1-4-2016

Respecting Anatomically Rigid Regions in Nonrigid Registration

Eric J. Spangler

Recommended Citation

Spangler, Eric J., "Respecting Anatomically Rigid Regions in Nonrigid Registration" (2016). Thesis. Rochester Institute of Technology. Accessed from

Respecting Anatomically Rigid Regions in Nonrigid Registration

by

Eric J. Spangler

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science in Applied Mathematics
School of Mathematical Sciences, College of Science

Rochester Institute of Technology
Rochester, NY
January 4, 2016

Committee Approval:

R·I·T
College of
SCIENCE

Nathan D. Cahill Date
School of Mathematical Sciences
Thesis Advisor

Tamas Wiandt Date
School of Mathematical Sciences
Committee Member

Elizabeth M. Cherry Date
School of Mathematical Sciences
Director of Graduate Programs

Abstract

Medical image registration has received considerable attention in the medical imaging and computer vision communities because of the variety of ways in which it can potentially improve patient care. This application of image registration aligns images of regions of the human body for comparing images of the same patient at different times, for example, when assessing differences in a disease over time; comparing images of the same anatomical structure across different patients, for example, to understand patient variability; and comparing images of the same patient from different modalities that provide complementary information, for example, CT and PET to assess cancer.

The two primary types of registration make use of rigid and nonrigid transformations. Medical images typically contain some regions of bone, which behave rigidly, and some regions of soft tissue, which are able to deform. While a strictly rigid transformation would not account for soft tissue deformations, a strictly nonrigid transformation would not abide by the physical properties of the bone regions. Over the years, many image registration techniques have been developed and refined for particular applications but none of them compute a continuous transformation simultaneously containing both rigid and nonrigid regions.

This thesis focuses on using a sophisticated segmentation algorithm to identify and preserve bone structure in medical image registration while allowing the rest of the image to deform. The registration is performed by minimizing an objective function over a set of transformations that are defined in a piecewise manner: rigid over a portion of the domain, nonrigid over the remainder of the domain, and continuous everywhere. The objective function is minimized via steepest gradient descent, yielding an initial boundary value problem that is discretized in both time and space and solved numerically using multigrid. The registration results are compared to results of strictly rigid and nonrigid registrations.

Contents

List of Figures

Chapter 1

Introduction

Image registration is the process of aligning multiple images into the same coordinate system. In medical imaging applications, the images are typically captured at different times or from different viewpoints. The process involves aligning a floating image with a reference image by computing the transformation that maps points in the floating image to relative points in the reference image.

Medical images can be produced from numerous modalities (e.g. computed tomography, magnetic resonance, ultrasound) and registration can be performed to align images within the same modality or from different modalities, and the images can be from the same or different patients. In this thesis, we will focus on registering magnetic resonance (MR) images from the same patient. The images in question will be three-dimensional images of a brain; hence, we can denote mathematically the floating image as $F : \Omega_F \to \mathbb{R}$ and the reference image as $R : \Omega_R \to \mathbb{R}$ where $\Omega_F, \Omega_R \subseteq \mathbb{R}^3$.

Image registration is comprised of three main components: a geometric transformation, a similarity measure, and an optimization process. The geometric transformation, $\Phi : \Omega_F \to \mathbb{R}^3$, is the mapping of all points in the floating image to the three-dimensional space that contains the reference image in an attempt to align like regions. The similarity measure is a function, $\mathcal{J}\left(R, F(\Phi)\right)$, that quantifies how well the reference and deformed

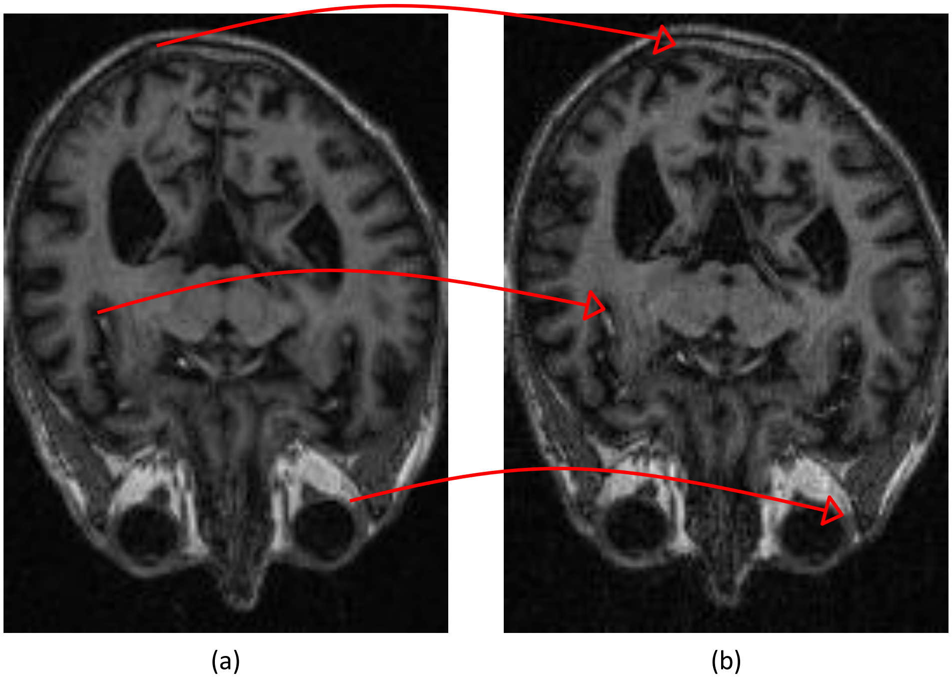


Figure 1.1: Mapping from floating image to reference image

floating images are aligned. The optimization process defines how to determine the geometric transformation that optimizes the similarity measure between the transformed floating image and the reference image.

Two different types of transformations will be considered on in this thesis: rigid transformations and deformable, or non-rigid, transformations. As the name suggests, a rigid transformation consists of rotations and translations that rigidly move the image. A non-rigid transformation allows for local deformations often viewed as a warping of the original image.

1.1 Motivation

As mentioned previously, the desired registration in this case will be between two different 3-D brain images of the same subject at different times. Using a rigid registration would roughly align the two images, but the entire head is not a rigid body, so this would not be ideal for an accurate solution. The soft tissue almost certainly requires non-rigid transformations to accurately register, but the skull obeys a rigid transformation. Therefore, a strictly non-rigid registration would also be insufficient, leaving the skull susceptible to deformations. Clearly, neither strictly rigid nor strictly nonrigid transformations would accurately register 3-D brain MR images in all areas of the head. When constructing a registration transformation, there are numerous factors that need to be considered: the rigid anatomical structures, the ability for soft tissue/organs to deform, the necessity for a transformation to be physically possible (no points crossing each other), and the lack of interest in the region of the image that does not depict the subject. Next, we will discuss what others have previously done to address these issues.

1.2 Literature Review

Image registration has been used in many medical applications such as tracking the growth of a tumor and comparing brain activity from numerous subjects. For examples of general forms of registration paired with medical imaging applications, see [24, 19, 21]. There are numerous methods for performing registration that range from fully rigid to nonrigid transformations, but they do not adequately or easily depict accurate anatomical behavior. Therefore, a large amount of work has been performed to create more sophisticated types of transformations. Characteristics of an ideal registration are generally specific to the desired application. For example, an image of a heart is entirely depicting soft tissue which is subject to deformations so a rigid transformation would not be desired when registering two cardiac images.

One particular area of interest that is very common in medical imaging is the registration of images containing both rigid bone regions and soft tissue regions. This is most easily seen in torso images, capturing the ribcage as well as internal organs, and also skull images which are surrounding soft-tissue brains. When registering two torso CT images, it is clear that a rigid transformation would not accurately represent the behavior of the soft tissue organs, yet a non-rigid transformation would not preserve the rigid nature of the ribs.

Nonrigid registration algorithms typically capture only very small deformations but torso images could potentially exhibit rather large deformations. In order to capture large deformations, a fluid model could be used to measure small velocity of change which can then be integrated to result in larger deformations [9, 6]. Woods et al. [23] recommend using an affine transformation when handling mixed regions between images from different subjects; however, for intrasubject registration, the affine transformation results contained more inconsistencies than those obtained with a strictly rigid model. Arsigny et al. [1] developed new types of transformations, known as polyrigid and polyaffine, in order to find a middle ground between the small number of degrees of freedom in a rigid transformation versus the large number of degrees of freedom in a fully nonrigid transformation. A group of anchor points are defined, where rigid or affine transformations are computed, and the transformation at all other points is equal to some interpolated combination of the transformation from all nearby anchor points.

Similar to this type of transformation is a proposed registration method that identifies rigid bodies in an image, solves for the ideal transformation for each rigid body in the image, then transforms all remaining points using a weighted linear combination of the pre-defined linear transformations of the rigid regions [16]. Not too different from this method, Edwards et al. [11] suggests observing a brain MRI as nonrigid as a result of surgery or radiotherapy. The procedure is to identify the rigid regions, use feature matching with 6 points from each image, transform the rigid region and nonrigid region separately, and then combine the results. The nonrigid regions are mapped using three feature points which determine how to interpolate the rest of the nonrigid region.

While these are all worthwhile techniques and substantial improvements from strictly rigid or nonrigid registration, none of these methods quite solve for both rigid and nonrigid transformations simultaneously. Rather than use landmarks or individually run the rigid and nonrigid transformations, it would be beneficial to view the registration as one continuous transformation since that is what it represents physically.

1.3 Problem to solve

This thesis proposes a method that enables non-parametric registration of soft tissue regions while simultaneously allowing rigid registration of bony regions. In Chapter 2, we discuss the main concepts necessary to construct and propose the desired registration method. Chapter 3 describes the proposed method using the discussed background components. Chapter 4 discretizes the continuous quantities in the proposed method in order to show how it can be implemented practically. In Chapter 5, we discuss experimentation from the proposed registration algorithm along with results.

Chapter 2

Background

In order to propose a new method of nonrigid registration that respects the rigid regions, we will present some preliminary concepts in this chapter. We will assume that the rigid regions of our images are exclusive to the points representing bone regions. In this chapter, we describe how image segmentation can be used to extract these regions in order to assist with the registration process. In addition, we describe both rigid and nonrigid registration separately.

2.1 Identifying bones

Many image segmentation algorithms could potentially find a high-quality segmentation of bone regions, but most of these methods are either user-intensive or particularly challenging to set up. Examples of algorithms with these two weaknesses utilize the methods of random walks [14], which requires the user to identify numerous seed points inside and outside the region of interest, and active contours [8], which requires the user to define an initial contour. Since segmentation is only a portion of this registration process, it would be ideal to use a segmentation algorithm that is fully automatic without compromising any of the segmentation quality. For this reason, an algorithm based on the theory of graph cuts [5] has been chosen.

