Spring 2017

Methods for evaluating image registration

Joo Hyun Song
University of Iowa

Recommended Citation
Song, Joo Hyun. "Methods for evaluating image registration." PhD (Doctor of Philosophy) thesis, University of Iowa, 2017.
https://doi.org/10.17077/etd.v0vailob

Follow this and additional works at: https://ir.uiowa.edu/etd
Part of the Electrical and Computer Engineering Commons

METHODS FOR EVALUATING IMAGE REGISTRATION

by

Joo Hyun Song

A thesis submitted in partial fulfillment of the
requirements for the Doctor of Philosophy
degree in Electrical and Computer Engineering
in the Graduate College of
The University of Iowa

May 2017

Thesis Supervisor: Professor Gary E. Christensen

Graduate College
The University of Iowa
Iowa City, Iowa

CERTIFICATE OF APPROVAL

PH.D. THESIS

This is to certify that the Ph.D. thesis of

Joo Hyun Song

has been approved by the Examining Committee for the thesis requirement for the Doctor of Philosophy degree in Electrical and Computer Engineering at the May 2017 graduation.

Thesis Committee: ______________________________
Gary E. Christensen, Thesis Supervisor

____
Oguz C. Durumeric

____
Hans J. Johnson

____
Jon G. Kuhl

____
Punam K. Saha

To my late father, who told me to never give up.

ACKNOWLEDGEMENTS

I would like to thank my parents and sister for their prayers, encouragements and sacrifices, without which I would have never been able to pull through till the end. I would especially like to thank my late father for believing in me and for continuously urging me to finish what I have started. Special thanks goes to my lovely wife for her gentle support and sometimes stern admonitions at times when I just wanted to give up.

I would like to thank my adviser Gary E. Christensen for his love and support throughout the years, especially for his patience with me in getting my work done. I would also like to thank professors Oguz C. Durumeric, Punam K. Saha, Joseph M. Reinhardt, Hans J. Johnson, and Jon G. Kuhl for sharing their time and knowledge to make this work happen. I would like to thank all the support and encouragements received from my labmates: Dinesh Kumar, Xiujuan Geng, Kunlin Cao, Cheng Zhang, Jeffrey Hawley, Jake Nickel, Amanda Pan, Wei Shao, Bowen Zhao, Ying Wei, James Harris, Nick Kiguta, Halim Choi, Ali Ghayoor, Matineh Shaker and Dakai Jin.

I would like to thank Tom Grabowski, David Rudrauf, Elizabeth K. Broeder, and Joel Bruss for their assistance in acquiring MRI image sets, segmenting the image sets and their help teaching me about neuroanatomy.

I would also like to thank all my friends for being their for me through thick and thin, and family at Believers in Grace for their words of encouragement and continuous prayers.

ABSTRACT

In the field of medical imaging, image registration methods are useful for many applications such as inter- and intra-subject morphological comparisons, creation of population atlases, delivery of precision therapies, etc. A user may want to know which is the most suitable registration algorithm that would work best for the intended application, but the vastness of medical image registration applications makes evaluation and comparison of image registration performance a non-trivial task. In general, evaluating image registration performance is not straightforward because in most image registration applications there is an absence of “Gold Standard” or ground truth correspondence map to compare against. It is therefore the primary goal of this thesis work to provide a means for recommending the most appropriate registration algorithm for a given task. One of the contributions of this thesis is to examine image registration algorithm performance at the component level. Another contribution of this thesis is to catalog the benefits and limitations of many of the most commonly used image registration evaluation approaches. One incremental contribution of this thesis was to demonstrate how existing evaluation methods can be applied in the midpoint coordinate system to evaluate some symmetric image registration algorithms such as the SyN registration algorithm. Finally, a major contribution of this thesis was to develop tools to evaluate and visualize 2D and 3D image registration shape collapse. This thesis demonstrates that many current diffeomorphic image registration algorithms suffer from the collapse problem, provides the first visualizations of the collapse problem in 3D for simple shapes and real human brain MR images, and provides the first experiments that demonstrate how adjusting image registration parameters can mitigate the collapse problem to some extent.

PUBLIC ABSTRACT

From ultrasound exams that an expectant mothers get before birth, to routine CT/MRI screenings during the course of our lives, medical imaging is an integral part of human life. Medical imaging has advanced to the point of not only visualizing human anatomy, but also being able to provide information for diagnosis and therapy. Image registration is a key component in medical imaging that allows mapping of anatomical structures between images. In this work, an assessment of current image registration techniques is made, and novel approaches for evaluating registration performance are introduced. The contributions made in this work will help improve image registration accuracy, which in turn will ultimately advance the research in diagnosis and therapy of diseases.

TABLE OF CONTENTS

CHAPTER

LIST OF TABLES

Table

LIST OF FIGURES

Figure

CHAPTER 1

INTRODUCTION

Image registration is the process of determining the geometrical transformation that maps point-wise correspondence from one image coordinate system to another. In the field of medical imaging, image registration methods are useful for many applications such as inter- and intra-subject morphological comparisons, creation of population atlases, delivery of precision therapies, etc. [18]. Rigid registration is a method in which the mapped transformation preserves all distance, straightness of lines (and the planarity of surfaces) and all nonzero angles between straight lines [28]. A rigid transformation is restricted to a few degrees of freedom, namely translation and rotation. In contrast, non-rigid image registration is more general and its transformation offers many degrees of freedom, but is more complex and computationally expensive to implement than rigid registration. In 1998, Maintz et al. in [46] presented a survey of publications concerning medical image registration techniques, summarizing various methods, dimensionality, nature of registration basis, nature and domain of the transformation, interaction, optimization procedure, modalities involved, subject, object, and validation schemes. The authors presented 18 classifications of head image registration applications alone, providing an estimate of the vastness of medical image registration applications. However, the vastness of medical image registration applications makes evaluating image registration performance a non-trivial task, and additional degrees of freedom and complexity makes evaluating non-rigid methods more challenging than rigid methods. In general, evaluating image registration performance is not straightforward because in most image registration applications there is an absence of “Gold Standard” or ground truth correspondence map to compare against. For instance, a feature present in one image may be missing in another, resulting in an unknown correspondence over that feature. But even if both images have the same number of features, it is still difficult to establish a perfect correspondence relationship at all scale levels.

While evaluating the performance of an image registration algorithm is challenging, it is yet more challenging to make comparisons between the performance of algorithms. A user may want to know which is the most suitable registration algorithm that would work best for the intended application. One of the biggest challenge in trying to compare the performance of one registration algorithm to another is that it is difficult to make unbiased one-to-one comparisons. More often than not, the data and methods used to evaluate the performance of one algorithm may not coincide with that of another algorithm. Unless the registrations are performed by those who are most experienced with the algorithms on a common dataset and evaluated with the same methods, there is no basis for saying that one algorithm is better than the other. Furthermore, every medical imaging application that requires image registration has different performance expectations and it may often be the case that a user may want an algorithm to perform well for a subset of evaluation criteria while its performance in other criteria is negligible. For example, an application that involves the propagation of a segmentation mask to a population to initialize a segmentation algorithm may only require approximate matching of the mask regions if the segmentation algorithm needs just an approximate initialization point to complete the task; but it may be the case that the user expects the algorithm to perform very swiftly to produce the approximate registrations. It is therefore the primary goal of this thesis work to provide a means for recommending the most appropriate registration algorithm for a given task.

1.1 History of Medical Image Regisration Evaluation

In this section, we briefly review some of the previous medical image registration evaluation studies.

One of the earliest large-scale evaluation work of image registration methods was the “Retrospective Image Registration and Evaluation” (RIRE) project led by Fitzpatrick of Vanderbilt University published [71], which involved 11 groups applying 15 different rigid image registration algorithms to selected registration tasks involving the registration of brain CT and/or PET to MR. All nine patient image data sets contained MR scans of three types (T1, PD and T2), with seven of the data sets containing CT scans and the other seven PET scans. The images were acquired with fiducial markers attached to the skull, and the “gold standard” transformations were computed using a fiducial-based registration method. Then the fiducial markers were removed from the images using an “air brushing” process and the resulting image data sets were made available to all project participants by means of FTP. The participants then submitted their registration results using a specified protocol of common coordinate system and format indicating the transformed positions of eight points. This project introduced three important image registration error measures. The first error measure is the target registration error (TRE). The TRE is defined as the Euclidean distance of each registered point. The target error TRE(x) is the true measure of registration error at the point x. Given a point in the template image, the TRE is defined as the Euclidean distance between the transformed location of the point and the location true corresponding point in the target image. Fitzpatrick et al. also introduced the notion of fiducial localization error (FLE) and fiducial registration error (FRE). The FLE is defined as the error between the computed center of a fiducial marker and the true center of a fiducial marker. The FRE is defined as the root-mean-square error in fiducial alignment between image space and physical space. FRE is easy to measure, but it is not as important as target registration error, TRE. The FRE is often used as a surrogate for the TRE. Fitzpatrick further proved that FRE and TRE are uncorrelated [27].

In 2007, Fitzpatrick et al. introduced version 2 of the RIRE project (RIRE2) [26]. In this ongoing study, investigators evaluate their non-rigid registration algorithms. Investigators download the evaluation images, register the multimodality images with their own algorithms and then submit their transformations for TRE evaluation.

Another notable early work conducted in a large scale on non-rigid image registration algorithms was the Vista project (INRIA-CNRS, Rennes) led by Christian Barillot [34], proposing an evaluation framework based on seven local and global measures of the relevance of the registration. In this work, six registration methods (five non-rigid and one rigid) were applied on a database of T1-weighted brain MR scans of 18 subjects. All participants downloaded the data and performed registrations without knowing the evaluation criteria, and submitted the resulting deformation fields for evaluation. Registration performance evaluation was focused on the matching of cortical regions since the methods are dedicated to the anatomical and functional normalization. Anatomically meaningful features were extracted from all MR image volumes for evaluation purposes, but the features were selected so that they were not related to the "forces" used to drive the registration process. Global evaluation measures used to evaluate registration performance were average volume (of registered deformed 17 subjects), overlap of gray and white matter tissues, correlation of Lvv (to distinguish between sulci and gyri) and consistency of the deformation field (using Jacobian of transformation). The local evaluation measures used were distance between registered sulci (which were segmented and compared visually first) and statistical study of deformed shapes (i.e., registered sulci). ANOVA tests were conducted on some of the measures to compare intergroup and intragroup variances for statistical significance of the measures. The authors questioned the impact of reference subject choice on the results, but cited studies that showed that this choice has minimal influence on the results. Some of the challenges faced in this work included the lack of feasibility to compare the methods on the basis of computation time, sensitivity to parameters, difficulty to implement, etc. because of independent registration processes by the participants, and resource management and transfer of large transformation data.

More recently, there has been a steady increase of image registration evaluation/comparison works being conducted. In 2009, Yassa et al. [76] evaluated the performance of cross-participant registration techniques for MRI studies of the medial temporal lobe. T1-weighted brain MR scans of 20 patients were used to evaluate the registration performance of 12 non-rigid image registration algorithms. The registrations were performed by the author on either manually labeled brain regions, weighted masks for these regions, or original unlabeled brains. The registration results were evaluated using two methods: overlap (target overlap) scores of registered manual segmentations of subregions of the MTL (hippocampus, parahippocampal cortex, perirhinal cortex and entorhinal cortex), and FWHM smoothness of the average warped structural scans produced by each method (based on the concept that misalignment due to tissue type mismatch resulting from registration errors leads to a blurry mean image). Additionally, the performance of diffeomorphic methods (DARTEL, Demons and LDDMM) at aligning subfields of the hippocampus in a subset of scans using the overlap score method. The authors made note of the computation speed of each algorithm, mentioning some semi-automatic procedures for some algorithms, and that LDDMM-based methods required software and hardware that are currently only available to very few sites (JHU-CIS being one of them).

In 2009, Klein et al. [41, 42] evaluated 14 non-rigid and one rigid image registration algorithms on 80 manually labeled T1-weighted brain MRIs. The image data sets were acquired from four different sources (LPBA40, IBSR18, CUMC12 and MGH10). The author acquired the registration software from their respective authors and performed over 45,000 registrations locally. The project employed eight different error measures to evaluate algorithms: target, mean and union volume overlap measures (over all labeled regions), target, mean and union surface overlap measures (of surfaces generated by Freesurfer), volume similarity (over all labeled regions) and distance error (of labeled region boundaries). In addition, two different statistical tests (permutation and one-way ANOVA) were performed to get around the issue of non-independence of samples, and an indifference-zone ranking was performed to highlight practical, rather than statistical significance of the measures. The authors noted the average runtime, degrees of freedom and year developed for each algorithm. Some of the difficulties encountered in this work include the challenges in learning how to run each algorithm, handling different data formats, and there remains the question of whether the algorithms performed optimally with default parameters. Additionally, the author notes that while the evaluation measures relying on information which is not directly included in the images may be good for evaluating the registrations, they do not inform us about the intrinsic properties of the spatial transformations. A similar work on volume-based and surface-based brain image registration methods was published in 2010 [42].

The EMPIRE10 (Evaluation of Methods for Pulmonary Image REgistration 2010) challenge was organized in conjunction with the Grand Challenge workshop at MICCAI 2010 [51, 52]. This competition provided a public platform for comparison of registration algorithms applied to thoracic CT data. A set of 30 thoracic CT intra-patient scan pairs were selected to represent a broad variety as possible of the scenarios encountered in clinical practice. Variations of the thoracic scan pairs consisted of breathhold inspiration pairs, breathhold inspiration and expiration pairs, 4D data pairs, ovine data pairs, contrast/non-contrast pairs, and artificially warped pairs. The challenge was divided into two phases in which participants downloaded 20 of the 30 pairs of thoracic CT scans, performed registrations and submitted the transformations to the organizers for the first phase, and participants took part in a 3-hour workshop to register the remaining 10 pairs and submit their results for the second phase. The performance of 20 non-rigid image registration algorithms were evaluated using four evaluation categories: alignment of lung boundaries, alignment of major fissures, correspondence of annotated landmark pairs, and singularities in the deformation field. A ranking system had been devised to measure a participant's overall performance and to compare participants with each other. Additional information such as automation of the registration process, transformation model, similarity measures, computational resource, etc. were collected and the results and ranking have been made available on the EMPIRE10 website `empire10.isi.uu.nl`.

