SPIE - The International Society for Optical Engineering*, volume 3034, pages 611–620, April 1997.

[64] E.P. Simoncelli. *Distributed analysis and representation of visual motion.* PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 1993.

[65] C. Studholme, D.L.G. Hill, and D.J. Hawkes. Incorporating connected region labelling into automated image registration using mutual information. *Mathematical methods in biomedical image analysis*, pages 23–31, June 1996.

[66] C. Studholme, D.L.G. Hill, and D.J. Hawkes. An overlap invariant entropy measure of 3d medical image alignment. *Pattern Recognition*, 32(1):71–86, 1999.

[67] P. Thévenaz and M.A. Unser. Spline pyramids for intermodal image registration using mutual information. In *Proceedings of the SPIE - The International Society for Optical Engineering*, volume 3169, pages 236–247, Oct 1997.

[68] M. Unser, A. Aldroubi, and C. Gerfen. A multiresolution image registration procedure using spline pyramids. In *Proceedings of the SPIE - Mathematical Imaging: Wavelets and Applications in Signal and Image Processing*, volume 2034, pages 160–170, 1993.

[69] P.A. Van den Elsen, E.-J.D. Pol, and M.A. Viergever. Medical image matching - a review with classification. *IEEE Engineering in Medicine and Biology*, 12(1):26–39, March 1993.

[70] B.C. Vemuri, J. Ye, Y. Chen, and C.M. Leonard. A level-set based approach to image registration. In *IEEE Workshop on Mathematical Methods in Biomedical Image Analysis*, pages 86–93, 2000.

[71] P. Viola and W.M. Wells III. Alignment by maximization of mutual information. In *IEEE International Conference on Computer Vision*, pages 16–23, Cambridge, MA, USA, 1995.

[72] J. Vlad, S. Eulenstein, W. Wlodarczyk, P. Wust, and R. Felix. Registration of portal images for online correction of positioning errors during radiation therapy of prostate cancer. In *Proceedings of the SPIE - The International Society for Optical Engineering*, volume 4319, pages 61–68, May 1995.

[73] A.B. Watson and A.J. Ahumada. *Motion: Perception and representation*, chapter A look at motion in the frequency domain, pages 1–10. ACM, 1983.

[74] J. West, J. Fitzpatrick, M. Wang, B. Dawant, C. Maurer, R. Kessler, and R. Maciunas. Comparison and evaluation of retrospective intermodality image registration techniques. In *Proceedings of the SPIE - The International Society for Optical Engineering*, pages 332–347, Newport Beach, CA, 1996.

[75] R.P. Woods, S.R. Cherry, and J.C. Mazziotta. Rapid automated algorithm for aligning and reslicing PET images. *Journal of Computer Assisted Tomography*, 16:620–633, 1992.

[76] R.P. Woods, S.T. Grafton, C.J. Holmes, S.R. Cherry, and J.C. Mazziotta. Automated image registration: I. General methods and intrasubject, intramodality validation. *Journal of Computer Assisted Tomography*, 22:141–154, 1998.

[77] R.P. Woods, J.C. Mazziotta, and S.R. Cherry. MRI-PET registration with automated algorithm. *Journal of Computer Assisted Tomography*, 17:536–546, 1993.