Consider image segmentation to map every pixel in the image $p \in \mathcal{P}$ to a finite set of values or labels, $\mathcal{L}$, which represent the various classes of objects found in the image. The goal of the segmentation is to find a labeling $f : \mathcal{P} \mapsto \mathcal{L}$ that maps each pixel $p \in \mathcal{P}$ to a label $f_p \in \mathcal{L}$ where f is both piecewise smooth and consistent with the observed image data. This problem can be imagined in terms of energy minimization as an attempt to find the labeling f that minimizes the energy function

$$\min_{f} \qquad E(f) = E_{smooth}(f) + E_{data}(f). \tag{2.1}$$

where E_{smooth} measures the extent to which f is not piecewise smooth and E_{data} measures the dissimilarity between f and the image data. Consider E_{data} to be of the form

$$E_{data}(f) = \sum_{p \in \mathcal{P}} D_p(f_p) = \sum_{p \in \mathcal{P}} \left(f_p - I_p\right)^2, \tag{2.2}$$

where D_p measures how well the label f_p fits point p and I_p is the image value at point p. Consider E_{smooth} to be of the form

$$E_{smooth}(f) = \sum_{\{p,q\} \in \mathcal{N}} V_{p,q}(f_p, f_q) = \sum_{\{p,q\} \in \mathcal{N}} \min\left(\mathcal{K}, |f_p - f_q|\right) \tag{2.3}$$

where $V_{p,q}$ measures the pair-wise difference in label values of points p and q, $\mathcal{N}$ is the set of all neighboring pixels, and $\mathcal{K}$ is a constant representing the maximum value of the pair-wise difference. Boykov et al. [5] describes a way to compute this energy minimization by finding a minimum weight cut set of a graph representing the image.

Edge	Weight	For
$t_p^{\bar{\alpha}}$	∞	$p \in \mathcal{P}_\alpha$
$t_p^{\bar{\alpha}}$	$\mathcal{D}_p\left(f_p\right)$	$p \notin \mathcal{P}_\alpha$
t_p^{α}	$\mathcal{D}_p\left(\alpha\right)$	$p \in \mathcal{P}$
$e_{\{p,a\}}$	$\mathcal{V}\left(f_p, \alpha\right)$	$\{p, q\} \in \mathcal{N}, f_p \neq f_q$
$e_{\{a,q\}}$	$\mathcal{V}\left(\alpha, f_q\right)$	
$t_a^{\bar{\alpha}}$	$\mathcal{V}\left(f_p, f_q\right)$	
$e_{\{p,q\}}$	$\mathcal{V}\left(f_p, \alpha\right)$	$\{p, q\} \in \mathcal{N}, f_p = f_q$

Table 2.1: Edge weights associated with an α expansion [5]

2.1.1 Graph cut expansion algorithm

The graph cuts expansion algorithm considers an expansion on the label $\alpha \in \mathcal{L}$ to be the process of switching current non-α pixels to α by finding the smallest weighted cut set to disconnect α from non-α labels in a graph with weighted edges corresponding to (2.2) and (2.3). The process of sequentially performing an expansion on every label in $\mathcal{L}$ will be referred to as a cycle. The algorithm continuously performs expansion cycles and then stops once an entire cycle finishes with zero label changes. Each expansion will have its own associated graph.

The graph $\mathcal{G}_\alpha = \langle \mathcal{V}_\alpha, \mathcal{E}_\alpha \rangle$ associated with an α-expansion is defined in the following way. The set of nodes $\mathcal{V}_\alpha$ consists of a node for every pixel in the image, one for the label α, one to represent all non-α labels, and one in between each set of neighboring pixels that are currently mapped to different labels (known as auxiliary nodes). The set of edges $\mathcal{E}_\alpha$ consists of an edge from each pixel node to both of the label nodes (t-links), an edge between all neighboring pixels of the same label (n-links), two edges between all neighboring pixels of different labels (one from p to the auxiliary node and one from the auxiliary node to q), one edge from each auxiliary node to the non-α label node. A 1-D representation of this can be seen in Figure 2.1. The weights associated with each edge in Figure 2.1 can be found in Table 2.1.

Given the current segmentation f, the ideal expansion on α is computed by finding

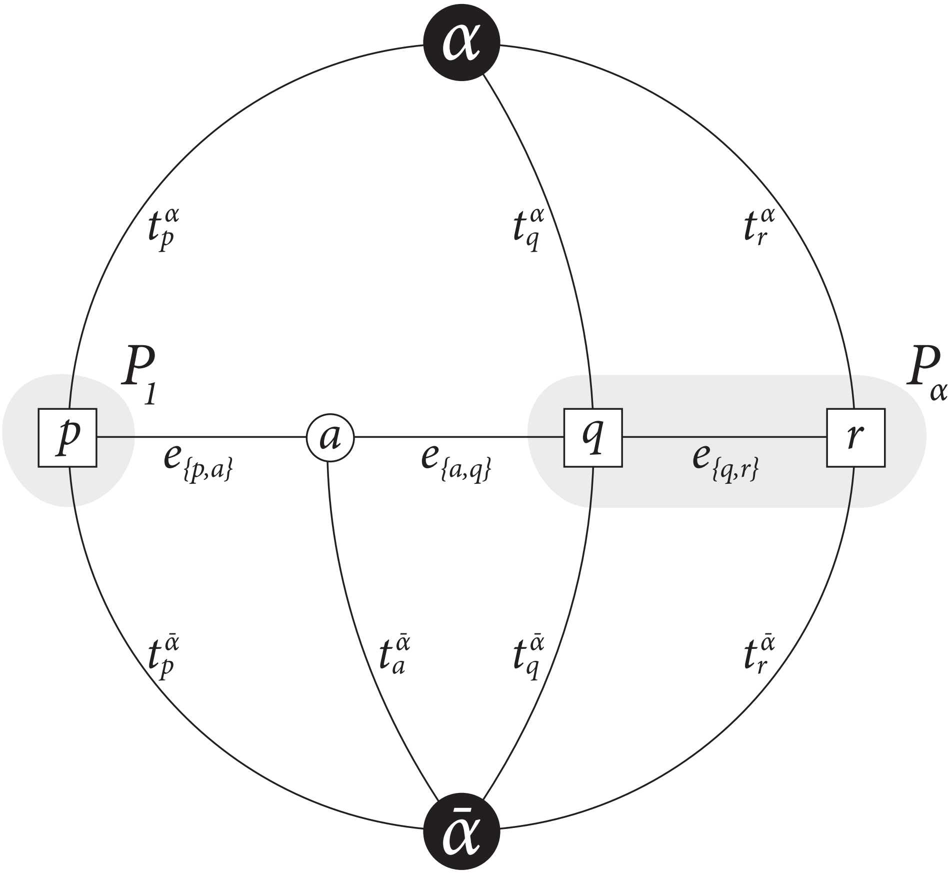


Figure 2.1: 1D example of $\mathcal{G}_\alpha$ from Boykov et al. [5]

a minimum weight cut set of edges $C \subset \mathcal{E}_\alpha$ that will disconnect α and $\bar{\alpha}$ in the graph $\mathcal{G}_\alpha^C = \langle \mathcal{V}_\alpha, \mathcal{E}_\alpha - C \rangle$. If C disconnects p from α then the resulting label of p is α. Otherwise, p keeps its previous label. A cut set will always contain one t-link adjacent to each pixel and one edge adjacent to each auxiliary node. The cut options for a single set of neighboring pixels are shown in Figure 2.2. The optimization process can be read about in detail in [5]. The result of the expansion algorithm is then used as the input when expanding upon the next label in $\mathcal{L}$.

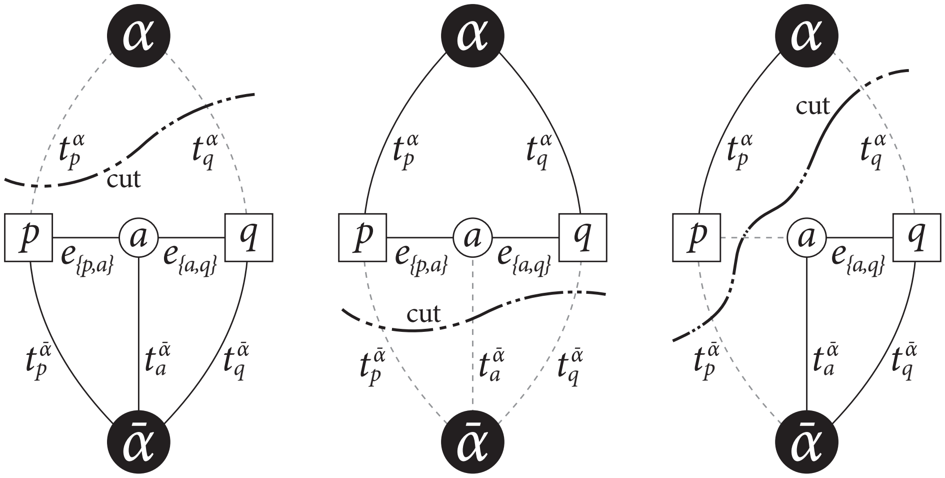


Figure 2.2: Possible cut sets between neighboring pixels as illustrated in Boykov et al. [5]. (a) labels both pixels as α, (b) leaves both pixels with their previous non-α label, (c) labels p as α and q with its previous label.

2.1.2 Defining inputs

In order to define the data cost and smoothness, a value must be associated with each label. To obtain these label values, the k-means algorithm is used to partition the image data into k clusters and compute their means [17]. As seen in Figure 2.3, setting $k = 6$ appears to best capture the bones in their own cluster when viewing chest CT images. This method of segmentation is not as sophisticated as the graph cuts method, therefore, it is only used to capture the estimated mean values.