Other related evaluation works include [40], [32], and a formal model for defining and reporting evaluation and validation protocols in medical image processing was proposed by Jannin et al. [37]. Overall, all of the major evaluation works mentioned above involved registering common image data sets using various image registration algorithms and evaluating the resulting transformations using a set of evaluation methods on common evaluation data. Some works involved the principal evaluator performing all tasks including the registrations, while others involved participants applying their own algorithms on the downloaded image data set and submitting the transformations for evaluation. The latter evaluation scheme is preferred because the algorithm should be configured by those who are thoroughly familiar with all aspects of its behavior in order to obtain optimal performance [52]. None of the works provided a common tool that participants could use to evaluate image registration performance so that further alterations could be made if necessary to make improvements to the registration results before reporting.

1.2 Dissertation Overview

The purpose of this dissertation is to provide a framework for evaluating non-rigid image registration algorithm performance. This dissertation is divided into three main chapters. Chapters 2 and 3 provide the mathematical formulation of many of the most common image registration algorithms and image registration evaluation methods, respectively. Chapter 4 provides practical experiments that puts the theory into practice. We now preview each chapter in more detail.

- **Chapter 2:** This chapter describes the major components of an image registration algorithm and provides the foundation for understanding how each component affects image registration performance. Almost any image registration algorithm can be constructed by combining the algorithmic components described in this chapter. This chapter is divided into three parts. The first part describes some of the most common transformation models and discusses their pros and cons. Specifically, we discuss how the number of degrees of freedom of each transformation model impacts how well one image can be deformed

Table 1.1. Summary of medical image registration evaluation studies.

Study	Ongoing	Who did registration	Who did evaluation	Type of DB used	Evaluation methods	Type of transformations evaluated	Anatomical region
RIRE[71]	Y	Participants	Organizers	CT, PET, T1/T2/PD MRI	Fiducials (TRE)	Rigid	Human Brain
RIRE2[26]	Y	Participants	Organizers	CT, PET, T1/T2/PD MRI	Fiducials (TRE)	Non-rigid, small deformation	Human Brain
Vista[34]	N	Participants	Organizers	T1 MRI	Avg Vol, Overlap, Correlation of Lvv, Jacobian of transformation	Rigid & Non-rigid, small deformation	Human Brain
NIREP[18, 70, 59]	Y	Participants	Participants	T1 MRI	Non-rigid, small and large deformation	SSD, Overlap, Jacobian of transformation, ICE (Expandable)	Human Brain (Customizable)

into the shape of another. The second part of this chapter describes some of the most common similarity cost functions. The similarity cost function determines which features and how these features are used for registration, e.g., landmarks, contours, surfaces or intensities. The final part of this chapter describes some of the most common regularization cost functions. A regularization cost function is used to impose desired properties on the transformation model such as smoothness or continuum mechanical behaviors like that of a linear elastic material or viscous fluid.

- **Chapter 3:** This chapter describes many of the major evaluation methods used for evaluating image registration performance. It also describes how these methods can be applied in the midpoint coordinate system to evaluate the performance of image registration algorithms that match two images in the midpoint coordinate system. This chapter discusses the shape collapse problem that many state of the art diffeomorphic image registration algorithms suffer from. This chapter provides new methods for predicting, detecting and visualizing 2D and 3D shape collapse.
- **Chapter 4:** This chapter provides experimental evaluation of some of the most common image registration algorithms. This chapter is not intended to be an exhaustive comparison between image registration algorithms, but rather as a demonstration of the image registration framework described previously in this dissertation. The last part of this chapter provides 2D and 3D experiments that demonstrate the shape collapse that may occur when registering binary images

and real 3D MRI brain images. These experiments are the first to visualize 3D shape collapse. This chapter also discusses how modifying image registration algorithm parameters can, to some extent, mitigate the shape collapse problem.

- **Chapter 5:** This chapter summarizes the contributions of this dissertation. One of the major contributions of this dissertation is the work on the 2D and 3D shape collapse problem.

CHAPTER 2

IMAGE REGISTRATION

In general, an image registration algorithm is composed of four major parts: transformation model, similarity cost function, regularization cost function and optimization method. Different registration algorithms can be synthesized by combining different combinations of these components. In this thesis, we seek to understand what components of the registration algorithm affect the quality of the registration. In most cases, the optimization method only impacts the speed of the computation and not the quality of the registration. However, this is not always true such as in the case of stochastic optimization methods. In stochastic optimization methods the optimization approach is a significant part of the algorithm since it may avoid some of the problems with getting stuck in local minima. Not with standing this issue, we chose to focus this thesis on the performance implications of the transformation model, cost function and regularization method and not on the optimization method.

Another practical reason that we did not focus on evaluating the impact of different optimization methods is that this would have required us to implement each algorithm using different optimization methods. This is clearly beyond the scope of this work.

This chapter is organized into three sections. The first section reviews some of the common small and large deformation transformation models, respectively. We use the continuum mechanics definitions of small and large deformations. A small deformation transformation is a transformation in which local strains and rotations are both small. On the other hand, a large deformation transformation is a transformation in which local strains and rotations are both large. The second section of this chapter reviews some of the common similarity cost functions. Finally, the chapter ends with a discussion of the some common regularization cost functions.

We now define some common terms that will be used for the remainder of this thesis. Let $I_1 : \Omega \rightarrow \mathbb{R}^d$ and $I_2 : \Omega \rightarrow \mathbb{R}^d$ be a pair of images, where d indicates the number of components of the image (i.e., $d = 1$ for scalar-valued images, $d = 3$ for RGB color images, etc.). Let $I_1(\mathbf{x})$ be the moving (or template) image, whose coordinate system $\mathbf{x} \in \Omega$ establishes the coordinate-intensity relationship of the image, where $\Omega \subseteq \mathbb{R}^n$ is the image domain ($n = 2$ for 2D and $n = 3$ in 3D space). Similarly, let $I_2(\mathbf{y})$ be the fixed (or target) image, with coordinate system $\mathbf{y} \in \Omega$. The fixed image coordinate system will also be referred to as the reference image coordinate system. Define $\varphi : \Omega \rightarrow \Omega$ as the push-forward (or Lagrangian) transformation that transforms a point $\mathbf{x}$ in the moving image coordinate system to a point $\mathbf{y}$ in the fixed image coordinate system. The moving image is transformed into the target image coordinate system by the action of transformation φ applied to the image I_1. We define the action of φ on I_1 as $\varphi \cdot I_1 \triangleq I_1 \circ \varphi^{-1} = I_1(\varphi^{-1})$ where $\circ$ denotes the composition of two functions. The inverse transformation φ^{-1} is called the pullback transformation and is often denoted as h in this work.

2.1 Transformation Models

This section describes some of the most common small and large deformation transformation models. A small transformation model is a transformation that has small local strains and small local rotations where as, a large transformation model is characterized by large local strains and large local rotations. Unlike large deformations, small deformations are not able to register images that require curved trajectories such as deforming a patch to a C-shape or a circle to a C-shape[12]. Small deformation transformation models are useful because many shapes that are registered in medical imaging only differ by small deformations. Small deformation models are often simple, i.e., have a small number of degrees of freedom. As a result small deformation algorithms often can be implemented efficiently and some can even be computed in real-time.

This section progresses from simple small deformation transformation models to more complex large deformation models. One should assume that a transformation model is a small deformation model unless otherwise stated.

2.1.1 Rigid Transformation Model

A rigid transformation is a geometric transformation that preserves the distance relationship between every pair of points in the coordinate system. Rigid transformation allows for three kinds of transformations - translation, rotation and reflection. There are many different ways to write the equation for a rigid transformation. One typically way to write a push forward rigid transformation $\varphi : \Omega \to \Omega$ is

$$\varphi(x) = Rx + b, \tag{2.1}$$

where R is the rotation matrix and $\mathbf{b}$ the translation vector. Alternatively, a rigid transformation is often written as a pullback transformation, i.e.,

$$h(y) = \varphi^{-1}(y) = R^{-1}(y - b) = \tilde{R}y + \tilde{b}, \tag{2.2}$$

where $\tilde{R} = R^{-1}$ is a rotation matrix and $\tilde{b} = -R^{-1}b$ is a translation vector.

To simplify things, we will work with the push forward transformation. In 2D, R may be written as

$$R = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}, \tag{2.3}$$

where θ is the rotation angle about the origin. In 3D, there are three axes of rotation. Thus, one possible parametrization of a rotation matrix is $R = R_z(\gamma)R_y(\beta)R_x(\alpha)$, where

$$\begin{aligned} R_x &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix} \\ R_y &= \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix} \\ R_z &= \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{2.4}$$

are rotation matrices along the x, y and z-axis with rotation angles α, β and γ, respectively. Note that the order of rotation (in other words, the order of matrix multiplication) is important. Changing the order rotation gives a different parametrization of the rotation matrix.

A disadvantage of writing a rotation matrix as a product of three rotation matrices is that it suffers from the problem of gimbal lock. Gimbal lock is the loss of one degree of freedom in a three-dimensional rotational system. It occurs when the axes of two of the three axes of rotation are driven into a parallel configuration, “locking” the system into rotation in a degenerate two-dimensional space (https://en.wikipedia.org/wiki/Gimbal_lock).

Parameterizing the rotation matrix in terms of quaternions is one way to avoid gimbal lock. A quaternion can be defined in a similar manner to a complex number, i.e., $q = a + b\mathbf{i} + c\mathbf{j} + d\mathbf{k}$ where the symbols $\mathbf{i}$, $\mathbf{j}$, and $\mathbf{k}$ satisfy the relationships $\mathbf{j}^2 = \mathbf{k}^2 = \mathbf{i}^2 = \mathbf{ijk} = -1$. One possible quaternion parametrization of a 3D rotation matrix is given by

$$R(q) = \begin{bmatrix} 1-2c^2-2d^2 & 2(bc-da) & 2(bd+ca) \\ 2(bc+da) & 1-2b^2-2d^2 & 2(cd-ba) \\ 2(bd-ca) & 2(cd+ba) & 1-2b^2-2c^2 \end{bmatrix}, \tag{2.5}$$

where $q = a + b\mathbf{i} + c\mathbf{j} + d\mathbf{k}$ is a unit quaternion.

Equations 2.1 and 2.2 describe rigid rotations about the origin. In order to rotate about the point p in the domain of the transformation, one must first translate the center of rotation p to the origin, apply the rotation and then translated back. For example, transforming Eq. 2.1 to rotate about the point p gives the equation

$$\varphi(x) = R(x-p) + b + p. \tag{2.6}$$

Notice that Eqs. 2.1 and 2.6 are equal when the center of rotation $p = \mathbf{0}$.

Rigid transformations are useful for registering intra-patient data scanned with the same scanner with same spatial settings, since the type of motion introduced for such a dataset usually only involves translation and rotation. Due to the small number of parameters, rigid registration algorithms are extremely fast to compute and are often used for real-time registration. Rigid transformation is customarily used to align the image pair (e.g., alignment of AC-PC points in brain MR image pair) prior to non-rigid image registration.

2.1.2 Affine Transformation Model

An affine transformation has more degrees of freedom than a rigid transformation. An affine transformation allows for scaling, shearing, rotation and translation. In fact, rigid transformation is a special case of the more general affine transformation with restrictions on the distances. A 3D push forward affine transformation can be written in the form

$$\varphi(x) = Ax + b, \tag{2.7}$$

where A is a 3×3 matrix with nonzero determinant and b is a 3×1 translation vector. This transformation can also be rewritten as

$$\begin{bmatrix}\varphi(x)\\ 1\end{bmatrix} = \left[\begin{array}{c|c} A & b \\ 0\dots0 & 1\end{array}\right]\begin{bmatrix}x\\ 1\end{bmatrix}. \tag{2.8}$$

2.1.3 N-th Order Polynomial Transformation Model

The Automatic Image Registration (AIR) by Woods et al. [72, 75, 73, 74] uses an N-th order polynomial transformation model. A polynomial in m variables $x_1, \dots, x_m$ with real coefficients can be defined in the following way. Let α denote a multi-index, i.e., α is defined as $\alpha = (\alpha_1, \dots, \alpha_m)$, where each α_i is a non-negative integer. A monomial with multi-index α is defined as

$$x^\alpha = \prod_{i=1}^{m} x_i^{\alpha_i} = x_1^{\alpha_1} \cdots x_m^{\alpha_m}. \tag{2.9}$$

We define the degree of a monomial, denoted as $|\alpha|$, as $|\alpha| = \sum_{i=1}^{m} \alpha_i$. A polynomial in m variables is defined as a finite linear combination of monomials with real coefficients

$$p = \sum_{\alpha} p_\alpha x^\alpha, \tag{2.10}$$

where $p_\alpha = p_{\alpha_1,\dots,\alpha_m} \in \mathbb{R}$, and only finitely many coefficients p_α are nonzero.[1] The degree of a polynomial p is the largest degree of a monomial occurring with non-zero coefficient in the expansion of p.

An N-th order polynomial pullback transformation $h : \mathbb{R}^3 \to \mathbb{R}^3$ is defined as

$$h(y) = \sum_{0 \leq |\alpha| \leq N} \begin{bmatrix} p_\alpha \\ q_\alpha \\ r_\alpha \end{bmatrix} y^\alpha \tag{2.11}$$

where $p_\alpha, q_\alpha, r_\alpha \in \mathbb{R}$.

Using the stars and bars algorithm[2], the degrees of freedom of an N-th order polynomial p in m dimensional space is $\mathrm{dof}(p) = m\binom{m+N}{m}$. Thus, the degrees of freedom of the transformation h given in Eq. 2.11 is $\mathrm{dof}(h) = m\binom{m+N}{m}$.

2.1.4 B-spline Transformation Model

Rueckert et al. [55] was the first group to use a cubic b-spline transformation model in the medical imaging community. An example of a b-spline transformation model is given by

$$h(y) = y + \sum_{l=0}^{3}\sum_{m=0}^{3}\sum_{n=0}^{3} B_l(u)B_m(v)B_n(w)a_{i+l,j_m,k+n} \tag{2.12}$$

where $i = \lfloor y_1/\delta_1 \rfloor - 1$, $j = \lfloor y_2/\delta_2 \rfloor - 1$, $k = \lfloor y_3/\delta_3 \rfloor - 1$, $u = y_1/\delta_1 - (i+1)$, $v = y_2/\delta_2 - (j+1)$, $u = y_3/\delta_3 - (k+1)$, and $a_{i+l,j_m,k+n} \in \mathbb{R}^3$ are the parameters. The

[1] For example, a 5^{th} order polynomial $p = 5x_1x_2^2x_3 + 3x_1^2x_2^2x_3 + x_1^5$ can be expressed as $p = p_{(1,2,1)}x^{(1,2,1)} + p_{(2,2,1)}x^{(2,2,1)} + p_{(5,0,0)}x^{(5,0,0)}$, where $p_{(1,2,1)} = 5$, $p_{(2,2,1)} = 3$, and $p_{(5,0,0)} = 1$.

[2] https://en.wikipedia.org/wiki/Stars_and_bars_(combinatorics)

B-spline basis functions $B_0(t)$ through $B_3(t)$ are shown in Fig. 2.1 and defined as

$$\begin{aligned} B_0(t) &= (-t^3 + 3t^2 - 3t + 1)/6, \\ B_1(t) &= (3t^3 - 6t^2 + 4)/6, \\ B_2(t) &= (-3t^3 + 3t^2 + 3t + 1)/6, \\ B_3(t) &= t^3/6 \qquad (2.13) \end{aligned}$$

for $0 \leq t \leq 1$.

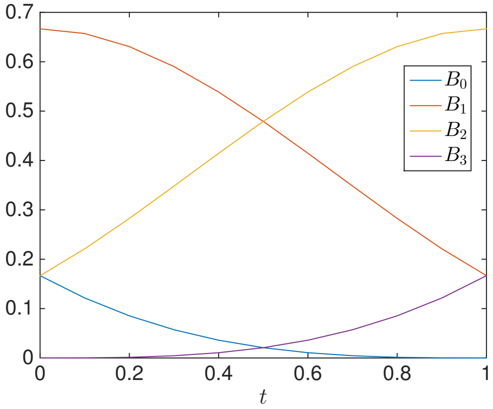


Figure 2.1. Cubic B-spline basis functions $B_0(t)$, $B_1(t)$, $B_2(t)$ and $B_3(t)$.

The cubic b-spline parametrization given in Eq. 2.12 is not convenient for computing partial derivatives. Therefore, an alternative but equivalent cubic b-spline transformation is given by

$$\begin{aligned} h(y) &= y + \sum_{i=-1}^{N_1} \sum_{j=-1}^{N_2} \sum_{k=-1}^{N_3} a_{i,j,k} \tilde{B}(y_1/\delta_1 - i)\tilde{B}(y_2/\delta_2 - j)\tilde{B}(y_3/\delta_3 - k) \qquad (2.14) \\ &= y + \sum_{i=-1}^{N_1} \sum_{j=-1}^{N_2} \sum_{k=-1}^{N_3} \begin{bmatrix} a_{i,j,k,1} \\ a_{i,j,k,2} \\ a_{i,j,k,3} \end{bmatrix} \tilde{B}(y_1/\delta_1 - i)\tilde{B}(y_2/\delta_2 - j)\tilde{B}(y_3/\delta_3 - k) \qquad (2.15) \end{aligned}$$

where

$$\tilde{B}(t) = \begin{cases} (3|t|^3 - 6|t|^2 + 4)/6, & |t| \leq 1 \\ (2 - |t|)^3/6, & 1 \leq |t| \leq 2 \\ 0, & \text{otherwise} \end{cases} \tag{2.16}$$

and $a_{i+l,j_m,k+n} \in \mathbb{R}^3$. The cubic b-spline basis function $\tilde{B}$ is shown in Figs. 2.2 and 2.3.

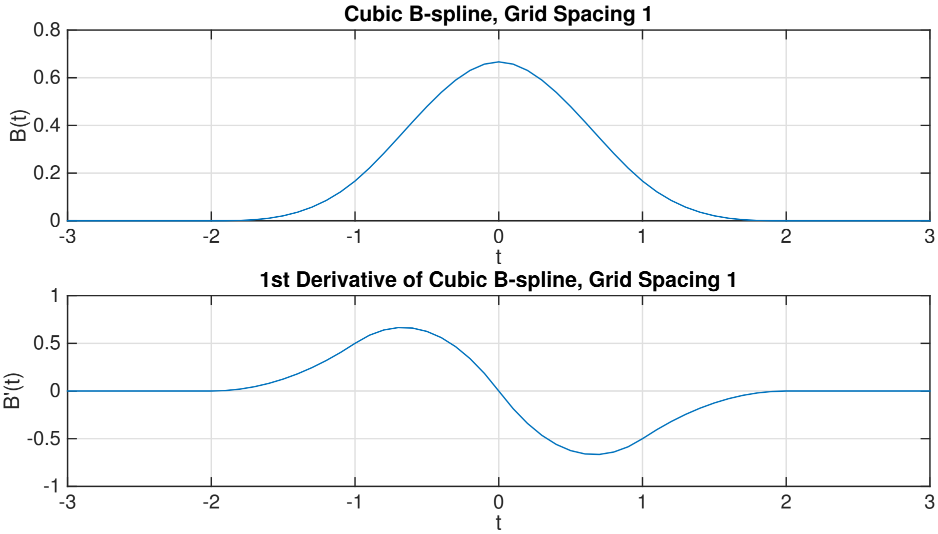


Figure 2.2. Cubic B-spline and the first derivative of a cubic B-spline with knot spacing one.

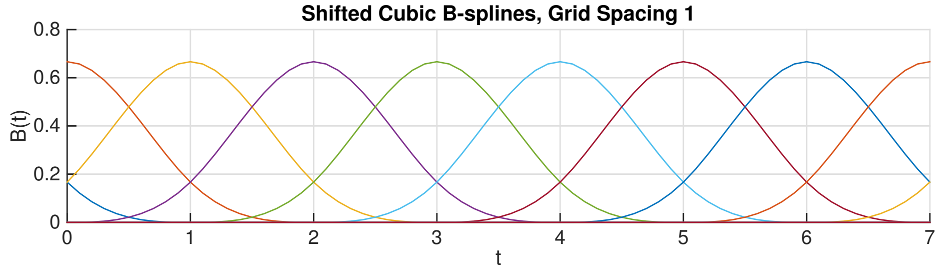


Figure 2.3. Shifted cubic B-splines with knot spacing one.

2.1.5 Fourier Series Transformation Model

A transformation can be parameterized using a 3D Fourier series [1, 14, 49, 16, 48]. In this model, each parameter is the weight of a harmonic component in a single coordinate direction.

Note that image registration algorithms are often formulated on the continuum but are discretized for implementation. Let I_1 and I_2 represent 3D image volumes of voxels dimension $N_1 \times N_2 \times N_3$. Define $\Omega_d = \{(n_1, n_2, n_3) | 0 \leq n_1 < N_1; 0 \leq n_2 < N_2; 0 \leq n_3 < N_3;$ and n_1, n_2, n_3 are integers$\}$ as the set of voxel lattice coordinates of the discrete images I_1 and I_2. Let $k = [k_1, k_2, k_3]$ and $n = [n_1, n_2, n_3]$. The displacement fields are defined to have the form

$$u(y) = \sum_{k \in \Omega_d} \mu[k] e^{j<y,N\theta[k]>} \tag{2.17}$$

for $y \in \Omega$ where $\mu[k]$ are 3×1 complex-valued vectors, $\theta[k] = [\frac{2\pi k_1}{N_1}, \frac{2\pi k_2}{N_2}, \frac{2\pi k_3}{N_3}]^T$ and $N\theta[k] = [2\pi k_1, 2\pi k_2, 2\pi k_3]^T$. The notation $< \cdot, \cdot >$ denotes the dot product of two vectors such that $< y, N\theta[k] >= 2\pi k_1 y_1 + 2\pi k_2 y_2 + 2\pi k_3 y_3$. The pullback transformation is then given by

$$h(y) = y + u(y). \tag{2.18}$$

A Fourier series transformation is periodic in x and therefore has cyclic boundary conditions for x on the boundary of Ω. The displacement fields are real because the coefficients $\mu[k]$ are constrained to have complex conjugate symmetry. Fourier series transformations are efficient to compute since they can be computed using the Fast Fourier Transform (FFT). One disadvantage of this parametrization is that each of its basis functions (i.e., sines and cosines) have infinite support. The implication of using a basis with infinite support is that that all the coefficients must be updated to accommodate a local deformation. As a result, the Fourier series coefficients are sensitive to small changes in the transformation and this causes slow convergence for registration algorithms that use this parametrization.

2.1.6 Vector Field Transformation Model

The most general form of a small deformation transformation is a vector field transformation model, i.e., a transformation that is parameterized by a displacement vector at each voxel location. Unlike the previous transformation models, adjacent values of the transformation/displacement field are not linked together by any parameters, i.e., the value of the transformation at each point in the voxel lattice Ω_d is independent of the transformation at any other point in Ω_d. It is for this reason that a vector field transformation model is referred to as a non-parameterized transformation.

A vector field transformation model has the most degrees of freedom of any transformation model. The number of degrees of freedom is equal to the number of voxels in the image times the dimension of the image being registered. For example, a 3D transformation of a 256^3 voxel image has $3 \times 256^3 \approx 5 \times 10^9$ parameters.

As a result of the large number of parameters, image registration algorithms that use vector field transformation models must use regularization (See Section 2.3.1). Regularization reduces the number of degrees of freedom and it imposes correlations between a transformation vector a point p and all the transformation vectors in a neighborhood of p.

2.1.7 Large Deformation Time-dependent Velocity Field Transformation Model

Large deformation transformation models are defined as transformations that have locally large rotation and locally large strains. In general, large deformation transformation models can accommodate registration problems that require either large or small deformations. However, large deformation models are generally more complicated than small deformation transformation models so would not necessarily be the correct choice for small deformation applications.

In this work, we will restrict our conversation to large deformation transformation models that can be modeled as a flow of diffeomorphisms. This approach was made popular through the work of Trouvé, Miller, Younes, Dupuis, Beg and Grenander [64, 22, 8] for large deformation diffeomorphic metric mapping (LDDMM). In this approach, the diffeomorphic transformation that registers image I_1 to I_2 is estimated as the end point of the flow associated to a smooth time-dependent vector field. Let $v : [0, 1] \to V$ be a time-dependent velocity vector field where V is a Hilbert space of smooth, compactly supported vector fields on Ω. The velocity vector field v defines a flow of diffeomorphisms $\phi^v : [0, 1] \to \mathcal{G}$ as the solution of the ODE

$$\frac{d}{dt}\phi^v_t(x) = v_t\left(\phi^v_t(x)\right) \tag{2.19}$$

where $\mathcal{G}$ is the set of diffeomorphisms on Ω and the superscript in ϕ^v denotes the dependence of ϕ on v. The initial condition of the flow ϕ^v at time $t = 0$ is $\phi^v_0 = Id$ where Id is the identity transformation. The end point of the flow ϕ^v at time $t = 1$ is the desired diffeomorphism $\varphi = \phi^v_1$ that registers I_1 to I_2. Thus, the goal for the large deformation tranformation model is to find a time-dependent velocity vector field v that when integrated via Eq. 2.19 generates the diffeomorphism φ that matches I_1 to I_2.

The time-dependent velocity field is often parametrized via a time varying vector field (See Section 2.1.6) [8] or by time varying momenta [50, 23, 24].

2.1.8 Large Deformation Stationary Velocity Field Transformation Model

A simple variation on the time-dependent velocity field diffeomorphic transformation model presented in the last section is to assume that the velocity vector field $v \in V$ is stationary, i.e., v is constant for time $0 \leq t \leq 1$. The large deformation stationary velocity field transformation model was made popular by the diffeomorphic demons image registration algorithm and its variants [66, 67].

For the case of stationary velocity fields, Eq. 2.19 changes to

$$\frac{d}{dt}\phi_t^v(x) = v\left(\phi_t^v(x)\right) \tag{2.20}$$

were the velocity vector field v is no longer a function of time t.

The diffeomorphic transformation produced by a stationary velocity field has fewer degrees of freedom than a transformation produced by a time-dependent velocity field. As a result, a diffeomorphism produced by a stationary velocity field is more limited than a diffeomorphism produced by a time varying velocity field. On the other hand, a stationary velocity field requires less computer storage than a time varying velocity field and there are efficient algorithms for generating the diffeomorphism from a stationary velocity field [10].

In a similar fashion to time varying velocity fields, stationary velocity fields are often parametrized using vector fields or by momenta.

Table 2.1. Summary of transformation models and their properties.

Transformation Model	**Small/Large Deformation**	**Degrees of Freedom 2D**	**Degrees of Freedom 3D**
Rigid	Small	3	6
Affine	Small	6	12
d-th order polynomial	Small	$\binom{d+2}{2}$	$\binom{d+3}{3}$
Cubic B-splines	Small	$(\lfloor \frac{N}{n} \rfloor + 3)^2$	$(\lfloor \frac{N}{n} \rfloor + 3)^3$
Fourier Series	Small	$2(2h+1)^2$	$3(2h+1)^3$
Displacement Field	Small	$2N^2$	$3N^3$
Viscous Fluid (vector field)	Large	$2N^2$	$3N^3$
Stationary velocity (momenta)	Large	varies	varies
Stationary velocity (vector field)	Large	$2N^2$	$3N^3$
Time-dependent velocity field (momenta)	Large	varies	varies
Time-dependent velocity field (vector field)	Large	$2N^2T$	$3N^3T$

2.2 Similarity Cost Functions

Similarity cost functions measure the similarity (or dissimilarity) between two images and are positive functions that measure the distance between two images or between features in the two images. The closer the cost is to zero, the closer the two images are in appearance. In most cases, image registration algorithms are formulated as a cost minimization problem of which the similarity cost is one of the costs that is minimized. The following examples illustrate how the similarity cost function is used in image registration. In the case of registering the faces of two individuals, we would expect the key features such as eyes, nose and mouth to be mapped together. A simple similarity cost function would take the center position of the eyes, nose and mouth of both faces and then compute the average distance between the corresponding center points. If the two images were identical, then the center points would align and the similarity cost would be zero. Alternatively, if one face was of a child and another of an adult, then the center lines would not align and the similarity cost would return some positive distance between the two images. In a more complex example of lung registration, we would expect features such as airways, vessels and fissures to match. In some special cases, physical markers (called fiducials) can be implanted in the subject being imaged so that they can be used as reference points for registration. Generally, however, most image registration applications lack such *a priori* information and usually resort to minimizing the overall difference of the metrics that are used for registration. In the following sections, the most commonly used similarity metrics in image registration will be outlined.