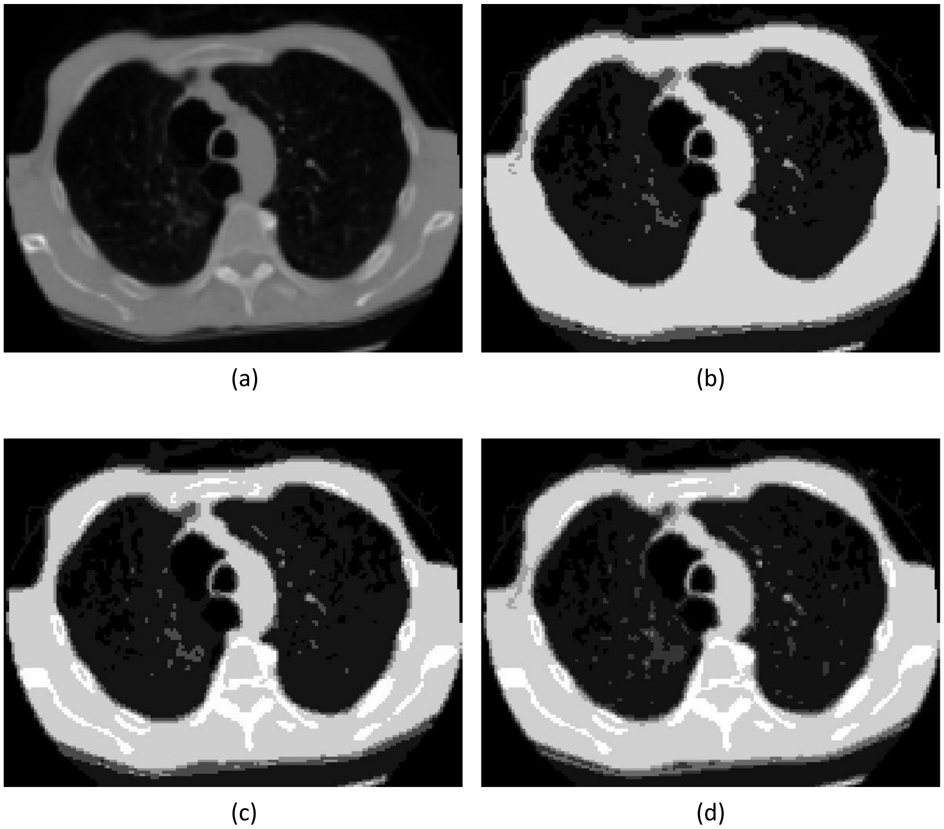


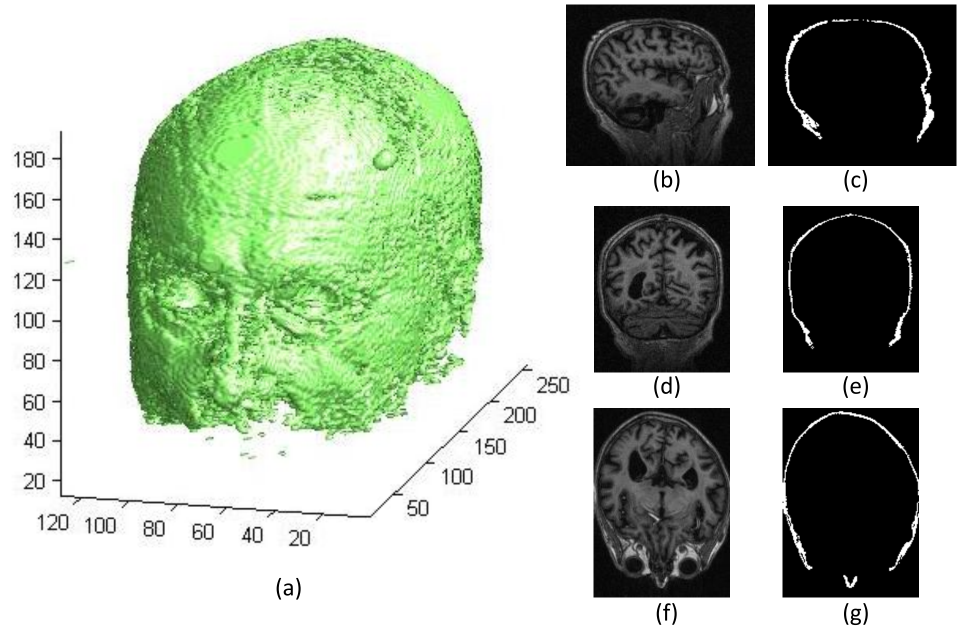
Figure 2.3: Clustering results using the k-means algorithm. (a) raw image, (b) k=5, (c) k=6, (d) k=7.

The publicly available code [22, 4, 15, 3] that implements the graph cuts expansion algorithm in [5] requires only three inputs: the data cost, smoothness, and the neighborhood, $\mathcal{N}$. With the label values output by the k-means algorithm, the data cost can be computed using formula (2.2) and stored in a $|\mathcal{P}| \times |\mathcal{L}|$ matrix containing the values for $D_p(f_p)$ for all $p \in \mathcal{P}$ and all possible labels $f_p \in \mathcal{L}$. Together, the smoothness and the neighborhood create (2.3). The smoothness is stored in a $|\mathcal{L}| \times |\mathcal{L}|$ matrix of all possible values of $V_{p,q}$ created using the recently found label values. The neighborhood, which determines the sets $\{p, q\}$ where $V_{p,q}$ is defined, is encoded in a $|\mathcal{P}| \times |\mathcal{P}|$ matrix where point (i, j) determines the weight of the connection between points i and j. In our case, we will not be uniquely weighting any neighboring pixel connections, so all neighboring pixel relations are set to a value of 1. Therefore, the smoothness will be the same for any pair of pixels mapped to the same labels. Neighbors are only defined in one direction so if $\{i, j\}$ is a set of neighbors, it is not appropriate to populate both (i, j) and (j, i) in the neighborhood matrix.

The result of the segmentation is a labeled volume that is the size of the initial image. For our application, we only really care about segmenting bone from other structures, so we will create a binary mask based solely on this cluster that can be fed into the rigid registration. A 3-D rendering of an example mask extracted from a brain MRI can be found in Figure 2.4. A segmentation of the image used for the k-means example is segmented using the graph cuts expansion algorithm in Figure 2.5.

2.2 Rigid Registration

Rigid transformations consist of translations and rotations. This means that the spatial relationship of all points in the image is preserved when transformed. Such a transformation can be described by the matrix equation $\Phi(\mathbf{x}) = \mathbf{R}\mathbf{x} + \mathbf{t}$, where $\mathbf{R}$ is an $n \times n$ rotation matrix and $\mathbf{t}$ is an $n \times 1$ vector that denotes the translation. In our case, $n = 3$.



(a) (b) (c) (d) (e) (f) (g)

Figure 2.4: (a) Rendering of bone mask as output by graph cut segmentation algorithm. (b) Sagittal slice with segmentation (c). (d) Coronal slice with segmentation (e). (f) Axial slice with segmentation (g).

Figure 2.5: Bone mask output by graph cuts segmentation of image previously used to observe various k-means algorithm outputs.

Given point $\mathbf{x} \in \Omega_F$, the transformation $\Phi(\mathbf{x})$ can be represented by:

$$\begin{pmatrix} \Phi_1(\mathbf{x}) \\ \Phi_2(\mathbf{x}) \\ \Phi_3(\mathbf{x}) \end{pmatrix} = \mathbf{R} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} + \begin{pmatrix} t_1 \\ t_2 \\ t_3 \end{pmatrix}, \tag{2.4}$$

where the rotation matrix $\mathbf{R} = \mathbf{R}_1\mathbf{R}_2\mathbf{R}_3$ is a product of the following Givens rotation matrices [13]:

$$\mathbf{R}_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta_1) & -\sin(\theta_1) \\ 0 & \sin(\theta_1) & \cos(\theta_1) \end{pmatrix}$$

$$\mathbf{R}_2 = \begin{pmatrix} \cos(\theta_2) & 0 & \sin(\theta_2) \\ 0 & 1 & 0 \\ -\sin(\theta_2) & 0 & \cos(\theta_2) \end{pmatrix}$$

$$\mathbf{R}_3 = \begin{pmatrix} \cos(\theta_3) & -\sin(\theta_3) & 0 \\ \sin(\theta_3) & \cos(\theta_3) & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

where matrix $\mathbf{R}_i$ rotates the point $\mathbf{x}$ around the i^{th}-axis by angle θ_i. Therefore, one way to parameterize rigid transformations can be expressed using three rotation parameters and three translation parameters. Let us refer to this set of parameters as

$$\mathbf{\Theta} = [\theta_1, \theta_2, \theta_3, t_1, t_2, t_3] \tag{2.5}$$

and subsequently let us rewrite our transformation as $\Phi(\mathbf{x}) = \mathbf{R}_{\boldsymbol{\Theta}}\mathbf{x} + \mathbf{t}_{\boldsymbol{\Theta}}$ where $\mathbf{R}_{\boldsymbol{\Theta}}$ and $\mathbf{t}_{\boldsymbol{\Theta}}$ are the rotation and translation according to the set of rigid parameters $\boldsymbol{\Theta}$.

These rigid transformation parameters are manipulated to optimize an energy function that observes the relationship between the reference image and the transformed floating image to determine the best possible registration transformation. The main component of the energy function is the dissimilarity measure, which computes the dissimilarity between the transformed floating image and the reference image. Common dissimilarity measures include the sum of squared differences, mean of squared differences, squared correlation coefficient, mutual information, as well as many others [19]. The energy function takes the following form:

$$\min_{\boldsymbol{\Theta}} \qquad \mathcal{E}(R, F, \boldsymbol{\Theta}) := \mathcal{J}(R, F(\Phi(\mathbf{x}))) = \mathcal{J}(R, F(\mathbf{R}_{\boldsymbol{\Theta}}\mathbf{x} + \mathbf{t}_{\boldsymbol{\Theta}}))\,, \tag{2.6}$$

where $\mathcal{J}$ is the dissimilarity measure that quantifies the dissimilarity between the reference image R and the deformed floating image $F(\Phi)$. Several examples of dissimilarity measures can be found in Table 2.2 and many other examples can be found in [7].

A variety of optimization procedures can be used to find the set of rigid parameters that best minimize the energy in (2.6). Some methods, like the simplex method [10], require evaluating only the similarity measure; others, like steepest descent, Newton's method, or conjugate gradient, require evaluating both the similarity measure and its gradient with respect to the rigid transformation parameters. Figure 2.6 displays example rigid registration results. In this example, the squares in (a) and (b) are the same size but the circle in (a) is slightly larger than the one in (b). Rigid registration creates a better alignment between the two images but its limitations are evident in its inability to fully align the circles as seen in (c). Similarly, in a torso or brain image, this limitation could result in a lack of alignment for all deformed soft-tissue regions.

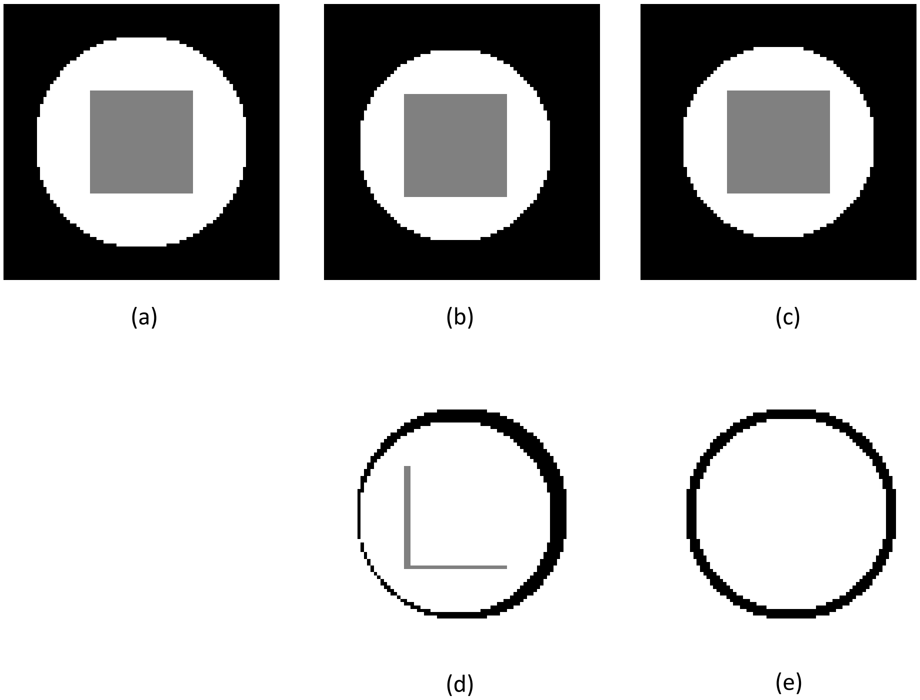


Figure 2.6: Sample rigid registration results: (a) reference image, (b) floating image containing a shifted square and a slightly smaller circle, (c) transformed floating image, (d) difference between floating and reference, (e) difference between transformed floating and reference.

Measure type	**Formula**
Sum of squared differences (SSD)	$\int_\Omega \left(R(\mathbf{x}) - F(\Phi(\mathbf{x})\right)^2 d\mathbf{x}$
Mean of squared differences (MSD)	$\frac{1}{\lvert\Omega\rvert} \int_\Omega \left(R(\mathbf{x}) - F(\Phi(\mathbf{x})\right)^2 d\mathbf{x}$
Squared correlation coefficient (SCC)	$\left(\frac{\mathrm{Cov}\left(R, F^\Phi\right)}{\sqrt{\mathrm{Var}(R)\mathrm{Var}(F^\Phi)}}\right)^2$, where $\mathrm{Var}\,(I) = \dfrac{1}{\lvert\Omega\rvert} \displaystyle\int_\Omega \left(I(\mathbf{x}) - \bar{I}\right)^2 d\mathbf{x},$ $\mathrm{Cov}\,(I_1, I_2) = \dfrac{1}{\lvert\Omega\rvert} \displaystyle\int_\Omega \left(I_1(\mathbf{x}) - \bar{I}_1\right)\left(I_2(\mathbf{x}) - \bar{I}_2\right) d\mathbf{x},.$ $\bar{I} = \dfrac{1}{\lvert\Omega\rvert} \displaystyle\int_\Omega I(\mathbf{x})\, d\mathbf{x}$
Mutual information (MI)	$H(F) + H(F(\Phi)) - H(R, F(\Phi))$ where $H(F) = -\int_{-\infty}^{\infty} p_F(a) \log p_F(a)\, da$ and $p_F(a)$ is the value of the probability density function of F at a [7]

Table 2.2: Common registration similarity measures: SSD, MSD, SCC, MI

2.3 Non-Rigid Registration

Non-rigid transformations consist of locally changing deformations that warp the floating image, stretching and compressing different regions to best align with the reference image. A sample nonrigid transformation is provided in Figure 2.7.

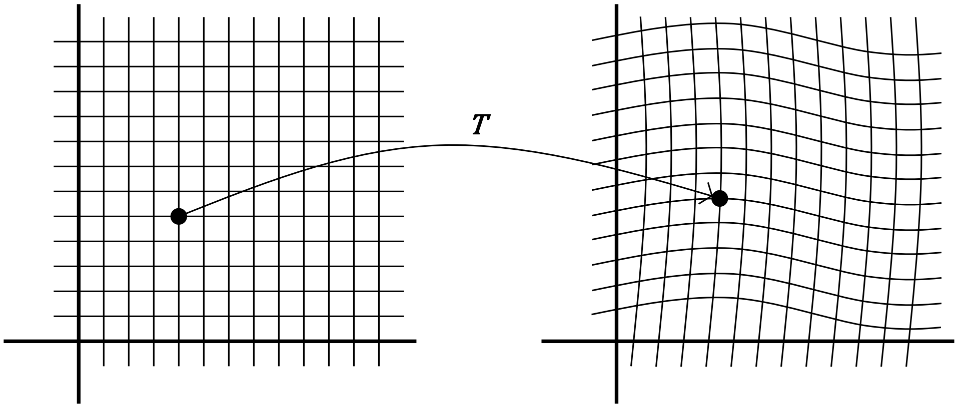


Figure 2.7: Deformable transformation

A non-rigid transformation can be written in the form $\Phi(\mathbf{x}) = \mathbf{x} - \mathbf{u}(\mathbf{x})$, where $\mathbf{u}$ is known as the displacement. The displacement can be written as $\mathbf{u}(\mathbf{x}) = [u_1(\mathbf{x}), ...u_m(\mathbf{x})]^\mathsf{T}$ where m is the number of dimensions in the original image, and where u_i provides the displacement at each point in the dimension i. Note that rigid transformations can be expressed using $\mathbf{u}(\mathbf{x}) = (\mathbf{I} - \mathbf{R})\mathbf{x} - \mathbf{t}$.

The optimality of the transformation is measured by an energy function composed of the similarity measure as well as an additional term known as the regularizer. The regularizer is in place to ensure that there is a smooth transformation and that the resulting minimization problem is well-posed. Like the similarity measure, there are many different common regularizers that can be used. Along with more complex constructed regularizers, the diffusion, elastic, and curvature regularizers are presented in [18] and compared in [7]. The energy function associated with a nonrigid transformation takes the following form:

$$\min_{\mathbf{u}} \qquad \mathcal{E}(R, F, \mathbf{u}) := \mathcal{E}_1(R, F(\Phi)) + \lambda \mathcal{E}_2(\mathbf{u}) \tag{2.7}$$

where $\mathcal{E}_1$ is the similarity measure that quantifies the dissimilarity between the reference image R and the deformed floating image $F(\Phi)$, $\mathcal{E}_2$ is the regularizer that ensures the minimization solution is smooth, and λ is a weighting parameter that controls the influence of the regularizer.

At a minimum of $\mathcal{E}(R, F, \mathbf{u})$, the derivative of the energy should be zero in any direction. Therefore, the following Euler-Lagrange equations are a necessary condition:

$$\mathcal{A}(\mathbf{u}(\mathbf{x})) = \alpha \mathbf{f}(\mathbf{x}; R, F(\Phi)) \tag{2.8}$$

where $\mathcal{A}$ is a partial differential operator and $\mathbf{f}$ is a force vector. These terms can be derived from the Gâteaux derivatives [12] of the similarity measure and the regularizer, respectively. We will only be observing the diffusion regularizer, given by:

$$\mathcal{E}_2(\mathbf{u}) = \int_\Omega \left(\sum_{i=1}^{3} \left\| \nabla u_i \left(\mathbf{x} \right) \right\|^2 \right) d\mathbf{x}, \tag{2.9}$$

because its Gâteaux derivative yields the negative Laplacian operator which produces the partial differential equation $-\Delta \mathbf{u}(\mathbf{x}) = \alpha \mathbf{f}(\mathbf{x}; R, F(\Phi))$.

Just as the case was for rigid registration, there are numerous optimization procedures that can be used to find the deformation that best minimizes the energy term. In theory, non-rigid registration algorithms are usually only intended to solve for displacements of up to one or two pixels; however, in practice this is usually pushed in hopes that it will also sufficiently solve for larger displacements. If pushed too far, a fold can be created in the displacement field. One method of avoiding this is to deform as much as possible without creating a fold, then resample the transformed image and start over as proposed in [9].

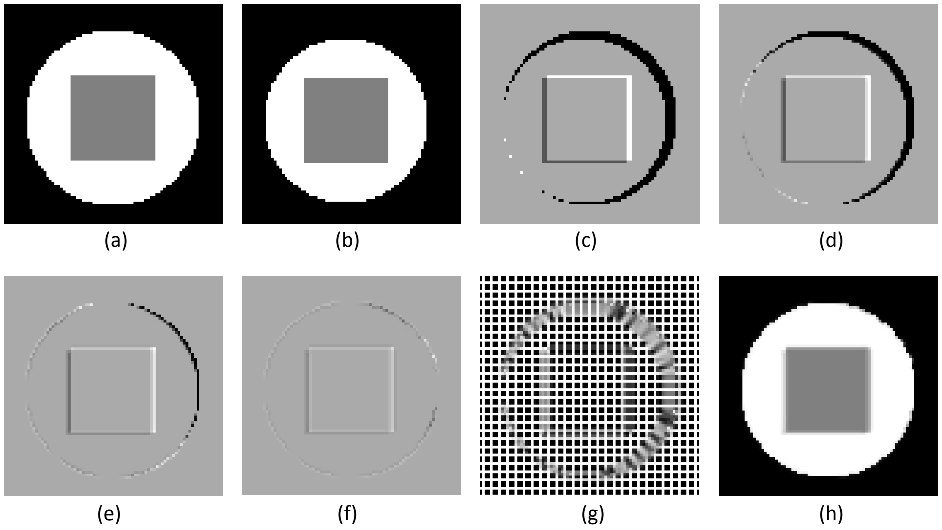


Figure 2.8: Sample non-rigid registration results over 5 iterations. (a) reference image, R; (b) floating image, F; (c) initial difference between R and F; (d) difference after 1^{st} iteration; (e) difference after 3^{rd} iteration; (f) difference after 5^{th} iteration; (g) deformation after 5^{th} iteration; (h) 5th iteration transformed F.

Figure 2.8 displays example non-rigid registration results for the toy example from Figure 2.6. In this example, the circle in (a) is larger than the one found in (b). Non-rigid registration stretches the smaller circle to map to the larger one but it also lessens the integrity of the center square. If the square is considered to be a rigid region then this would not be an acceptable registration. Similarly, when non-rigidly registering torso CT scans, this limitation would result in the potential manipulation of all bone structures in the image.

Chapter 3

Algorithm to simultaneously register rigid and nonrigid regions

When attempting to register two 3-D images of a single subject's torso, one could choose a nonrigid transformation to account for any adjustment to the location and size of organs. However, there are regions in the torso and head, such as the ribs and skull respectively, where the spatial relationship between the points should be preserved and a rigid transformation is desired. In this section, we propose a registration technique that enables a geometric transformation that is partitioned into rigid and nonrigid regions.

3.1 Approach

Under the assumption that bone regions will deform rigidly and all other regions will deform nonrigidly, the first step is to identify the bone regions. Bones are the most dense part of the torso, which means they have the greatest intensity in a CT image. In order to capture these regions, we will use the graph-cuts segmentation method discussed in [5] and described in Section 2.1.

An initial rigid registration will be performed using all of the image data in order to roughly align the two images. Then, the proposed new method will be used to perform a primarily nonrigid registration that respects the behavior of the rigid regions.

3.2 Formulation

This section will discuss how the proposed registration method is mathematically defined in a continuous space. Following this, the next chapter will discuss discretization and implementation details.

3.2.1 Preliminary steps

As previously mentioned, the first step required is the extraction of the bone regions in the floating image. This is performed using the exact process described in Section 2.1. Prior to registration, an initial coarse alignment is performed to ensure the data of interest occupy relatively the same space in each image. The necessity and measure of this alignment is performed purely based on manual observation of the data. Once the two images occupy the same relative space, they are ready to be registered.

3.2.2 Registration

Consider two images in $\mathbb{R}^3$, a reference image R and floating image F. Define a deformation $\Phi := \mathbb{R}^3 \mapsto \mathbb{R}^3$ by $\Phi(x) = \mathbf{R}_{\mathbf{\Theta}} x + \mathbf{t}_{\mathbf{\Theta}} - \mathbf{u}(x)$ where $\mathbf{R}_{\mathbf{\Theta}}$ is a rotation matrix according to the set of rigid parameters $\mathbf{\Theta}$, $\mathbf{t}_{\mathbf{\Theta}}$ is a translation according to $\mathbf{\Theta}$, and $\mathbf{u}$ is the displacement. The general registration problem is defined as:

$$\min_{\mathbf{\Theta},\mathbf{u}} \qquad \mathcal{E}(R, F, \mathbf{\Theta}, \mathbf{u}) := \mathcal{E}_1(R, F(\Phi)) + \lambda \mathcal{E}_2(\mathbf{u}) \tag{3.1}$$

where $\mathcal{E}_1$ is a similarity measure that quantifies the dissimilarity between the reference image R and the deformed floating image $F(\Phi)$, $\mathcal{E}_2$ is a regularizer that ensures the minimization solution is smooth, and λ is a weighting parameter.