2.2.1 Point-based Similarity Cost Functions

The simplest case of measuring correspondence between two image features is the zero-dimensional case of point-based metrics.

2.2.1.1 Landmark Distance Similarity Cost Functions

Suppose there are identifiable points (such as branching points of airways and vessels) in an image, and their corresponding counterparts can also be identified in the image to be registered. This set of points is known as landmarks, and the usual implication is that there is a 1-to-1 relationship between the landmarks. Landmarks can either be a physical marker (fiducials) that is inserted into the subject before being imaged, or a virtual marker picked out manually by experts or automatically (or semi-automatically) by landmarking software. Landmark distance measures the distance (usually Euclidean) between corresponding points in the images being registered. The solution to a registration problem is obtained by minimizing the sum of distances of all the corresponding points.

2.2.1.2 Closest Point Similarity Cost Functions

In some cases, it may not be possible to identify correspondence between points in the images. Moreover, it may be that the number of identified points in one image may not equal the number of points in another image. To tackle such problems, Besl et al. [9] introduced a method that iteratively finds the closest distance between clouds of unmatched points, known as the Iterative Closest Point (ICP) method.

2.2.2 Subvolume-based Similarity Cost Function

Another feature that is used to register images are sub-volumes. Sub-volumes are more commonly referred to as regions of interest (ROIs). Matching ROIs is similar to matching landmarks, curves and surfaces. ROIs are used for matching when one wants to match sub-volumes of an image and the intensity within the sub-volume is not important. Thus, an ROI similarity cost function can be used for both mono-modality or multi-modality registration.

Assume that we want to register images I_1 and I_2 that contain M ROIs. Define the indicator image $I_{1,k}(x)$ to be one for x in the ROI of image I_1 and zero otherwise for $k = 1, \ldots, M$. Define the images $I_{2,k}$ in a similar fashion using the target image I_2. The ROI similarity cost function is then given by Eq. 2.22 where the images $I_{1,k}$ and $I_{2,k}$ are the ROI indicator images for $k = 1, \ldots, M$.

2.2.3 Voxel Intensity-based Similarity Cost Functions

2.2.3.1 Sum-of-Squared Difference (SSD) Similarity Cost Functions

The sum-of-squared difference (SSD) similarity cost functions was the first intensity based similarity cost functions used for medical image registration [7, 49]. The SSD similarity cost function assumes that corresponding points in the two images that are being registered have the same or similar intensities. The SSD similarity cost function is given by

$$SSD(I_1, I_2) = \int_{\Omega} (I_1(x) - I_2(x))^2 dx. \qquad (2.21)$$

The SSD cost function can be extended to register two image sets that have N corresponding modalities. For example, it is common to acquire multiple MRI modalities such as T1, T2 and spin density on the same subject. In this case, each modality measures a different magnetic quantity at each coordinate location. The extension of Eq. 2.21 to register two image sets having N corresponding modalities is given by

$$SSD(I_1, I_2) = \sum_{k=1}^{N} \int_{\Omega} w_k (I_{1,k}(x) - I_{2,k}(x))^2 dx \tag{2.22}$$

where it is assumed that the k-th image of I_1 is the same modality as the k-th image of I_2 and w_k are weights for each modality. The weights can be used to make one modality more important than another. If all modalities are equality important, then the weights are all set to one.

It is also possible to make the weights very spatially as shown in the following cost function

$$SSD(I_1, I_2) = \sum_{k=1}^{N} \int_{\Omega} w_k(x)(I_{1,k}(x) - I_{2,k}(x))^2 dx \tag{2.23}$$

Spatially varying weights are useful if you want to mask out regions of the image that are not important for registration. For example, it is common in brain image registration to set $w_k(x)$ to one for x locations inside the brain and set $w_k(x)$ to zero otherwise.

2.2.3.2 Normalized Cross Correlation Similarity Cost Function

The normalized cross correlation (NCC) similarity cost function [62, 4] is an intensity based similarity cost. Two images are put into register by maximizing the NCC. The NCC similarity cost only cares about correlations of intensities in local neighborhoods of the template and target images and not the actual voxel intensities.

Thus, it can be used for both mono-modality and cross-modality image registration.

The NCC cost is computed as follows. For each point x in the target image, the correlation is computed for a neighborhood patch centered at x the template image with a neighborhood patch centered at x in the target image. The correlation at each point x is then normalized. Finally, the NCC cost is then computed by integrating each of these normalized neighborhood correlations.

We now provide the numerical formulation of the NCC similarity cost function. For each $x \in \Omega$, let $N(x) \subset \Omega$ denote a neighborhood of x. Let $\mu_1(x) = \frac{\int_{N(x)} I_1(y)dy}{\int_{N(x)} dy}$ and $\mu_2(x) = \frac{\int_{N(x)} I_2(y)dy}{\int_{N(x)} dy}$ denote the average intensities of I_1 and I_2 in the neighborhood $N(x)$, respectively. Next, define $\bar{I}_1(x, y) = I_1(y) - \mu_1(x)$ and $\bar{I}_2(x, y) = I_2(y) - \mu_2(x)$ for $y \in N(x)$ to be the intensity normalized, neighborhood image patches at $x \in \Omega$ of the template and target images, respectively. The normalized correlation between the template and target images at x is then given by

$$NCC(I_1, I_2, x) = \frac{\left(\int_{N(x)} \bar{I}_1(y)\bar{I}_2(y)dy\right)^2}{\sqrt{\int_{N(x)} \bar{I}_1(y)\bar{I}_1(y)dy}\sqrt{\int_{N(x)} \bar{I}_2(y)\bar{I}_2(y)dy}} \tag{2.24}$$

and the NCC similarity cost function is then given by

$$NCC(I_1, I_2) = \int_{\Omega} NCC(I_1, I_2, x)dx. \tag{2.25}$$

Equations 2.24 and 2.25 are discretized for implementation. The neighborhood is often chosen to be a 3 × 3 × 3 or 5 × 5 × 5 voxel neighborhood.

2.2.3.3 Mutual Information Between Two Images

The mutual information (MI) between two images [69, 36, 44] measures the similarity of two images as a function of image intensity. Two images are put into register by maximizing the MI. The mutual information between two images is a function of the joint histogram between the template and target images. The MI computed from the joint histogram is based on the histogram values and not the pixel intensities. Thus, MI can be used for both mono-modality and multi-modality image registration.

The MI can be formulated mathematically as

$$MI(I_1, I_2) = H(I_1) + H(I_2) - H(I_1, I_2), \tag{2.26}$$

where $H(I_1, I_2)$ is the joint entropy of images I_1 and I_2; and $H(I_1)$ and $H(I_2)$ are the marginal entropies of $H(I_1, I_2)$. The joint entropy is calculated as

$$H(I_1, I_2) = -\sum_{m \in M} \sum_{n \in N} p(m, n) \log(p(m, n)) \tag{2.27}$$

where M is the set of gray levels in image I_1, N is the set of gray levels in image I_2, and $p(m, n)$ is the probability of intensity m in I_1 and intensity n in I_2 occurring at the same location $x \in \Omega$. The probabilities $p(m, n)$ are computed from the joint histogram of the images I_1 and I_2. Parzen-window density estimation is often used to form a better estimate of the probabilities than just using the normalized histogram values[69, 36]. The marginal entropies are given by

$$H(I_1) = -\sum_{m \in M} p(m) \log(p(m)) \tag{2.28}$$

and

$$H(I_2) = -\sum_{n \in N} p(n) \log(p(n)). \tag{2.29}$$

Using these equations, the mutual information between two images can be rewritten as

$$MI(I_1, I_2) = \sum_{m \in M} \sum_{n \in N} p(m,n) \log \left(\frac{p(m,n)}{p(m)p(n)} \right). \tag{2.30}$$

Many variations of MI have been proposed throughout the years. Common variants include those that vary the intensity bin sizes, those that use *a priori* models for estimating joint intensity probabilities, and those that use different intensity interpolation schemes.

2.2.3.4 Matching using Demons

The original demons image registration algorithm proposed by Thirion [63] was based on the optical flow equations. Since then, Pennec et al. [53, 66, 67, 68] have shown that the demons image registration approach can be reformulated as a close approximation to minimizing the SSD similarity cost function. Thus, all the modern implementations of demons use the SSD similarity cost function.

2.2.4 Attribute Matching Similarity Cost Function

The hierarchical attribute matching mechanism for elastic registration (HAMMER) [57, 58] image registration algorithm is an example of an algorithm that uses attribute matching. An attribute matching similarity cost function is any cost function that uses local intensity attributes or features to register two images. The term attribute matching similarity cost function encompasses a whole class of similarity cost functions that are all related to each other. The defining characteristic of this type of similarity cost function is that they all compute the distance between two images based on some function of the local features extracted from the images. The attribute matching similarity cost function can be used for both mono-modality and multi-modality image registration.

An example of an attribute matching similarity cost function is given by

$$AM(I_1, I_2; h) = \int_{\Omega} \omega(x) \left(\frac{\int_{N(x)} \varepsilon(z)(1 - m(a_1(h(z)), a_2(z)))dz}{\int_{N(x)} \varepsilon(z)dz} \right) dx \tag{2.31}$$

where h is the pullback transformation that maps attributes in the template coordinate system to the target coordinate system. The function $\varepsilon(x)$ assigns a spatially varying weight function that is used to give variable weights to each voxel in the target image I_2. The function $N(x)$ refers to the neighborhood of the point x in the target coordinate system. The weight function $\varepsilon(z)$ assigns larger values to the boundary voxels in the target coordinate system, since those voxels are important anatomical features and are typically easy to find in the images of sufficient contrast. The vector valued functions $a_1(z)$ and $a_2(z)$ are the attribute vectors at the location z in the template and target images, respectively. The function m measures the similarity of two attribute vectors with the range from zero to one. See [57, 58] for examples of attribute vectors and attribute similarity functions.

2.3 Registration Regularization and Constraints

Image registration algorithms often incorporate one or more types of regularization and/or constraints. Regularization is used to reduced the number of degrees of freedom of a registration algorithm while constraints are used to impose some desired property on the resulting correspondence map. Regularization and constraints are often implemented as cost functions, as Lagrange multipliers or by construction. It is often the case that a single cost function can both have a regularization effect and a constraining effect. For example, applying a low pass filter to a nonparametric transformation can have a regularization effect by inducing neighborhood correlations and a constraining effect by imposing a continuity property on the transformation. For this reason, regularization conditions are often referred to as constraints. This section will discuss the effects of regularization and constraints on image registration algorithms.

2.3.1 Regularization

Image registration algorithms are often under-determined, i.e., the transformation used in an image registration algorithm has more unknown parameters/degrees of freedom than equations needed to solve for them all. One way to reduce the number of degrees of freedom is to use regularization. Regularization reduces the number of degrees of freedom by inducing correlations between parameters. For example, image registration algorithms that use a nonparametric transformation, i.e., those that are parametrized by a vector field, always require regularization. A common regularization strategy used in this case is to correlate vector values in a local spatial neighborhood at each voxel location. The most common way this is accomplished is by using some sort of a bending energy constraint or a linear elasticity constraint.

The bending energy cost function is often written as

$$BE_1(u) = \int_\Omega ||Lu(x)||^2 dx = \int_\Omega < Lu(x), Lu(x) > dx = \int_\Omega < L^\dagger Lu(x), u(x) > dx \tag{2.32}$$

or

$$BE_2(u) = \int_\Omega < Gu(x), u(x) > dx \tag{2.33}$$

where L and $G = L^{\dagger}L$ are self adjoint, differential operators and $\dagger$ is the adjoint operator.

Examples of L are

- Membrane model: $G = \nabla^2$ where $\nabla = [\frac{\partial}{\partial x_1}, \ldots, \frac{\partial}{\partial x_n}]^T$ and $\nabla^2 = \frac{\partial^2}{\partial x_1^2} + \ldots + \frac{\partial^2}{\partial x_n^2}$ are the n-th dimensional gradient and Laplacian operator, resepectively.
- Thin-plate spline model: $L = \nabla^2$.
- Linear elasticity model: $G = \mu\nabla^2 + \eta\nabla\cdot\nabla$ where μ and η are the Lamé constants.
- Thin-plate Linear elasticity model: $L = \mu\nabla^2 + \eta\nabla \cdot \nabla$

In general, L can be any matrix of differential operators as long as it is self adjoint [31].

These same linear differential operators can be used to constrain the velocity field for diffeomorphic image registration (see Sec 2.1.7). Using these differential operators to constrain the velocity field induces a reproducing kernel Hilbert space (RKHS) structure on the set of allowable velocity fields. The kernel of the RKHS is often chosen to be a Gaussian kernel. Velocity fields with a kernel with large the standard deviation are stiff and have a small number of degrees of freedom. On the other hand, velocity fields with a kernel with small standard deviation are more flexible and have a large number of degrees of freedom. Thus, adjusting the kernel size (standard deviation) of the Gaussian kernel is one way to regularize diffeomorphic image registration algorithms.

2.3.2 Image Registration Constraints

Image registration constraints are used to induce desired properties on the transformation that defines the correspondence between two images. Image registration constraints are achieved by using cost functions or by construction.

2.3.3 Constraints on the Jacobian of the Transformation

One of the common constraints placed on a transformation is that it is a diffeomorphism. A diffeomorphism is a transformation that is a bijection (one-to-one and onto), it is differentiable and its inverse is differentiable.

A transformation is said to fold space at $x \in \Omega$ if its Jacobian determinant is negative at x. A transformation is said to be singular at $x \in \Omega$ if its Jacobian determinant is zero at x. The reason why it is important that the a transformation be a diffeomorphism is so the transformation does not fold space. A sufficient condition that a transformation does not fold space is that the Jacobian determinant of the transformation is positive for all $x \in \Omega$.