The similarity measure that we will focus on is the sum of squared differences, which is defined by

$$\mathcal{E}_1(R, F(\Phi)) := \int_\Omega (R(\mathbf{x}) - F(\Phi(\mathbf{x}))^2 d\mathbf{x} \ . \tag{3.2}$$

We also choose the diffusion regularizer defined by

$$\mathcal{E}_2(\mathbf{u}) = \int_\Omega \left(\sum_{i=1}^{3} \|\nabla u_i\,(\mathbf{x})\|^2 \right) d\mathbf{x}. \tag{3.3}$$

Consider floating image F to be defined over a region $\Omega_F \subset \mathbb{R}^3$ while reference image R is defined over $\Omega_R \subset \mathbb{R}^3$. The region over which the similarity measure and regularizer are computed is the the intersection of these two regions, $\Omega = \Omega_R \cap \Omega_F$.

Consider Ω_F to be composed of two distinct regions, one of which corresponds to the regions that will transform rigidly under $F(\Phi)$ while the other corresponds to regions that will deform. Let $\Omega_{F'}$ represent the rigid regions and $\Omega_{F''}$ represent the non-rigid regions such that $\Omega_F = \Omega_{F'} \cup \Omega_{F''}$. The overlapping region between images can then be viewed as $\Omega = \Omega_1 \cup \Omega_2$ where Ω_1 and Ω_2 are defined as in Figure 3.1. The registration problem can be described over this newly defined region.

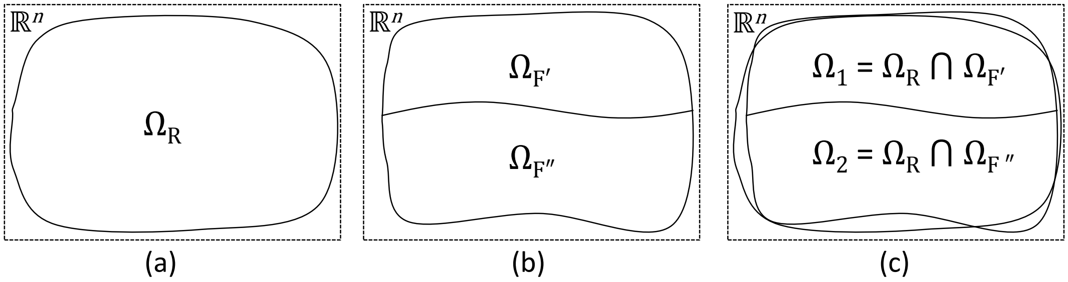


Figure 3.1: Regions over which reference image (a) and floating image (b) are defined, plus the intersection of these regions (c).

Given a domain Ω_a, we can refer to the interior of the domain as Ω_a^o and the boundary of the domain as $\partial\Omega_a$ such that $\Omega_a = \Omega_a^o \cup \partial\Omega_a$. Since $\mathbf{u}$ is a displacement, we can say that

$\mathbf{u}(x) = 0$ for $x \in \Omega_1$ and $x \in \partial\Omega_2$. Inserting this into (3.3) concludes that $\mathcal{E}_2(\mathbf{u}) = 0$ on Ω_1. Forcing a homogeneous Dirichlet boundary condition on $\partial\Omega_2$ allows for a piece-wise smooth transformation. Therefore, it can be stated such that

$$\mathcal{E}_2(\mathbf{u}) = \int_{\Omega} \left(\sum_{i=1}^{3} \left\| \nabla u_i \left(\mathbf{x} \right) \right\|^2 \right) d\mathbf{x} \quad = \quad \int_{\Omega_2} \left(\sum_{i=1}^{3} \left\| \nabla u_i \left(\mathbf{x} \right) \right\|^2 \right) d\mathbf{x} \tag{3.4}$$

In order to find a minimum energy it is critical to determine its derivatives with respect to each parameter. First, let us recall the set of rigid parameters $\boldsymbol{\Theta}$ as defined in (2.5) and the displacement $\mathbf{u}$ mentioned in Section 2.3. Now, we must find the gradient of the energy $\mathcal{E}$ with respect to $\boldsymbol{\Theta}$ and $\mathbf{u}$:

$$\nabla_{\boldsymbol{\Theta}} \mathcal{E}(R, F, \Phi) = \nabla_{\boldsymbol{\Theta}} \mathcal{E}_1(R, F(\Phi)) + \nabla_{\boldsymbol{\Theta}} \mathcal{E}_2(\mathbf{u}) \tag{3.5}$$

Since $\mathcal{E}_2$ does not depend on $\boldsymbol{\Theta}$, we know that $\nabla_{\boldsymbol{\Theta}} \mathcal{E}_2(\mathbf{u}) = 0$. Then, we can write:

$$\begin{aligned}
\nabla_{\boldsymbol{\Theta}} \mathcal{E}(R, F, \Phi) &= \nabla_{\boldsymbol{\Theta}} \mathcal{E}_1(R, F(\Phi)) && (3.6)\\
&= \nabla_{\boldsymbol{\Theta}} \int_{\Omega} \left(R(\mathbf{x}) - F\left(\mathbf{R}\mathbf{x} + \mathbf{t} - \mathbf{u} \right) \right)^2 d\mathbf{x} && (3.7)\\
&= \int_{\Omega} \nabla_{\boldsymbol{\Theta}} \left(R(\mathbf{x}) - F\left(\mathbf{R}\mathbf{x} + \mathbf{t} - \mathbf{u} \right) \right)^2 d\mathbf{x} && (3.8)\\
&= -\int_{\Omega} 2 \left(R(\mathbf{x}) - F\left(\Phi(\mathbf{x}) \right) \right) \nabla_{\boldsymbol{\Theta}} F\left(\Phi(\mathbf{x}) \right) d\mathbf{x}. && (3.9)
\end{aligned}$$

In order to determine the value of $\nabla_{\boldsymbol{\Theta}} F\left(\Phi(\mathbf{x}) \right)$, consider the partial derivative of F with respect to one variable $\Theta_i \in \boldsymbol{\Theta}$:

$$\begin{aligned}
\frac{\partial F\left(\Phi(\mathbf{x}) \right)}{\partial \Theta_i} &= \frac{\partial F}{\partial \Phi_1} \frac{\partial \Phi_1}{\partial \Theta_i} + \frac{\partial F}{\partial \Phi_2} \frac{\partial \Phi_2}{\partial \Theta_i} + \frac{\partial F}{\partial \Phi_3} \frac{\partial \Phi_3}{\partial \Theta_i}\\
&= \left[\nabla F\left(\Phi(\mathbf{x}) \right) \right]^{\mathsf{T}} \frac{\partial \Phi}{\partial \Theta_i}\\
&= \left(\frac{\partial \Phi}{\partial \Theta_i} \right)^{\mathsf{T}} \nabla F\left(\Phi(\mathbf{x}) \right),
\end{aligned}$$

and so, we can write:

$$\nabla_{\mathbf{\Theta}} F\left(\Phi(\mathbf{x})\right) = \left(\frac{\partial \Phi}{\partial \mathbf{\Theta}}\right)^{\mathsf{T}} \nabla F\left(\Phi(\mathbf{x})\right) \tag{3.10}$$

where

$$\left(\frac{\partial \Phi}{\partial \mathbf{\Theta}}\right)^{\mathsf{T}} = \mathbf{Q} = \begin{pmatrix} \frac{\partial \Phi_1}{\partial \Theta_1} & \frac{\partial \Phi_2}{\partial \Theta_1} & \frac{\partial \Phi_3}{\partial \Theta_1} \\ \frac{\partial \Phi_1}{\partial \Theta_2} & \frac{\partial \Phi_2}{\partial \Theta_2} & \frac{\partial \Phi_3}{\partial \Theta_2} \\ \frac{\partial \Phi_1}{\partial \Theta_3} & \frac{\partial \Phi_2}{\partial \Theta_3} & \frac{\partial \Phi_3}{\partial \Theta_3} \\ \frac{\partial \Phi_1}{\partial \Theta_4} & \frac{\partial \Phi_2}{\partial \Theta_4} & \frac{\partial \Phi_3}{\partial \Theta_4} \\ \frac{\partial \Phi_1}{\partial \Theta_5} & \frac{\partial \Phi_2}{\partial \Theta_5} & \frac{\partial \Phi_3}{\partial \Theta_5} \\ \frac{\partial \Phi_1}{\partial \Theta_6} & \frac{\partial \Phi_2}{\partial \Theta_6} & \frac{\partial \Phi_3}{\partial \Theta_6} \end{pmatrix}, \tag{3.11}$$

and each element of $\mathbf{Q}$ is computed the following way:

$$\begin{aligned} \frac{\partial \Phi_j}{\partial \Theta_k} &= \frac{\partial}{\partial \Theta_k} \left[\mathbf{R}\mathbf{x}\right]_j + \frac{\partial t_j}{\partial \Theta_k} && (3.12) \\ &= \frac{\partial}{\partial \Theta_k} \left[R_{j1}\mathbf{x}_1 + R_{j2}\mathbf{x}_2 + R_{j3}\mathbf{x}_3\right] + \frac{\partial t_j}{\partial \Theta_k} && (3.13) \\ &= \sum_{i=1}^{3} \frac{\partial}{\partial \Theta_k} R_{ji}\mathbf{x}_i + \frac{\partial t_j}{\partial \Theta_k} && (3.14) \\ &= \begin{cases} \sum_{i=1}^{3} \frac{\partial}{\partial \Theta_k} R_{ji}\mathbf{x}_i & , k <= 3 \\ \frac{\partial t_j}{\partial \Theta_k} & , k > 3. \end{cases} && (3.15) \end{aligned}$$

We can conclude that

$$\nabla_{\mathbf{\Theta}} \mathcal{E}(R, F, \Phi) = -\int_{\Omega} 2\left(R(\mathbf{x}) - F\left(\Phi(\mathbf{x})\right)\right) \mathbf{Q}\, \nabla F\left(\Phi(\mathbf{x})\right) d\mathbf{x}, \tag{3.16}$$

with $\mathbf{Q}$ given in 3.11.

In order to determine the gradient of $\mathcal{E}$ with respect to nonrigid displacement $\mathbf{u}$, consider the definition of the Gâteaux derivative in the direction of $\mathbf{w}$.

$$\begin{aligned}
d\mathcal{E}(R, F, \mathbf{u}; \mathbf{w}) &= \lim_{h\to 0} \frac{1}{h} \left[\mathcal{E}(R, F, \Phi(\mathbf{u} - h\mathbf{w})) - \mathcal{E}(R, F, \Phi(\mathbf{u}))\right] \qquad (3.17)\\
&= \lim_{h\to 0} \frac{1}{h} \left[\int_\Omega \mathcal{E}_1(R, F, \Phi(\mathbf{u} - h\mathbf{w})) + \lambda \mathcal{E}_2\left(\Phi(\mathbf{u} - h\mathbf{w})\right) d\mathbf{x} \right.\\
&\qquad \left. - \int_\Omega \mathcal{E}_1(R, F, \Phi(\mathbf{u})) + \lambda \mathcal{E}_2\left(\Phi(\mathbf{u})\right) d\mathbf{x}\right] \qquad (3.18)\\
&= \lim_{h\to 0} \frac{1}{h} \left[\int_\Omega \mathcal{E}_1(R, F, \Phi(\mathbf{u} - h\mathbf{w})) - \mathcal{E}_1(R, F, \Phi(\mathbf{u}))\, d\mathbf{x} \right.\\
&\qquad \left. + \lambda \int_{\Omega_2} \mathcal{E}_2\left(\Phi(\mathbf{u} - h\mathbf{w})\right) - \mathcal{E}_2\left(\Phi(\mathbf{u})\right) d\mathbf{x}\right] \qquad (3.19)\\
&= \int_\Omega \langle -2\left(R(\mathbf{x}) - F(\Phi(\mathbf{x}))\right) \nabla F(\Phi(\mathbf{x})), \mathbf{w}\rangle \, d\mathbf{x}\\
&\qquad + \lambda \lim_{h\to 0} \frac{1}{h} \int_{\Omega_2} \mathcal{E}_2\left(\Phi(\mathbf{u} - h\mathbf{w})\right) - \mathcal{E}_2\left(\Phi(\mathbf{u})\right) d\mathbf{x}. \qquad (3.20)
\end{aligned}$$

To simplify the second integral, consider:

$$\begin{aligned}
&\lim_{h\to 0} \frac{1}{h} \int_{\Omega_2} \mathcal{E}_2\left(\Phi(\mathbf{u} - h\mathbf{w})\right) - \mathcal{E}_2\left(\Phi(\mathbf{u})\right) d\mathbf{x} \qquad (3.21)\\
&= \lim_{h\to 0} \frac{1}{h} \int_{\Omega_2} \sum_i \left(\|\nabla(\mathbf{u}_i(\mathbf{x}) - h\mathbf{w}_i(\mathbf{x}))\|^2 - \|\nabla \mathbf{u}_i(\mathbf{x})\|^2\right) d\mathbf{x} \qquad (3.22)\\
&= \lim_{h\to 0} \frac{1}{h} \int_{\Omega_2} \sum_i \left(\|\nabla \mathbf{u}_i(\mathbf{x}) - h\nabla \mathbf{w}_i(\mathbf{x}))\|^2 - \|\nabla \mathbf{u}_i(\mathbf{x})\|^2\right) d\mathbf{x}. \qquad (3.23)
\end{aligned}$$

Note:

$$\begin{aligned}
\|\mathbf{a} - h\mathbf{b}\|^2 - \|\mathbf{a}\|^2 &= \left(\mathbf{a} - h\mathbf{b}\right)^\mathsf{T} \left(\mathbf{a} - h\mathbf{b}\right) - \mathbf{a}^\mathsf{T}\mathbf{a}\\
&= -h\mathbf{b}^\mathsf{T}\mathbf{a} - h\mathbf{a}^\mathsf{T}\mathbf{b} + h^2\mathbf{b}^\mathsf{T}\mathbf{b}\\
&= -2h\mathbf{a}^\mathsf{T}\mathbf{b} + h^2\mathbf{b}^\mathsf{T}\mathbf{b},
\end{aligned}$$

and so:

$$\begin{aligned}
&\lim_{h\to 0} \frac{1}{h} \int_{\Omega_2} \mathcal{E}_2\left(\Phi(\mathbf{u}-h\mathbf{w})\right) - \mathcal{E}_2\left(\Phi(\mathbf{u})\right) d\mathbf{x} && (3.24)\\
&= \lim_{h\to 0} \frac{1}{h} \int_{\Omega_2} \sum_i \left(\|\nabla \mathbf{u}_i(\mathbf{x}) - h\nabla \mathbf{w}_i(\mathbf{x}))\|^2 - \|\nabla \mathbf{u}_i(\mathbf{x})\|^2\right) d\mathbf{x} && (3.25)\\
&= \lim_{h\to 0} \frac{1}{h} \int_{\Omega_2} \sum_i \left(-2h\left(\nabla \mathbf{u}_i(\mathbf{x})\right)^{\mathsf{T}} \nabla \mathbf{w}_i(\mathbf{x}) + h^2 \|\nabla \mathbf{w}_i(\mathbf{x})\|^2\right) d\mathbf{x}. && (3.26)
\end{aligned}$$

Distributing the outer fraction, we find one term independent of h:

$$\begin{aligned}
&= \lim_{h\to 0} \int_{\Omega_2} \sum_i \left(-2\left(\nabla \mathbf{u}_i(\mathbf{x})\right)^{\mathsf{T}} \nabla \mathbf{w}_i(\mathbf{x}) + h\|\nabla \mathbf{w}_i(\mathbf{x})\|^2\right) d\mathbf{x} && (3.27)\\
&= \int_{\Omega_2} \sum_i -2\left(\nabla \mathbf{u}_i(\mathbf{x})\right)^{\mathsf{T}} \nabla \mathbf{w}_i(\mathbf{x}) d\mathbf{x} && (3.28)\\
&= -2\sum_i \int_{\Omega_2} \left(\nabla \mathbf{u}_i(\mathbf{x})\right)^{\mathsf{T}} \nabla \mathbf{w}_i(\mathbf{x}) d\mathbf{x}. && (3.29)
\end{aligned}$$

Using multi-dimensional integration by parts, (3.29) reduces to:

$$-2\sum_i \left(\int_{\partial\Omega_2} \mathbf{w}_i(\mathbf{x})\langle \nabla \mathbf{u}_i(\mathbf{x}), \mathbf{n}\rangle d\mathbf{s} - \int_{\Omega_2} \mathbf{w}_i \Delta \mathbf{u}_i(\mathbf{x}) d\mathbf{x}.\right) \tag{3.30}$$

By definition, $\mathbf{w}_i \to 0$ on $\partial\Omega_2$, so (3.30) reduces to

$$\begin{aligned}
&2\sum_i \int_{\Omega_2} \mathbf{w}_i(\mathbf{x}) \Delta \mathbf{u}_i(\mathbf{x}) d\mathbf{x} && (3.31)\\
&= 2\int_{\Omega_2} \sum_i \mathbf{w}_i(\mathbf{x}) \Delta \mathbf{u}_i(\mathbf{x}) d\mathbf{x} && (3.32)\\
&= 2\int_{\Omega_2} \left\langle \mathbf{w}(\mathbf{x}), \Delta \mathbf{u}(\mathbf{x})\right\rangle d\mathbf{x}. && (3.33)
\end{aligned}$$

Substituting (3.33) back into the derivation of $d\mathcal{E}$ yields:

$$
\begin{aligned}
& d\mathcal{E}(R, F, \mathbf{u}; \mathbf{w}) && (3.34)\\
&= \int_{\Omega} \langle -2\left(R(\mathbf{x}) - F(\Phi(\mathbf{x}))\right) \nabla F(\Phi(\mathbf{x})), \mathbf{w}\rangle \, d\mathbf{x} + \lambda 2 \int_{\Omega_2} \langle \mathbf{w}(\mathbf{x}), \Delta \mathbf{u}(\mathbf{x})\rangle \, d\mathbf{x} && (3.35)\\
&= \int_{\Omega} \langle -2\left(R(\mathbf{x}) - F(\Phi(\mathbf{x}))\right) \nabla F(\Phi(\mathbf{x})), \mathbf{w}\rangle \, d\mathbf{x} + \int_{\Omega_2} \langle 2\lambda \Delta \mathbf{u}(\mathbf{x}), \mathbf{w}(\mathbf{x})\rangle \, d\mathbf{x}. && (3.36)
\end{aligned}
$$

Since $\mathbf{u}(\mathbf{x}) = 0$ on $\partial\Omega_2, \Omega_1$, we can argue that the last term would not change if over all of Ω; i.e.,

$$
\begin{aligned}
& d\mathcal{E}(R, F, \mathbf{u}; \mathbf{w}) && (3.37)\\
&= \int_{\Omega} \langle -2\left(R(\mathbf{x}) - F(\Phi(\mathbf{x}))\right) \nabla F(\Phi(\mathbf{x})), \mathbf{w}\rangle \, d\mathbf{x} + \int_{\Omega} \langle 2\lambda \Delta \mathbf{u}(\mathbf{x}), \mathbf{w}(\mathbf{x})\rangle \, d\mathbf{x} && (3.38)\\
&= \int_{\Omega} \langle -2\left(R(\mathbf{x}) - F(\Phi(\mathbf{x}))\right) \nabla F(\Phi(\mathbf{x})) + 2\lambda \Delta \mathbf{u}(\mathbf{x}), \mathbf{w}(\mathbf{x})\rangle \, d\mathbf{x}. && (3.39)
\end{aligned}
$$

For $d\mathcal{E}$ to equal zero for any $\mathbf{w}(\mathbf{x})$, the other element of the inner product must equal zero, resulting in the following Euler-Lagrange equation:

$$
\begin{aligned}
-\lambda \Delta \mathbf{u}(\mathbf{x}) &= -\left(R(\mathbf{x}) - F(\Phi(\mathbf{x}))\right) \nabla F(\Phi(\mathbf{x})) \quad && x \in \Omega_2,\\
\mathbf{u}(\mathbf{x}) &= 0 && x \in \Omega \setminus \Omega_2^o.
\end{aligned}
\tag{3.40}
$$

In summary, we want to simultaneously register rigid and nonrigid regions by solving the energy minimization problem:

$$
\min_{\boldsymbol{\Theta}, \mathbf{u}} \qquad \mathcal{E}(R, F, \boldsymbol{\Theta}, \mathbf{u}) := \int_{\Omega} (R(\mathbf{x}) - F(\Phi(\mathbf{x}))^2 d\mathbf{x} + \lambda \int_{\Omega_2} \left(\sum_{i=1}^{3} \left\| \nabla u_i\left(\mathbf{x}\right) \right\|^2 \right) d\mathbf{x} \tag{3.41}
$$

over $\Omega = \Omega_R \cap \Omega_F$ where Ω_2 is the intersection of Ω_R with the nonrigid region of F, $\Omega_{F''}$. Necessary conditions for a minimum of $\mathcal{E}(R, F, \boldsymbol{\Theta}, \mathbf{u})$ are defined by the Euler-Lagrange equation (3.40).