One way to enforce that a transformation is a diffeomorphism is by construction as in Section 2.1.7.

An alternative way to constrain a transformation to be a diffeomorphism is by penalizing the Jacobian determinant and the inverse Jacobian determinant. The Jacobian registration constraint is given by [14]

$$JRC(h) = \int_{\Omega} |\operatorname{det}(J(h(x)))| dx + \int_{\Omega} \left| \frac{1}{\operatorname{det}(J(h(x)))} \right| dx \tag{2.34}$$

where $J(h)$ is the Jacobian matrix of the transformation h and $\operatorname{det}(\cdot)$ is the determinant operator. Note that this cost function does not enforce that the transformation

h is a diffeomorphism. Instead, the JRC penalizes transformations that are close to being singular, i.e, minimizing the JRC function attempts to prevent singular transformations by giving these transformations large penalty costs.

2.3.4 Inverse Consistency

An image registration algorithm is said to be inverse consistent if it produces a forward transformation and a backward transformation that are inverses (or near inverses) of each other. Equivalently, an image registration algorithm is inverse consistent if it is symmetric, i.e., it does not matter which image is chosen as the template image and which image as the target image.

An image registration algorithm can impose inverse consistency or near inverse consistency on a forward transformation h and a backward transformation g using an inverse consistency cost (ICC) function. The inverse consistency cost is given by

$$ICC(h,g) = \int_\Omega ||h(x) - g^{-1}(x)||^2 dx + \int_\Omega ||g(x) - h^{-1}(x)||^2 dx. \tag{2.35}$$

Note that h and g are only inverses of each other if the inverse constency cost is zero. In practice, the ICC can be made very small so h and g can be considered nearly inverse consistent.

Another way to enforce inverse consistency is by construction. One such construction is given by the following. An unidirectional similarity cost function, such as those given in Section 2.2, can be symmetrized. For example, suppose $SSD(I_1, I_2, h) = \int_\Omega ||I_1(h(x)) - I_2(x)||^2 dx$ is the sum of square differences (SSD) cost function. The unidirectional SSD cost function can be made symmetric by adding to it the cost function $SSD(I_2, I_1, h^{-1})$ [57], i.e.,

$$\begin{aligned} SSD_{SYM} &= SSD(I_1, I_2, h) + SSD(I_2, I_1, h^{-1}) \\ &= \int_\Omega ||I_1(h(x)) - I_2(x)||^2 dx + \int_\Omega ||I_1(x) - I_2(h^{-1}(x))||^2 dx. \end{aligned}$$

Notice that for this similarity cost function, it does not matter if I_1 is the template and I_2 is the target or the roles are reversed.

Another construction to ensure inverse consistency is to deform the template and target images to a midpoint coordinate system. The difference between the template and target images are compared in the midpoint coordinate system. For example, suppose $SSD(I_1, I_2) = \int_\Omega ||I_1(x) - I_2(x)||^2 dx$ is the sum of square differences (SSD) cost function. This cost function can be converted to a midpoint coordinate system SSD cost function as follows

$$SSD_{MP} = \int_\Omega ||I_1(h(x)) - I_2(g(x))||^2 dx \tag{2.36}$$

where h is the transformation that maps image I_1 into the midpoint coordinate system and g is the transformation that maps image I_2 into the midpoint coordinate system. The transformation that deforms image I_1 into the coordinate system of I_2 is given by $h \circ g^{-1}$. Likewise, the transformation $g \circ h^{-1}$ transforms image I_2 into the coordinate system of image I_1. One can see that these two transformations are inverses of each other by composing them, i.e., $(h \circ g^{-1}) \circ (g \circ h^{-1}) = Id$ where Id is the identity map.

2.3.5 Transitivity

The transitivity property of a transformation is a property on a group of transformations that register a group of images [15]. Let $\mathcal{H} = \{h_1, \ldots, h_N\}$ be a set of diffeomorphic transformations from Ω to Ω that registers the images $\mathcal{I} = \{I_1, \ldots, I_M\}$. The set $\mathcal{H}$ is said to satisfy the transitivity property if $h \circ g \circ f = Id$ for all $h, g, f \in \mathcal{H}$ such that f is the transformation from I_i to I_j, g is the transformation from I_j to I_k and h is the transformation from I_k to I_i for $i \neq j \neq k$ and $1 \leq i, j, k \leq M$. An image registration algorithm that produces transformations that satisfy the transitivity property also satisfies the inverse consistency property. However, an image registration algorithm that produces inverse consistent transformation do not necessarily satisfy the transitivity property.

Characterizing groupwise image registration algorithms is beyond the scope of this thesis. See Christensen et al. [15] for more information about a groupwise image registration algorithm that has a transitivity constraint.

CHAPTER 3

IMAGE REGISTRATION EVALUATION METHODS

In order to evaluate how well a registration algorithm performed in mapping the coordinate systems of two images, we first need to identify what sort of common features are shared between the two images. For example, prominent features in lung CT volumes typically consist of lung boundaries, fissures, airways, and vessels. Naturally, we expect each of these features in one image to map to its corresponding feature in a successfully registered image. However, due to reasons such as poor image quality or illness, extraction of certain features from images may not be reliable, or worse yet, not possible at all. Furthermore, in situations such as multi-modality registration, features that are obvious in one modality may not be as obvious in another modality. It is also important to emphasize that just because certain features lined up perfectly does not imply perfect registration (e.g., the aperture problem is a classic example of bad registration despite perfect boundary correspondence.) The following sections describe various different types of measures that can be used to evaluate image registration performance.

3.1 Feature and Image Similarity Evaluation

The most natural approach to evaluating image registration performance is to measure how well the algorithm performed in terms of similarity metrics. Oftentimes, the similarity metrics used to evaluate registration performance are not the same as the ones used for registration. For example, a registration algorithm might only use voxel intensity information to drive the registration, while landmarks are used for evaluation. Public registration competitions such as the EMPIRE10 challenge did just that - only the lung images and masks were made available to the participants, but landmarks and segmentations of boundaries and fissures were used to evaluate registration performance. In the following sections, common similarity metric-based evaluation methods and their pros and cons will be outlined.

3.1.1 Landmark Error

Let Ω be a subset of $\mathbb{R}^3$. Let $I_m : \Omega \to \mathbb{R}$ and $I_f : \Omega \to \mathbb{R}$ be template and target images to be registered, respectively. Let a pair of points $p \in \Omega$ and $q \in \Omega$ identify a landmark in images I_m and I_f, respectively. Then, for N landmark locations, we obtain point sets $P = \{p_1, p_2, ..., p_N\}$ and $Q = \{q_1, q_2, ..., q_N\}$. Before registration, the misalignment of the landmarks in image pair I_m and I_f is measured with Preregistration Mean Landmark Error ($MLE_{preregistration}$), which is defined as

$$MLE_{preregistration} = \sum_{i=1}^{N} \|p_i - q_i\|,$$

where $\|\cdot\|$ is the standard Euclidean norm.

For a pair of registered images where $\phi : \Omega \to \Omega$ is the push-forward (i.e., Lagrangian) transformation that maps coordinate system $x \in \Omega$ of I_m to coordinate system $y \in \Omega$ of I_f. Then the MLE for the registered landmark set is defined as

$$MLE = \sum_{i=1}^{N} \|\phi(p_i) - q_i\|, \tag{3.1}$$

evaluating the location of each deformed landmark point in the template image coordinate space with respect to their corresponding location in the target image coordinate space. However, because of issues arising from discretization of images, registration algorithms generally represent transformations in the pullback (i.e., Eulerian) sense, mapping coordinate system y to coordinate system x. This means that for pullback transformation $h : \Omega \rightarrow \Omega$, the MLE can be written as

$$MLE = \sum_{i=1}^{N} \left\| p_i - h(q_i) \right\|, \qquad (3.2)$$

evaluating the location of the deformed landmark points with respect to the template image landmark points.

Typically, most landmark error evaluations are carried out using Eq. 3.1, because it is straight-forward and does not require any more than the original transformation generated by the registration algorithm. This does not pose any problem or inconvenience if comparing landmark error values before and after registration is the desired mode of evaluation. However, it is often desirable that landmark points in the template image coordinate space be projected into the target image coordinate space for evaluation, especially because the landmark points can be juxtaposed on top of the deformed image for visual inspection. If a pullback transformation has a closed-form inverse, the inverse is the push-forward transformation. However in general, most inverses of transformations must be obtained through an iterative estimation process, and transformations of some algorithms are not invertible at all (i.e., creates singularities). Certain groups of registration algorithms, such as [14], [33], [8], [29], [2], [66], [3], [67] and [4] ensure that the transformations are invertible by enforcing diffeomorphism or symmetric constraints. Given that the registration algorithm provides either a push-forward transformation out of the box, or an invertible transformation, it is more useful to perform landmark evaluations on the target image coordinate system so that landmark alignment performance can be evaluated against other evaluation metrics on the same coordinate system.

The advantage of landmark-based evaluation of registration performance is that it provides a direct measure of how well known correspondences line up after registration. However, landmark error only ensures registration correspondence accuracy at landmark locations and does not provide any correspondence information at non-landmark locations. This means that in order to get better estimation of registration accuracy using landmark error, a dense population of landmark points is desired.

Furthermore, as mentioned in [28], there are errors involved with localization of landmarks. Fiducial Localization Error (FLE) is a measure of of erroneous displacement of a landmark point from its true location. FLE can occur from several sources: discretization error, noise and algorithm error. Typically, features that are used to localize landmark points are several voxels in size, and quite likely irregular in shape. This means that locating the centroid of the landmark feature is challenging, and even if a centroid was successfully located at a voxel level, it does not guarantee this location is the true centroid of the feature in physical space. Furthermore, if noise is present in the image, this may affect the accuracy of localizing the centroid of the feature being landmarked. Finally, error may be introduced by how accurately the localization algorithm (manual, semi-automatic or automatic methods) locates the landmark point. FLE is a type of error that cannot be observed directly, but its impact can be observed indirectly from the registration error caused by it. Since FLE is present in both images being registered, there is a level of uncertainty involved with landmark error measures - in other words, perfect landmark error does not equal perfect alignment of landmarks in physical space. One of the ways to alleviate FLE is to have multiple observers localize the landmarks, and averaging the location.

3.1.2 ROI Overlap Evaluation

The alignment of objects, structures, organs, regions of interest (ROIs), etc., are a good indicator of how well two images are registered. These subvolumes are defined by partitioning or segmenting an image into objects or ROIs. The relative overlap of segmentations is a measure of how well two corresponding segmented regions agree with each other.

Assume S_i and T_i are defined as the i^{th} segmented region in the deformed source and target volumes. Then the three different kinds of volume overlap measures are defined as follows:

- Target overlap

$$TO_i(S_i, T_i) = \frac{|S_i \cap T_i|}{|T_i|}$$

- Mean overlap (also known as DICE coefficient)

$$MO_i(S_i, T_i) = 2\frac{|S_i \cap T_i|}{|S_i| + |T_i|}$$

- Union overlap (also known as relative overlap)

$$UO_i(S_i, T_i) = \frac{|S_i \cap T_i|}{|S_i \cup T_i|}$$

Apart from determining the minimum and maximum overlaps, each overlap measures can be totaled or averaged over the entire volume.

Target overlap is a simple way of measuring region overlap which is defined by the volume the deformed region in the source overlaps with the corresponding region in the target. This way of defining overlap works fine as long as the two related regions have the same volume. However, if the two regions being compared have different volumes, the percentage overlap values may not accurately report the same concept. 50% target overlap does not imply half of one region overlaps half of another. In fact, 100% target overlap of S2 and T2 can be achieved by half of region S2 overlapping entire region T2. Therefore, it can be concluded that target overlap is generally a measure that is susceptible to volume discrepancies.

Mean overlap, otherwise known as the DICE coefficient, is an improvement over target overlap which takes the ratio of the overlapping volume over the mean volume of both regions. With this modification, mean overlap of S1 and T2 is 50%, and mean overlap of S2 and T2 is 33.3%. This is certainly a major improvement over target overlap.

Union overlap, which is also called relative overlap, is very similar to mean overlap and also tackles the deficiency of target overlap. The difference of union overlap over mean overlap is that instead of the denominator being the mean volume of both regions, it is now the union volume of both regions. With this definition, union overlap of S1 and S2 is 33.3%, and 20% for S2 and T2.

An intrinsic problem of any volume overlap measure is the fact that the overlap ratio does not automatically account for the volume biases. In other words, 90% relative overlap of two regions of volume 1000 voxel^3 would yield an overlapping region of volume 900 voxel^3, while 100% relative overlap of volume 100 voxel^3 would yield an overlapping region of volume 90 voxel^3. Clearly, while 100% relative overlap map be an impressive number, it may not necessarily imply better match if the actual volumes of regions in comparison are considered. Therefore, it is recommended to account for volumes biases when computing any overlap measure, such as normalizing by volume.

Another fundamental problem with volume overlap measure is that definitions of regions may not be clear-cut and be susceptible to variance. This variance may be large in applications where there is a lack of contrast between regions. Some such example would be the segmentation of prostate and bladder from CT volumes, or defining a functional atlas of brains using fMRI. Even having the same expert perform the segmentations does not guarantee repeatability over time and equipment used.

While volume overlap may provide an indication of macroscopic global match performance, it fails to provide any information about the match performance within the regions. For example, even if the deformed source region and the target region shows good overlap measures, it does not guarantee that the mapping of each microscopic local regions or points are equally as good. Therefore, volume overlap may be considered a better global performance measure than local performance measure.

3.1.3 Average Volume Difference

Defining volume as the sum of voxel intensities, similarity between two registered images can be measured by comparing their volumes. One method to measure registration performance using volume similarity is to compute average region-wise volume difference between a population of images registered with a target image. Thus, Average Volume Difference (AVD) across region R of M images registered with target image T_j is defined as

$$AVD_{j,R} = \frac{1}{M}\sum_{i=1}^{M}\left(\frac{1}{|R|}\sum_{x\in R} T_i(h_{ij}(x)) - \frac{1}{|R|}\sum_{x\in R} T_j(x)\right)^2 \tag{3.3}$$

where h_{ij} is the Eulerian transformation that maps a point in image T_i to point x in T_j, $R = \{x_1, x_2, \cdots, x_N\}$ is a subregion of N voxels in T_j, and $|\cdot|$ is the cardinality operator denoting the number of voxels in a region. The differences of the mean volumes were used instead of direct sums because a pair of registered regions may not have the same number of voxels. For an ideal set of registrations, AVD will equal zero.