Chapter 4

Practically Solving the Rigid-Nonrigid Registration Problem

Suppose we want to minimize a general function $f(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}$ using an initial estimate of the location of the minimum, $\mathbf{x}^{(k)}$. To find a better estimate of the minimum, we can perform gradient descent:

$$\frac{d\mathbf{x}(t)}{dt} = -\nabla f\left(\mathbf{x}(t)\right) \tag{4.1}$$

If we discretize in time and define a sequence of time points $t^{(k)} = t^{(0)} + k\tau$, then $x^{(k)}$ is defined as $x\left(t^{(k)}\right)$. Substituting these discretized terms yields the following equation:

$$\frac{\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}}{\tau} = -\nabla f\left(\mathbf{x}^{(k)}\right) \tag{4.2}$$

which can be manipulated to create the steepest-descent equation

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \tau\left(-\nabla f\left(\mathbf{x}^{(k)}\right)\right). \tag{4.3}$$

For the problem defined in (3.41) and (3.40), the gradient descent equations are

$$\mathbf{\Theta}'(t) = \frac{d\mathbf{\Theta}(t)}{dt} = 2\int_{\Omega} \left(R(\mathbf{x}) - F\left(\Phi(\mathbf{x})\right)\right)\mathbf{Q}\nabla F\left(\Phi(\mathbf{x})\right)d\mathbf{x} \tag{4.4}$$

$$\mathbf{u}_t(\mathbf{x}) = -\left(R(\mathbf{x}) - F(\Phi(\mathbf{x}))\right)\nabla F(\Phi(\mathbf{x})) + \lambda\Delta\mathbf{u}(\mathbf{x}). \tag{4.5}$$

Discretizing in time, explicitly for $\mathbf{\Theta}$ and semi-implicitly for $\mathbf{u}$, yields the steepest descent iteration:

$$\mathbf{\Theta}^{(k+1)} = \mathbf{\Theta}^{(k)} + \tau\left(2\int_{\Omega}\left(R(\mathbf{x}) - F\left(\Phi^{(k)}(\mathbf{x})\right)\right)\mathbf{Q}^{(k)}\nabla F\left(\Phi^{(k)}(\mathbf{x})\right)d\mathbf{x}\right) \tag{4.6}$$

$$\mathbf{u}^{(k+1)}(\mathbf{x}) = \mathbf{u}^{(k)}(\mathbf{x}) + \tau\left(-\left(R(\mathbf{x}) - F(\Phi^{(k)}(\mathbf{x}))\right)\nabla F(\Phi^{(k)}(\mathbf{x})) + \lambda\Delta\mathbf{u}^{(k+1)}(\mathbf{x})\right). \tag{4.7}$$

Note that (4.7) is actually a linear partial differential equation in the unknown displacement field $\mathbf{u}^{(k+1)}$; i.e.

$$\left[\mathbf{I} - \lambda\Delta\right]\mathbf{u}^{(k+1)}(\mathbf{x}) = \mathbf{u}^{(k)}(\mathbf{x}) - \tau\left(R(\mathbf{x}) - F(\Phi^{(k)}(\mathbf{x}))\right)\nabla F(\Phi^{(k)}(\mathbf{x})) \tag{4.8}$$

Using these formulas, we describe in Algorithm 1 a steepest-descent iteration for solving our proposed rigid-nonrigid registration algorithm.

The discrete updating formula for $\mathbf{\Theta}^{(k+1)}$ can easily be solved given the current estimates of all parameters. Updating the displacement field requires solving a linear PDE system that cannot easily be performed analytically; however, there are options for numerically approximating the solution to the PDE system. By using a centered difference approximation to the Laplacian operator, the PDE system is approximated by a linear system of equations that can be solved by Gauss-Seidel iteration or successive over-relaxation (SOR). Unfortunately, these methods can be extremely slow, often involving thousands or tens of thousands of iterations simply to approximate $\mathbf{u}^{(k+1)}$ given $\mathbf{u}^{(k)}$. Instead, we recommend multigrid [20], which scales linearly with the number of discrete grid points.

Algorithm 1 Steepest Descent Iteration for Registration

```
Set $\boldsymbol{\Theta}^{(0)} = 0$, $\mathbf{u}^{(0)} = 0$, $k = 0$; Select $\tau$ such that $0 < \tau << 1$.
repeat
  Solve for the updated rigid step
  $\boldsymbol{\Theta}^{(k+1)} = \boldsymbol{\Theta}^{(k)} + 2\tau \int_{\Omega} \Big( R(\mathbf{x}) - F\Big(\Phi\big(\mathbf{x}\,|\,\boldsymbol{\Theta}^{(k)}, \mathbf{u}^{(k)}\big)\Big)\Big) \mathbf{Q}^{(k)} \nabla F\Big(\Phi\big(\mathbf{x}\,|\,\boldsymbol{\Theta}^{(k)}, \mathbf{u}^{(k)}\big)\Big) d\mathbf{x}$
    for $\boldsymbol{\Theta}^{(k+1)}$.
  Solve the linear PDE system
  $[\mathbf{I} - \tau\lambda\Delta]\, \mathbf{u}^{(k+1)} = \mathbf{u}^{(k)} - \tau \Big( R(\mathbf{x}) - F\Big(\Phi\big(\mathbf{x}\,|\,\boldsymbol{\Theta}^{(k+1)}, \mathbf{u}^{(k)}\big)\Big)\Big) \nabla F\Big(\Phi\big(\mathbf{x}\,|\,\boldsymbol{\Theta}^{(k+1)}, \mathbf{u}^{(k)}\big)\Big)$
    for $\mathbf{u}^{(k+1)}$.
  Compute $\mathcal{E}^{(k+1)} = \mathcal{E}\Big(R, F, \Phi\big(\mathbf{x}\,|\,\boldsymbol{\Theta}^{(k+1)}, \mathbf{u}^{(k+1)}\big)\Big)$.
  Increment $k$.
until $|\mathcal{E}^{(k)} - \mathcal{E}^{(k-1)}| < \tau$.
```

This method solves for the updated $\mathbf{u}$ by constructing a sequence of grids at varying spatial resolutions and carrying out V, W, or full-multigrid cycles that propagate estimates of the solution across grid levels. [20]

It was observed that the change in displacement size could vary significantly between gradient descent iterations. Therefore, a constant step size τ was not very effective in regulating the additions to the displacement. Instead, a line search is performed at each iteration to determine the value of τ that minimized the energy function. While beneficial to the algorithm, this process is very time intensive and could definitely benefit from improvement in the future.

Chapter 5

Experiments and Results

While CT images of a torso were used for the majority of preliminary testing, we found that the nonrigid deformations required to accurately align torso images are quite large, typically beyond what might be considered to be within a realistic distance from the true minimum. The best results were instead found on skull MRIs in which the nonrigid components of the deformation are small.

The graph-cuts algorithm was used with six clusters to get a preliminary segmentation to identify the bone regions. With CT images on a torso, this sufficiently and automatically captures only bone regions; however, with MRIs of the skull some extra regions are erroneously identified as bone so they need to be manually removed from the segmentation.

In order to construct useful initial parameter estimates for registration, the two images were manually compared and a coarse initial alignment was performed to ensure the skulls occupied the same relative area. The pixel values of the images are also normalized to a range of $[0, 1]$ to ensure that our algorithm can be applied to a variety of types of medical images that are stored with values comprising different types or ranges of units.

An initial rigid registration was then performed using multi-resolution pyramids of the images. A pyramid is a set of images, or levels, of decreasing resolutions with the first level being the original image and each subsequent level being a blurred and subsampled version of the image in the previous level. Pyramids of the same size are constructed for both the floating and reference images. Registration is performed using the pyramids by first registering the images on the lowest resolution level. The resulting transformation is applied to the next level of the pyramid before that level is then registered. This is repeated for each level until the greatest resolution level is registered. The purpose of this method is to have an initial estimate to the registration, resulting in a smaller computation at the highest resolution level.

Here, rigid registration was performed on multi-resolution pyramids with 3 levels and the similarity measure between the images was minimized using the active-set method as implemented in MATLAB's fmincon function. Figure 5.1 displays a slice from the reference image as well as the slice of the same index in the floating image, captured both before and after rigid registration. After completing the initial rigid registration, it is time to run the steepest-descent algorithm. The following results were computed using a rigid step size of $\tau_{\mathbf{\Theta}} = 10^{-6}$ and a regularizer weight of $\lambda = 10^{-7}$ which were both chosen . The line search for the best nonrigid step size $\tau_{\mathbf{u}}$ searched the range of $[10^{-6}, 1]$ using a golden section search [2] with only two steps (for speed). In order for the rigid step size to affect all rigid parameters at a similar scale, the rotation angles were rescaled such that a rotation of 1 is equivalent to 0.125°.

The algorithm's termination criterion was initially chosen just to observe when the difference in energy dropped below a specified threshold. After testing, this was found not to be reliable. Now, a term is in place to measure a decaying amount of acceptable increase to the energy function. Given an initial acceptable increase value K, where $0 < K < 1$, in the energy function, the algorithm will terminate at any iteration n where the energy function is calculated to increase by more than K^n. This initial acceptable energy is determined manually based on preliminary tests. In our case, we chose $K = 1/4$.

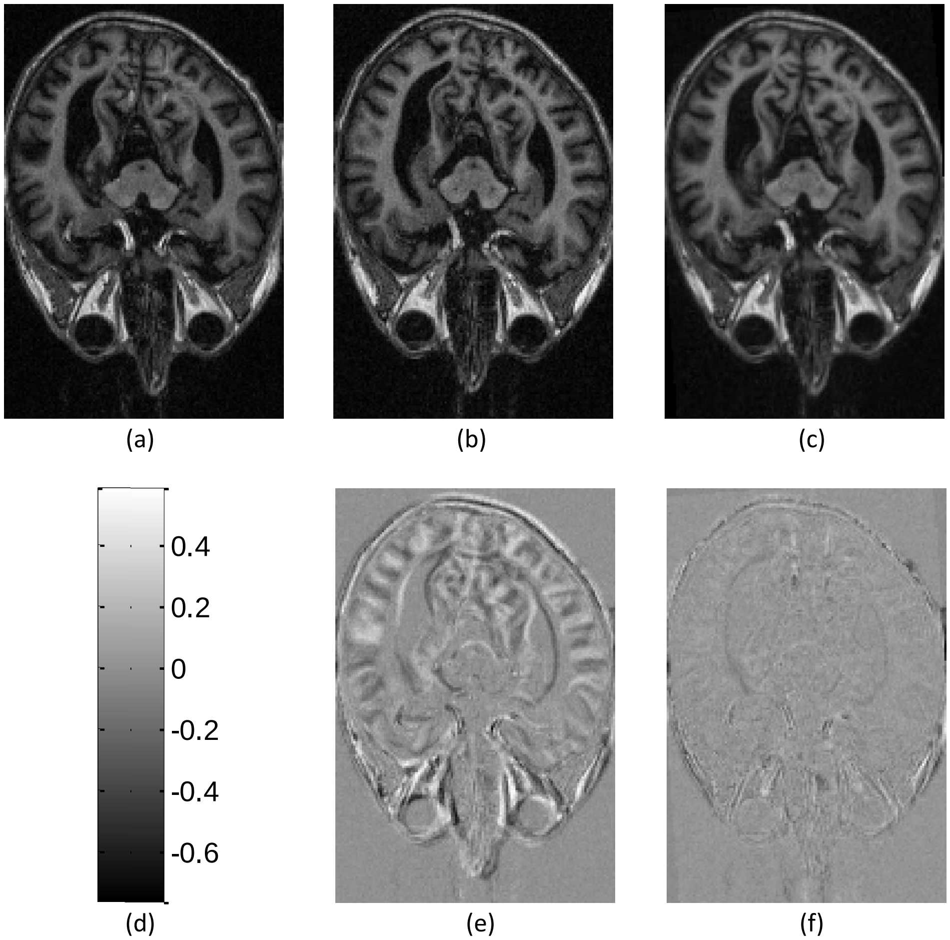


Figure 5.1: Sample slices from results of initial rigid registration (a) reference image, R; (b) floating image, F; (c) transformed floating image; (d) color bar for difference images; (e) difference between R and F; (f) difference between R and transformed F.

Figure 5.3 displays a sample of the result of the registration algorithm on the section of the image highlighted in Figure 5.2. It met the termination criteria and stopped after 5 iterations. Figure 5.4 compares this result to the results found when using explicitly rigid registration or nonrigid registration.

If we look at the difference images, there does not appear to be much difference between each method's performance; however, looking at the mesh in Figure 5.5 we can see a clear distortion of the rigid regions in the nonrigid registration result.