3.1.4 Average Sum of Squared Differences

Assuming that a pair of registered images differ only by Gaussian noise, registration accuracy can be evaluated by measuring voxel intensity difference of the registered image pair. A common way to measure voxel intensity difference is sum of squared differences, which is sensitive to small number of voxels with very large intensity differences [56, 20]. To minimize the effects of noise, population-based SSD will be employed in this work. For a population of M images registered with target image T_j, Average Sum of Squared Differences (ASSD) across region R is defined as

$$ASSD_{j,R} = \frac{1}{M}\sum_{i=1}^{M}\sum_{x\in R}(T_i(h_{ij}(x)) - T_j(x))^2 \tag{3.4}$$

where h_{ij} is the Eulerian transformation that maps a point in image T_i to point x in T_j, and $R = \{x_1, x_2, \cdots, x_N\}$ is a subregion of N voxels in T_j. For an ideal set of registrations, ASSD will equal zero.

3.1.5 Intensity Variance

Besides measuring sum of squared differences of registered image intensities, registration performance can be evaluated by registering a population of images with a target image and averaging the intensities of the registered images. The idea behind this is that the better the registration algorithm is, the closer each registered image looks to the target image and the sharper the intensity average image [59, 70]. One way to measure the sharpness of the intensity average image is to compute the intensity variance of the registered images. Intensity Variance (IV) image of a population of M images registered to image T_j is computed as

$$IV_j(x) = \frac{1}{M-1}\sum_{i=1}^{M}(T_i(h_{ij}(x)) - ave_j(x))^2 \qquad \text{where} \qquad ave_j(x) = \frac{1}{M}\sum_{i=1}^{M}T_i(h_{ij}(x)), \tag{3.5}$$

T_i is the i^{th} image of the population and $h_{ij}(x)$ is the transformation from image T_i to T_j with respect to a Eulerian coordinate system. In an ideal case where all images are perfectly registered with the target image, IV will equal zero.

3.1.6 Average Correlation Coefficient

Correlation coefficient measures the linear dependence between the intensities of two registered images based on the assumption that registered images have linear intensity relationship [56, 20]. Average Correlation Coefficient (ACC) of a population of M images registered to image T_j over region R is defined as

$$ACC_{j,R} = \frac{1}{M}\sum_{i=1}^{M} \frac{\sum\limits_{x\in R}(T_i(h_{ij}(x)) - \bar{T}_i)(T_j(x)-\bar{T}_j)}{\sqrt{\sum\limits_{x\in R}(T_i(h_{ij}(x))-\bar{T}_i)^2 \cdot \sum\limits_{x\in R}(T_j(x)-\bar{T}_j)^2}}, \tag{3.6}$$

where h_{ij} is the Eulerian transformation mapping image T_i to T_j, and $R = \{x_1, x_2, \cdots, x_N\}$ is a subregion of N voxels in T_j. For mono-modal image registration applications, a set of perfectly registered images will result in an ACC of one.

3.1.7 Average Mutual Information

Mutual information measures the statistical dependence between the intensities of corresponding voxels in a pair of images, which is assumed to be maximal if the images are registered [19, 43, 61, 44, 45, 54]. Average Mutual Information (AMI) of a population of M images registered to image T_j over region R is defined as

$$AMI_{j,R} = \frac{1}{M}\sum_{i=1}^{M}\sum_{x\in R} p_{ij}(T_i(h_{ij}(x)), T_j(x)) log_2 \frac{p_{ij}(T_i(h_{ij}(x)), T_j(x))}{p_i(T_i(h_{ij}(x))) \cdot p_j(T_j(x))} \tag{3.7}$$

where h_{ij} is the Eulerian transformation mapping image T_i to T_j, $R = \{x_1, x_2, \cdots, x_N\}$ is a subregion of N voxels in T_j, p_{ij}, p_i and p_j are the joint and marginal distributions of the pair $(T_i(h_{ij}(x)), T_j(x))$ and of $T_i(h_{ij}(x))$ and $T_j(x)$, respectively.

3.1.8 Midpoint Registration Error

The traditional approach to measure registration performance is to evaluate correspondence either in the coordinate system of the moving image or the fixed image. However, with certain group of registration algorithms such as [8], [29], [2], [66], [3], [67], [35], [4] and [6] that parametrize transformation as a time-integral of time-varying or stationary velocity fields, registration evaluation can be performed in the transient space. In this group of algorithms, the push-forward transformation $\varphi : \Omega \to \Omega$ from the moving image coordinate system, $\mathbf{x} \in \Omega$, to the fixed image coordinate system, $\mathbf{y} \in \Omega$, is defined as

$$\mathbf{y} = \varphi(\mathbf{x}) = \phi_0(\mathbf{x}) + \int_0^1 v_t(\phi_t(\mathbf{x}))dt \tag{3.8}$$

where $v_t : \Omega \to \mathbb{R}^n,\ t \in [0,1]$ is the time-varying velocity vector field specified by the ODE $\dot{\phi_t} = v_t(\phi_t)$ and $\phi_t : \Omega \to \Omega,\ t \in [0,1]$ is the transformation that maps coordinate system $\mathbf{x}$ at time $t = 0$ to its new location at time t (i.e., $\phi_0 = Id$ and $\varphi = \phi_1$). In this setting, the optimal transformation that solves the image registration problem in space V (a Hilbert space of smooth, compactly supported vector fields on Ω) of $v_t \in V$ can be obtained by solving

$$\hat{v} = \underset{v:\dot{\phi_t}=v_t(\phi_t)}{\operatorname{argmin}} \left(\int_0^1 \|v_t\|_V^2 \, dt + \gamma \int_\Omega Sim\left(I_m \circ \phi_1^{-1}, I_f\right) d\mathbf{x} \right) \tag{3.9}$$

where $Sim(\cdot,\cdot)$ is an image similarity function such as squared-difference, cross-correlation, etc., and I_m and I_f are moving and fixed images, respectively. It can be ensured that the solution to Eq. 3.9 is in the space of diffeomorphisms if norm is defined on V through a differential operator of the type $L = (-\alpha\Delta + \gamma)^\beta I_{n\times n}$ where $\beta > 1.5$ in 3-dimensional space such that $\|f\|_V = \|Lf\|_{L^2}$.

The symmetric diffeomorphic registration formulation is illustrated in Fig. 3.1 and described as follows:

For a pair of images $I_1(\mathbf{x})$ and $I_2(\mathbf{y})$ with coordinate systems $\mathbf{x}, \mathbf{y} \in \Omega$, let the time-variant diffeomorphic forward and reverse push-forward maps be $\phi_t : \Omega \to \Omega$ and $\psi_t : \Omega \to \Omega$ respectively, where $t \in [0, 1]$. At zero optimization time $t = 0$, we have the identity mappings $\mathbf{x} = \phi_0(\mathbf{x})$ and $\mathbf{y} = \psi_0(\mathbf{y})$; and at full optimization time $t = 1$, we have the full forward and reverse mappings $\mathbf{y} = \phi_1(\mathbf{x})$, and $\mathbf{x} = \psi_1(\mathbf{y})$. Then, by setting similarity term such as the squared difference term $\left\| I_1(\phi_{0.5}^{-1}(\mathbf{z})) - I_2(\psi_{0.5}^{-1}(\mathbf{z})) \right\|^2$, the forward and reverse mappings are solved until time $t = 0.5$, which is referred here as the "midpoint." Finally, to get the mapping from $\mathbf{x}$ to $\mathbf{y}$, and vice versa, the half-time forward and reverse maps are composed as $\mathbf{y} = \psi_{0.5}^{-1}(\phi_{0.5}(\mathbf{x}))$ and $\mathbf{x} = \phi_{0.5}^{-1}(\psi_{0.5}(\mathbf{y}))$, respectively.

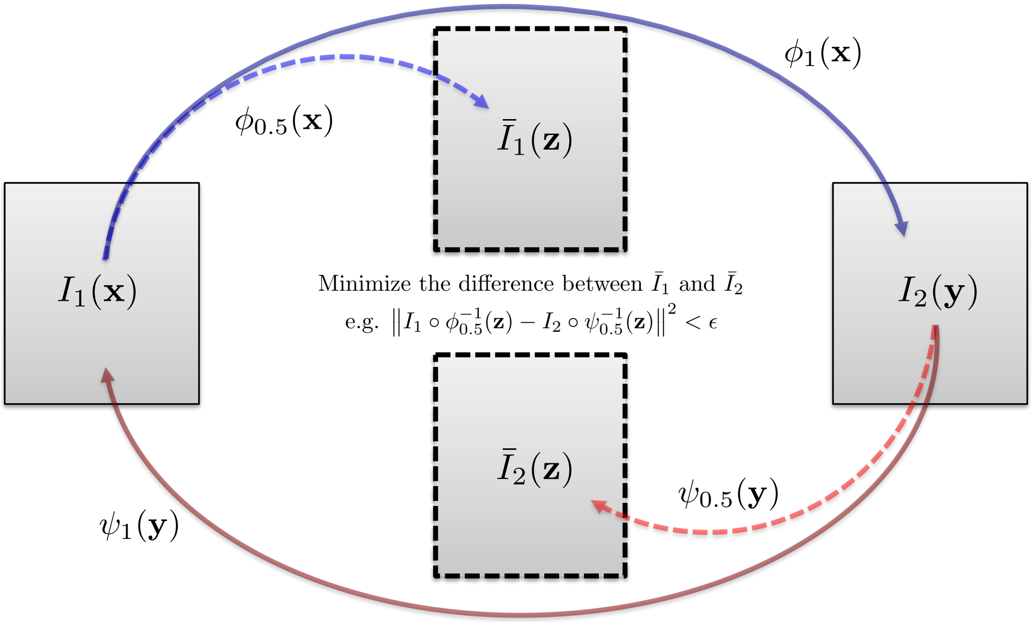

$\phi_1(\mathbf{x})$
$\phi_{0.5}(\mathbf{x})$
$\bar{I}_1(\mathbf{z})$
$I_1(\mathbf{x})$
Minimize the difference between $\bar{I}_1$ and $\bar{I}_2$
e.g. $\left\| I_1 \circ \phi_{0.5}^{-1}(\mathbf{z}) - I_2 \circ \psi_{0.5}^{-1}(\mathbf{z}) \right\|^2 < \epsilon$
$I_2(\mathbf{y})$
$\bar{I}_2(\mathbf{z})$
$\psi_{0.5}(\mathbf{y})$
$\psi_1(\mathbf{y})$

Figure 3.1. Illustration of the SyN registration method.

One claim of advantage of these symmetric registration methods is that because of the diffeomorphic nature of the maps, the composition maps $\psi_{0.5}^{-1}(\phi_{0.5}(\mathbf{x}))$ and $\phi_{0.5}^{-1}(\psi_{0.5}(\mathbf{y}))$ are exact inverses of each other, resulting in perfect inverse consistency in both directions. However, the symmetric property of transformations does not guarantee proper registration. Fig. 3.2) illustrates how point $\mathbf{p}$ in image I_1 may be mapped to point $\hat{\mathbf{q}}$ instead of its true correspondence $\mathbf{q}$, even when the forward and reverse maps between the two images are perfectly inverse consistent.

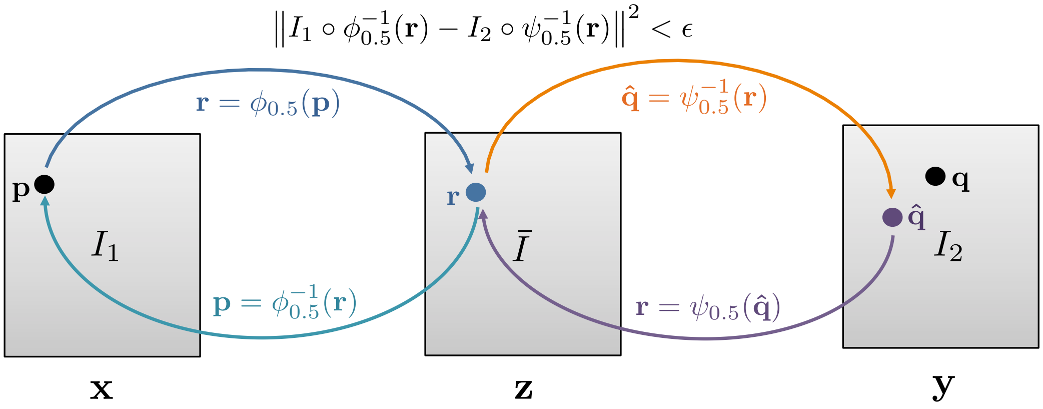

$\left\|I_1 \circ \phi_{0.5}^{-1}(\mathbf{r}) - I_2 \circ \psi_{0.5}^{-1}(\mathbf{r})\right\|^2 < \epsilon$
$\mathbf{r} = \phi_{0.5}(\mathbf{p})$
$\hat{\mathbf{q}} = \psi_{0.5}^{-1}(\mathbf{r})$
$\mathbf{p}$
I_1
$\mathbf{r}$
$\bar{I}$
$\mathbf{q}$
$\hat{\mathbf{q}}$
I_2
$\mathbf{p} = \phi_{0.5}^{-1}(\mathbf{r})$
$\mathbf{r} = \psi_{0.5}(\hat{\mathbf{q}})$
$\mathbf{x}$
$\mathbf{z}$
$\mathbf{y}$

Figure 3.2. Illustration of midpoint error.

The main problem with such registration method is that there is no guarantee that the midpoint image actually is the "mean shape" between I_1 and I_2. For example, suppose we are trying to register a large circle to a small circle as shown in Fig. 3.3. The large circle and the small circle may both map to an oval, a rectangle, a point, or any other arbitrary midpoint shape. Since both ϕ and ψ map to the same midpoint shape, we have "perfect" registration according to the symmetric registration algorithm in discussion.

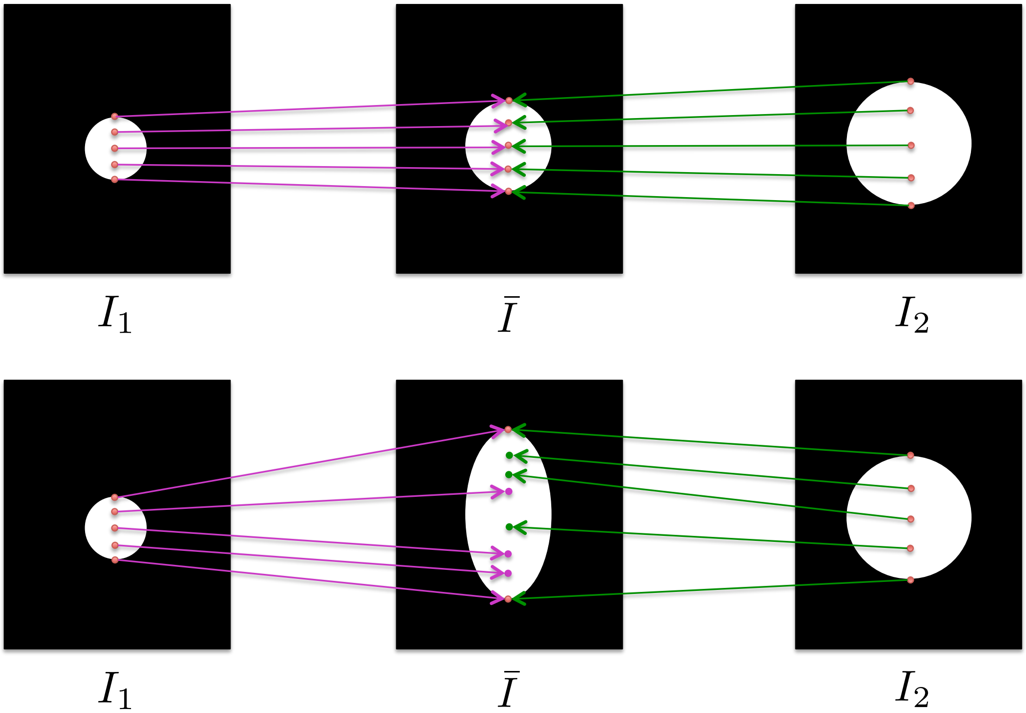
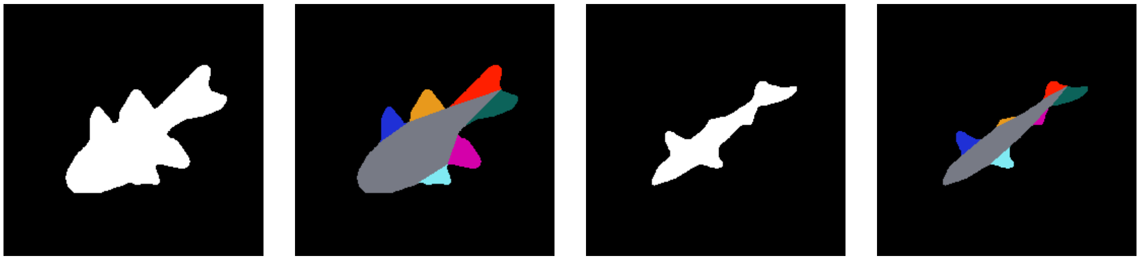


Figure 3.3. Illustration of two different midpoint images that satisfy the registration problem.

Furthermore, there is no guarantee that point $\mathbf{r}_1$, which is a mapping of point $\mathbf{p}$ in I_1 to the midpoint image coordinates by $\phi(\mathbf{p}, 0.5)$, corresponds to point $\mathbf{r}_2$, which is a mapping of point $\mathbf{q}$ in I_2 to the midpoint image coordinates by $\psi_{0.5}(\mathbf{q})$. In other words, although the midpoint images M_1 and M_2 may look exactly the same in terms of voxel intensity, it does not guarantee that the coordinate correspondences actually match.

Real examples of midpoint registration error is demonstrated here with simple

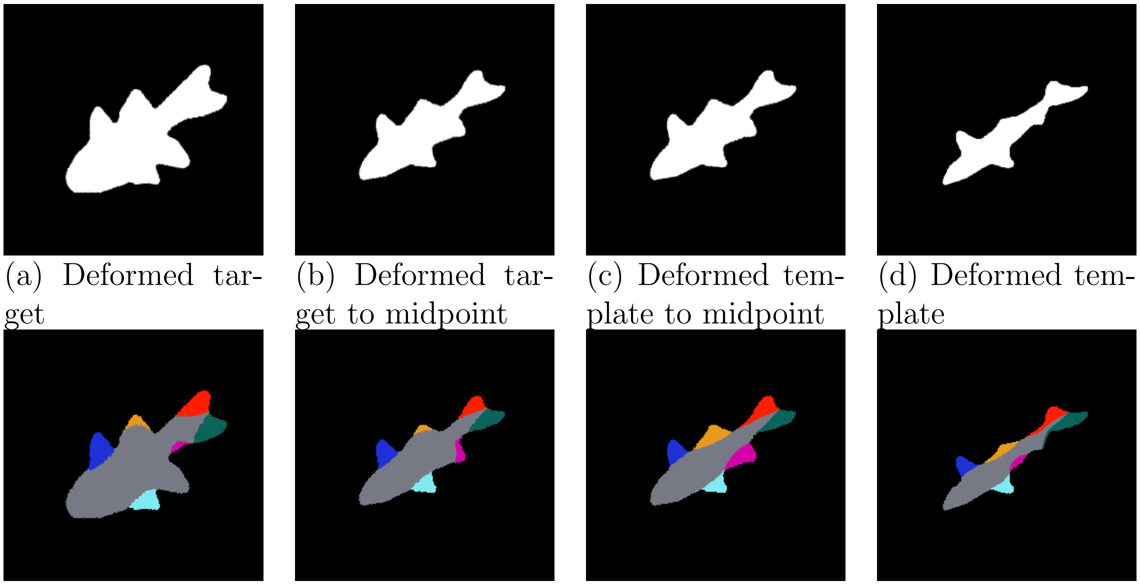
2D figures as shown in Fig. 3.4. The big and small fish images in (a) and (c) were registered using SyN registration, and the fin segmentations in (b) and (d) were used to evaluate the pointwise correspondence within the registered fish.



(a) Template (b) Target (c) Template Mask (d) Target Mask

Figure 3.4. Big and small fish images used to demonstrate the midpoint registration error. The fins of the fish were segmented to demonstrate where they get mapped to after registration.

Figure 3.5 shows the registration result of the big and small fish images in both directions, as SyN transformations are diffeomorphic. Comparing deformed target (a) with the template, and deformed template (d) with the target, it appears superficially that the registration was successful. Furthermore, the midpoint images (b) and (c), which are generated by deforming the template and the target using “half transformations” generated by the SyN algorithm as illustrated in Fig. 3.1, also show that the forward and reverse midpoint images match. However, applying the generated transformations to the mask images reveal that the correspondences of the fins do not match. The deformed masks (f) and (g) at the midpoint show that although the sum of squared difference error of (b) and (c) is small, the correspondence within the image is off.



(a) Deformed target

(b) Deformed target to midpoint

(c) Deformed template to midpoint

(d) Deformed template

(e) Deformed target mask

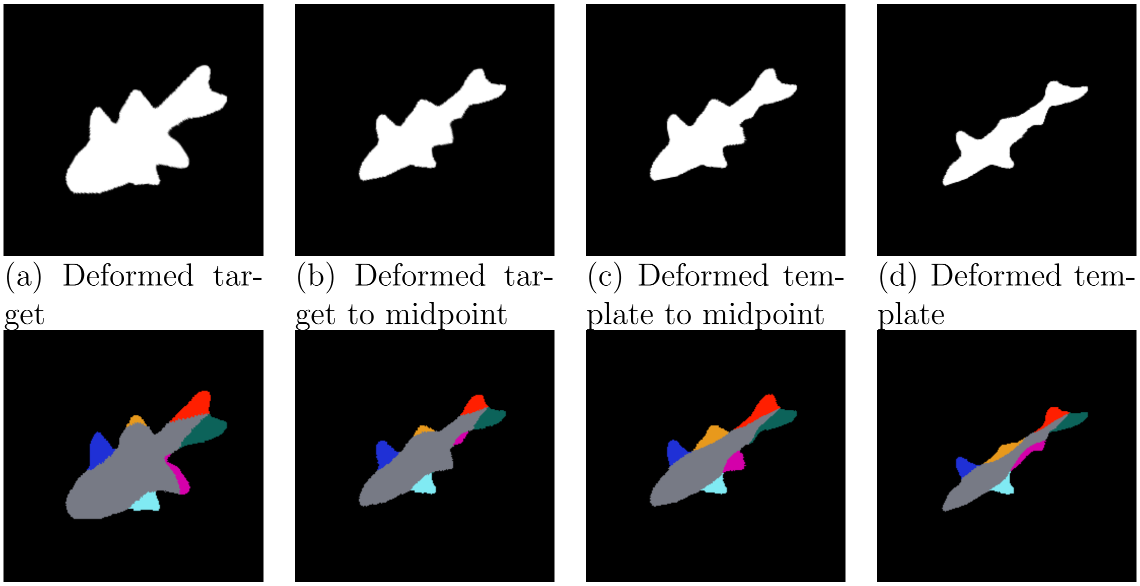
(f) Deformed target mask to midpoint

(g) Deformed template mask to midpoint

(h) Deformed template mask

Figure 3.5. (a) & (d) Deformed target and template towards reverse and forward registration directions. (b) & (d) Deformed target and template towards reverse midpoint and forward midpoint. Notice that the deformed target matches the template shape, and vice versa. The deformed midpoint template and target also match each other well. However, the deformed masks shown in (e) - (f) reveal that the fins are misregistered.

Figure 3.6 shows the result of yet another SyN registration of the same fish images. Again, the deformed template and the target appear to be registered well. But notice that the midpoint images (b) and (c) look different from the previous registration attempt. Although the end registration result appears identical to the previous registration, the midpoint images tell a different story. The deformed masks in (e) - (h) further confirms that the registration is indeed different. Particularly, deformed masks (f) and (g) at the midpoint shows that there are essentially infinite number of midpoint images that can result with the same deformed template and target images.



(a) Deformed target

(b) Deformed target to midpoint

(c) Deformed template to midpoint

(d) Deformed template

(e) Deformed target mask

(f) Deformed target mask to midpoint

(g) Deformed template mask to midpoint

(h) Deformed template mask

Figure 3.6. Another registration result similar to Fig. 3.5 with different registration parameters. Notice that the midpoint images look different to the previous registration result. This shows that there are infinitely many midpoint mappings that can produce the same deformations of the template and target.

Having shown that a registration algorithm can generate infinitely many different mappings to the midpoint image to produce the same end registration result, a method for evaluating midpoint registration error is defined below.

Let $I_1(\mathbf{x})$ and $I_2(\mathbf{y})$ be a pair of images with coordinate systems $\mathbf{x}$ and $\mathbf{y}$, respectively. We then define $\phi(\mathbf{x}, t)$ as the correspondence map from image I_1 to I_2 with image registration optimization time, $t \in [0, 1]$, such that $\mathbf{x} = \phi(\mathbf{x}, t = 0)$ and $\mathbf{y} = \phi(\mathbf{x}, t = 1)$. Similarly, we define $\psi(\mathbf{y}, t)$ as the reverse mapping from image I_2 to I_1, such that $\mathbf{y} = \psi(\mathbf{y}, 0)$ and $\mathbf{x} = \psi(\mathbf{y}, 1)$. Then let $M(\mathbf{z})$ be the midpoint image between I_1 and I_2, mapped by $\mathbf{z} = \phi(\mathbf{x}, 0.5)$ and $\mathbf{z} = \psi(\mathbf{y}, 0.5)$, simultaneously.

Let point $\mathbf{p}_i$ be the i^{th} landmark in image I_1, and $\mathbf{q}_i$ the i^{th} landmark in image I_2 that has correspondence mapping with $\mathbf{p}_i$. Then, the voxel-wise Midpoint Landmark Error (MLE) is defined as

$$MLE_i = \left\|\phi(\mathbf{p}_i, 0.5) - \psi(\mathbf{q}_i, 0.5)\right\|^2 . \tag{3.10}$$

It then follows that for N landmark points, the Average Midpoint Landmark Error (AMLE) is defined as

$$AMLE = \frac{1}{N}\sum_{i=1}^{N} \left\|\phi(\mathbf{p}_i, 0.5) - \psi(\mathbf{q}_i, 0.5)\right\|^2 , \tag{3.11}$$

and Maximum Midpoint Landmark Error (MMLE) as

$$MMLE = \max_{1 \le i \le N} \left\|\phi(\mathbf{p}_i, 0.5) - \psi(\mathbf{q}_i, 0.5)\right\|^2 , \tag{3.12}$$

We can observe that the Average Landmark Error (ALE) can be written as

$$ALE = \frac{1}{N}\sum_{i=1}^{N} \left\|\psi^{-1}(\phi(\mathbf{p}_i, 0.5), 0.5) - \mathbf{q}_i\right\|^2 .$$

Similarly, image intensities at the midpoint can be evaluated. For image domain Ω of image $\bar{I}$, the Average Midpoint Intensity Error (AMIE) can be written as

$$AMIE = \frac{1}{|\Omega|}\int_{\Omega}(I_1(\phi^{-1}(\mathbf{z},0.5)) - I_2(\psi^{-1}(\mathbf{z},0.5))^2 dz, \tag{3.13}$$

and Maximum Midpoint Intensity Error (MMIE) as

$$MMIE = \max_{\mathbf{z}\in\Omega}(I_1(\phi^{-1}(\mathbf{z},0.5)) - I_2(\psi^{-1}(\mathbf{z},0.5))^2 \tag{3.14}$$

Note that the inverses of the transformations ϕ and ψ are used (i.e., Eulerian transformation), because the coordinate systems of the transformations have to match.