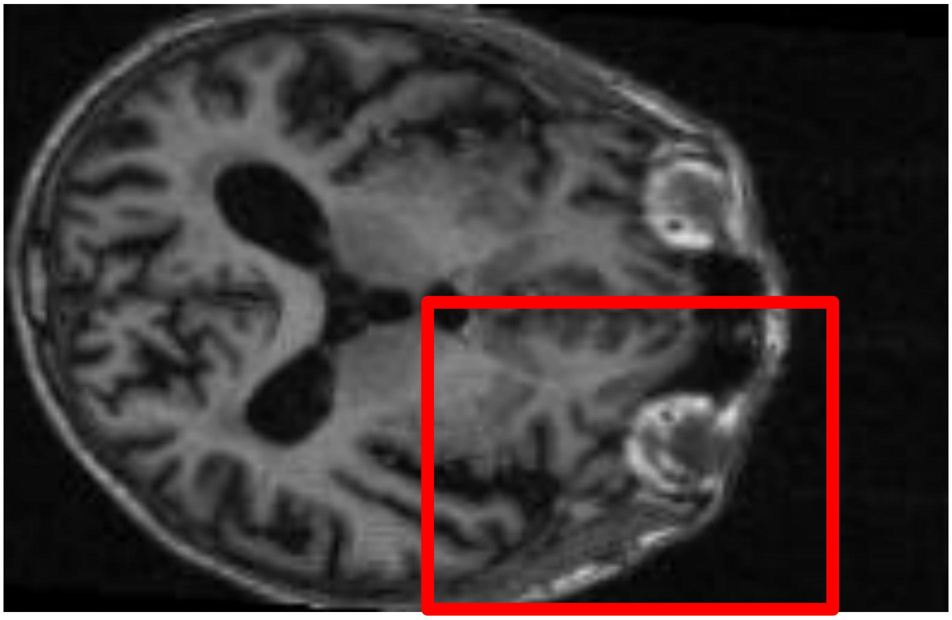


Figure 5.2: Portion of slice displayed to show algorithm results.

5.1 Limitations

While testing this algorithm, it was observed that the deformation field could easily capture small deformations (up to roughly 1.6 voxels), but when trying to deform further, other areas of the image would fold in on themselves. Therefore, the two images must initially be closely aligned in order to sufficiently capture the true deformation. Otherwise, the result can be reinterpolated and run through the algorithm multiple times to

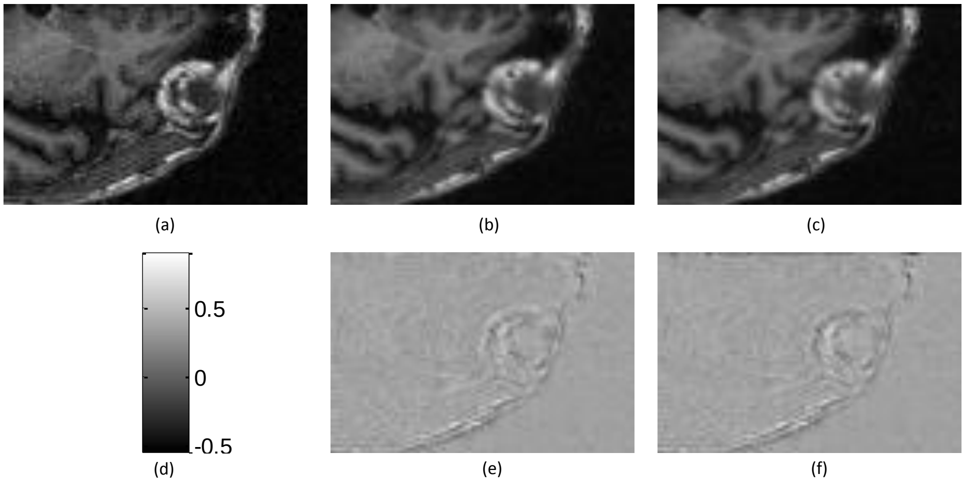


Figure 5.3: Sample slices from results of steepest descent registration: (a) reference image, R; (b) floating image, F; (c) transformed floating image; (d) color bar for difference images; (e) difference between R and F; (f) difference between R and transformed F.

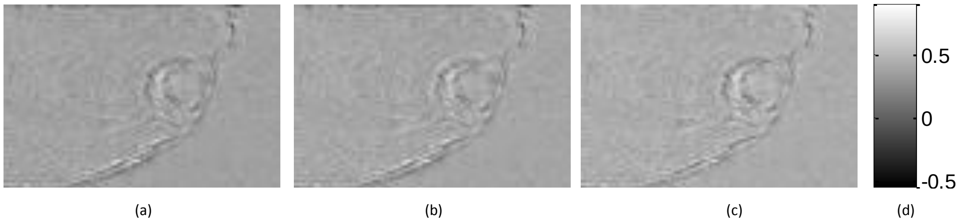


Figure 5.4: Comparison of sample slices from algorithm results versus strictly rigid or strictly nonrigid registration results: (a) difference between algorithm result and R; (b) difference between rigid result and R; (c) difference between nonrigid result and R; (d) color bar for difference images.

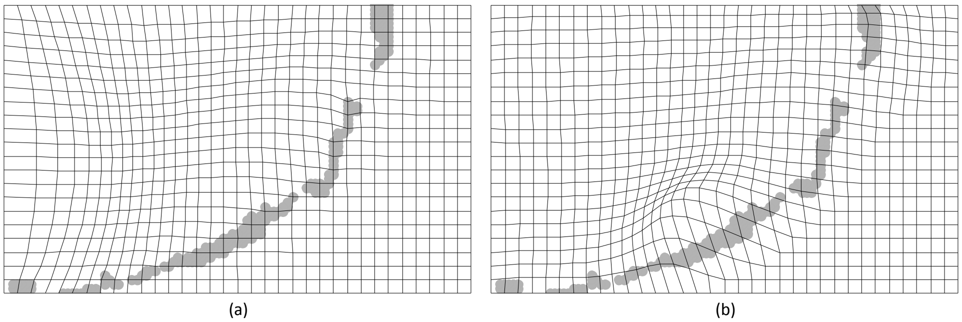


Figure 5.5: Deformation of sample slice after 1 iteration of (a) our algorithm and (b) exclusively nonrigid registration. The rigid regions are depicted in grey. Note that the rigid regions in (b) appear to be distorted in a nonrigid manner by the deformation field.

capture larger deformations. The downside to this is that every reinterpolation is slightly smoothing the floating image data, resulting in a loss of accuracy in representing the original floating image. The process of regridding is common practice in fluid registration [9]. The user will need to keep track of the desired deformation and the mentioned trade-off when running the algorithm.

No such limitation was found for the rigid transformation; however, a poor choice in step size may not converge or reach a useful resulting transformation.

5.2 Computational cost

For MRIs of size 128x205x161, the full algorithm took 17 hours on an HP 64-bit laptop with a 1.90GHz (AMD A4-3300M) processor running Matlab R2011b. This time includes segmentation, coarse rigid registration, and nonrigid registration with rigid regions. The majority of this time (64%) is spent on the main registration, particularly searching for the best deformation step size for each iteration.

Chapter 6

Conclusions and Future Work

This thesis proposed an automatic method for registering medical images that respects the rigid properties of the anatomical regions. Current medical image registration methods primarily use a nonrigid algorithm with application-specific customizations. The benefit of our proposed method is that it is not specific to one particular image subject or modality so there should be a large number of applications that could make use of its ability to respect bone regions.

6.1 Future Work

While this algorithm does provide sufficient results, there is always room for improvement to the existing work. To increase efficiency, the most beneficial thing to work on would be optimizing the search for the best displacement step size. It may be beneficial to explore more robust segmentation algorithms (for different types of images) that minimize the amount of post-processing required. It would be beneficial to find a different strategy to capture large deformations rather than regridding. These options are all seeds for more extensive research; however, for now, the defined algorithm is a sufficient starting place that offers results more respectful of anatomical regions than existing methods.

Bibliography

[1] Vincent Arsigny, Xavier Pennec, and Nicholas Ayache. Polyrigid and polyaffine transformations: a novel geometrical tool to deal with non-rigid deformations - application to the registration of histological slices. *Medical Image Analysis*, 9(6):507–523, December 2005.

[2] Mordecai Avriel and Douglass J. Wilde. Optimality proof for the symmetric fibonacci search technique. *Fibonacci Quarterly*, 4:265–269, 1966.

[3] Yuri Boykov and Vladimir Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1124–1137, September 2004.

[4] Yuri Boykov, Olga Veksler, and Ramin Zabih. Efficient approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(12):1222–1239, November 2001.

[5] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, November 2001.

[6] Morten Bro-Nielsen and Claus Gramkow. Fast fluid registration of medical images. In *Visualization in Biomedical Computing*, volume 1131 of *Lecture Notes in Computer Science*, pages 265–276. Springer Berlin Heidelberg, 1996.

[7] Nathan D. Cahill. *Constructing and Solving Variational Image Registration Problems.* PhD thesis, University of Oxford, 2009.

[8] T. F. Chan and L. A. Vese. Active contours without edges. *IEEE Transactions on Image Processing*, 10(2):266–277, February 2001.

[9] Gary E. Christensen, Richard D. Rabbitt, and Michael I. Miller. Deformable templates using large deformation kinematics. *IEEE Transactions on Image Processing*, 5(10):1435–1447, October 1996.

[10] George B. Dantzig and Mukund N. Thapa. *Linear Programming 1: Introduction.* Springer-Verlag New York, Inc., 1997.

[11] P. J. Edwards, D. L. G. Hill, J. A. Little, V. A. S. Sahni, and D. J. Hawkes. Medical image registration incorporating deformations. In *6th British Machine Vision Conference*, volume 2, pages 691–699, 1995.

[12] R. Gâteaux. Fonctions d'une infinité des variables indépendantes. *Bulletin de la Société Mathématique de France*, 47:70–96, 1919.

[13] Gene H. Golub and Charles F. Van Loan. *Matrix Computations.* Johns Hopkins University Press, 3 edition, 1996.

[14] Leo Grady. Random walks for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(11):1768–1783, November 2006.

[15] Vladimir Kolmogorov and Ramin Zabih. What energy functions can be minimized via graph cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):147–159, February 2004.

[16] J. A. Little, D. L. G. Hil, and D. J. Hawkes. Deformations incorporating rigid structures. In *Mathematical Methods in Biomedical Image Analysis, 1996., Proceedings of the Workshop on*, pages 104–113. IEEE, June 1996.

[17] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, pages 281–297. University of California Press, 1967.

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

[19] Francisco P.M. Oliveira and João Manuel R.S. Tavares. Medical image registration: a review. *Computer Methods in Biomechanics and Biomedical Engineering*, 17(2):73–93, 2014.

[20] U. Trottenber, C. W. Oosterlee, and A. Schüller. *Multigrid.* Academic Press, 1 edition, 2001.

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

[22] Olga Veksler and Andrew Delong. Gcoptimization. `http://vision.csd.uwo.ca/code/gco-v3.0.zip`, 2010.

[23] Roger P. Woods, Scott T. Grafton, Colin J. Holmes, Simon R. Cherry, and John C. Mazziotta. Automated image registration: I. general methods and intrasubject, intramodality validation. *Journal of Computer Assisted Tomography*, 22(1):139–152, January 1998.

[24] Barbara Zitová and Jan Flusser. Image registration methods: a survey. *Image and Vision Computing*, 21(11):977–1000, October 2003.