The sum of the tangent vectors of the deformations ϕ and ψ with respect to time should cancel each other at the midpoint. The analogy to this is building the transcontinental railway from both ends of the coasts in towards the middle - if the tangent vectors of the two railroads do not match when they meet, that will result in an unintended sharp bend in the middle. Likewise, a geodesic registration path should join smoothly in the middle with matched tangent vectors. Following this intuition, the voxel-wise Midpoint Tangent Error (MTE) is written as

$$MTE(\mathbf{z}) = \left\| \left.\frac{\partial\phi^{-1}(\mathbf{z},t)}{\partial t}\right|_{t=0.5} + \left.\frac{\partial\psi^{-1}(\mathbf{z},t)}{\partial t}\right|_{t=0.5} \right\|^2 . \tag{3.15}$$

Once again, the inverses of the transformations ϕ and ψ are used to match the coordinate systems. It is necessary for the transformation to be differentiable with respect to optimization time in order for to quantify this error measure. In discrete time, this implies that access to intermediate transformations along optimization time, or geodesic path, must be available.

3.2 Transformation Evaluation

3.2.1 Jacobian

The Jacobian [11, 17, 14] measures the pointwise expansion and contraction at each point of the image. The Jacobian of the transformation at point x is defined as

$$J(h(x)) = \begin{vmatrix} \frac{\partial h_1(x)}{\partial x_1} & \frac{\partial h_1(x)}{\partial x_2} & \frac{\partial h_1(x)}{\partial x_3} \\ \frac{\partial h_2(x)}{\partial x_1} & \frac{\partial h_2(x)}{\partial x_2} & \frac{\partial h_2(x)}{\partial x_3} \\ \frac{\partial h_3(x)}{\partial x_1} & \frac{\partial h_3(x)}{\partial x_2} & \frac{\partial h_3(x)}{\partial x_3} \end{vmatrix}. \tag{3.16}$$

In the Eularian frame of reference, a Jacobian value of one corresponds to zero expansion or contraction, a value greater than one corresponds to contraction and a value less than one corresponds to expansion. A negative Jacobian indicates singularity, which produces a 1-to-many mapping and as a result folds the domain inside out [17, 13]. The Jacobian evaluates the quality of the transformation rather than the quality of registration, measuring how well the transformation preserves topology.

A transformation has to be continuous, piecewise differentiable and has a positive Jacobian in the continuous domain. However, the positivity of the Jacobian at discretization nodes does not necessarily ensure that this property is true in the continuous domain.

3.2.2 Inverse-Consistency

The inverse consistency and transitivity metrics evaluate registration performance based on desired transformation properties [14, 15, 13]. The inverse consistency metric measures the inverse consistency error between a forward and reverse transformation between two images. Ideally the forward transformation equals the inverse of the reverse transformation implying a consistent definition of correspondence between two images, i.e., correspondence defined by the forward transformation should be the same as that defined by the reverse transformations. Thus, composing the forward and reverse transformations together produces the identity map when there is no inverse consistency error. The inverse consistency error is defined as the squared difference between the composition of the forward and reverse transformations and the identity mapping.

The voxel-wise cumulative inverse consistency error (CICE) with respect to template image j is computed as

$$CICE_j(x) = \frac{1}{M}\sum_{i=1}^{M} \|h_{ji}(h_{ij}(x)) - x\|^2$$

where h_{ij} is the transformation from image i to j, M is the number of images in the evaluation population and $\|\cdot\|$ is the standard Euclidean norm. Inverse consistency error is a measure that measures how similar the forward transformation estimated from source to target volume is to the reverse transformation estimated from target to source. While inverse consistency does not measure the accuracy of the transformation, it measures the consistency of the correspondence defined by forward and reverse transformations between two coordinate systems [14]. It is important to note that zero inverse consistency does not imply accuracy of the correspondence. For example, an identity mapping has perfect inverse consistency, but the registration is inaccurate for non-identical image pairs.

3.2.3 Transitivity

The transitivity metric [15, 39, 30] evaluates how well all the pairwise registrations of the image population satisfy the transitivity property. The transitivity property is important to minimize correspondence errors when two transformations are composed together. Ideally, transformations that define correspondence between three images should project a point from image A to B to C to A back to the original position. The transitivity error for a set of transformations is defined as the squared error difference between the composition of the transformations between three images and the identity map.

The voxel-wise cumulative transitivity error (CTE) with respect to template image j is computed as

$$CTE_k(x) = \frac{1}{(M-1)(M-2)} \sum_{\substack{i=1 \\ i \neq k}}^{M} \sum_{\substack{j=1 \\ j \neq i \\ j \neq k}}^{M} \left\| h_{ki}(h_{ij}(h_{jk}(x))) - x \right\|^2$$

Similarly to inverse consistency error, transitivity error is a measure of consistency of the correspondence defined by compositions of transformations. For example, transitivity measures how similar a transformation that maps image A to B, and subsequently to image C, to a transformations that maps image A directly to C. Another demonstration of perfect transitivity is when the composition of transformations from A to B to C to A projects a point from image A back to its original position.

3.2.4 Evaluation vs. Known Transformation

Supposing the true correspondence between a pair of images is known, the task of evaluating the performance of an image registration algorithm mapping image correspondence becomes a simple task of comparing the known correspondence map to the algorithm-estimated correspondence map. While the true correspondence map between a pair of images is difficult to find in practice, a common method used to measure image registration performance is to generate image pairs using known transformations generated based on some reasonable assumptions of shape variability of real anatomies. Figure 3.7 illustrates how a known transformation can be used to evaluate the performance of image registration algorithms.

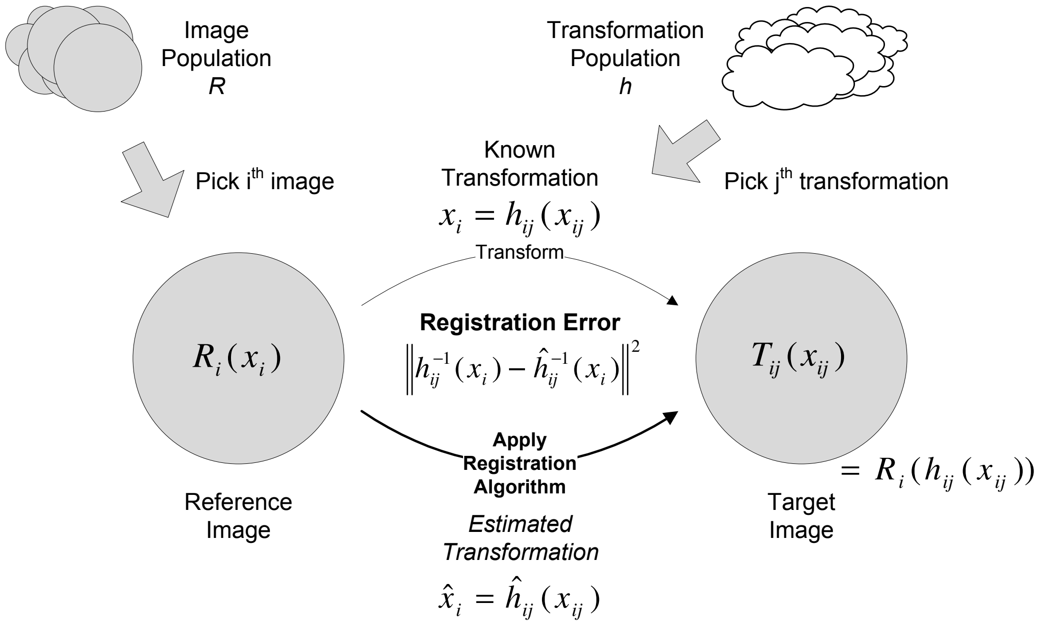


Figure 3.7. Schematic of evaluating registration algorithm performance using known transformations.

A population of images $R = \{R_1, R_2, \ldots, R_M\}$ is used as reference images to minimize the effect of reference image choice to the evaluation; and likewise, a population of random transformations $h = \{h_1, h_2, \ldots, h_N\}$ is used. To evaluate the performance of an algorithm, a reference image R_i is picked from R and a known transformation h_j is picked from H. The known transformation h_j is used to transform R_i into target image T_{ij}. Then the algorithm under evaluation is used to estimate the transformation from T_i to T_{ij} to produce the estimated transformation $\hat{h}_{ij}$.

The error between the estimated transformation and the known transformation can be computed using the following:

The average known transformation error over a region of interest $A \subseteq \Omega$ is defined as

$$AKTE_{ij}(A) = \frac{\int_A \left\| h_j(x) - \hat{h}_{ij}(x) \right\|^2 dx}{\int_A dx}, \tag{3.17}$$

where $\|\cdot\|$ is the standard Euclidean norm.

The maximum known transformation error in a region of interest A is defined as

$$MKTE_{ij}(A) = \max_{x \in A} \left\| h_j(x) - \hat{h}_{ij}(x) \right\|^2. \tag{3.18}$$

It is useful to plot the magnitude of the transformation error as an image to see the spatial distribution of the largest errors. For example, the magnitude error image shows whether or not the errors are distributed evenly across the image or if they only occur at the edges of objects.

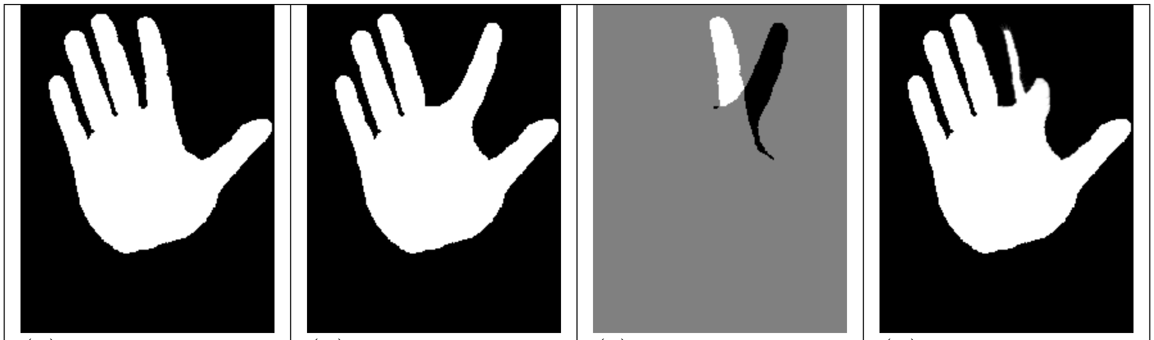
Still another way to illustrate the agreement of the estimated and known transformations is to plot the histogram of transformation errors $\left\| h_{ij}(x) - \hat{h}_j(x) \right\|^2$, $\forall x \in A$, i.e., generate the histogram of the transformation error magnitude for all of the points in the region of interest.

3.3 Shape Collapse Evaluation

3.3.1 Introduction

In [25], Durumeric et al. have addressed the shape collapse problem that affects both small and large-deformation volumetric image registration algorithms. In short, the shape collapse problem is a problem that may occur when an appendage of a foreground object in a moving image does not overlap with the foreground object in the fixed image and the registration process causes the appendage to collapse to a set of zero measure. Conversely, the shape collapse problem can also be seen as a shape growth problem - i.e., background in the moving image collapses and produces the effect of foreground object in the fixed image "growing into" an appendage. Figure 3.8 illustrate both the foreground collapse and background collapse problem. The natural solution to the hand registration problem illustrated is to simply rotate the index finger along the base joint while preserving the rigid shape of the finger. However, deformable registration algorithms that employ greedy optimization schemes will invariably choose a path of steepest gradient that optimizes the cost function. By choosing to do so, the registration process forces the index finger in the moving image (a non-overlapping appendage) to collapse into a set of zero measure while the background region where the index finger is located in the fixed image collapses and pulls the surrounding foreground into the region (causing a finger to grow out of the hand.) The collapse problem illustrated by this simple example addresses a more complex collapse problem that has often been neglected in the past.

As illustrated in Fig. 3.9, it is plausible that in real life image registration applications such as the alignment of the cortex, shape collapse may occur. A shape collapse in such application causes a serious problem, as an activation region in one brain mapping to a collapsed region in another brain would result in incorrect correspondence. In fact, many such shape collapses had been observed in the past, but



(a) Moving image (b) Fixed image (c) Difference before registration (d) Deformed moving image

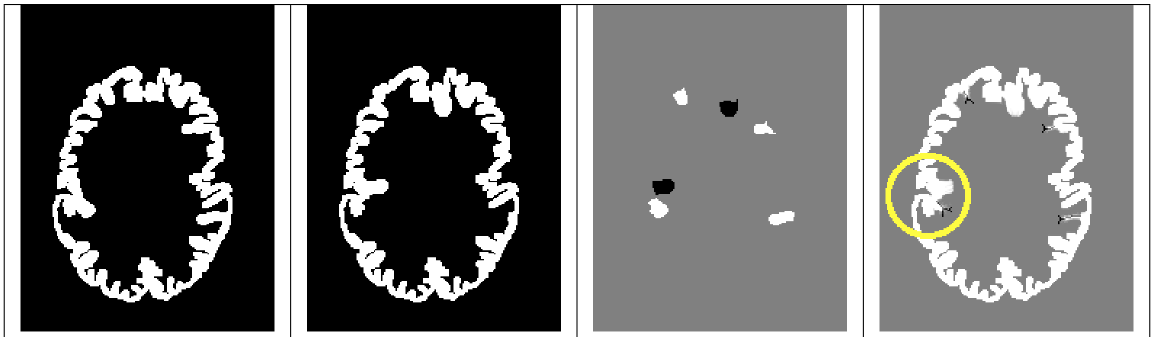
Figure 3.8. Example of an undesirable shape collapse during volumetric image registration.

often were counteracted by regularization or neglected entirely. In this work, the shape collapse problem is addressed in a formal manner and show how shape collapse regions can be predicted, and how actual shape collapse can be detected and measured.

In the following sections, method for predicting shape collapse regions based on mathematical understanding of shapes and optimization methods will be outlined. Then 2D and 3D phantoms of various shapes will be used to show how shape collapse occurs with one of the most popular registration suites available. Finally, comparisons will be made between predicted shape collapse points and actual shape collapse points and theoretical reasons behind the difference will be discussed.

3.3.2 Prediction of Collapse Points

In order to understand the mechanism of shape collapse, a formal definition of shape collapse for a pair of binary images will first be discussed. Let $I_1 : \Omega \to \mathbb{R}$



(a) Moving image

(b) Fixed image

(c) Difference before registration

(d) Difference after registration with annotated collapse region

Figure 3.9. Example of how shape collapse may occur in real life image registration applications, such as the alignment of the cortex illustrated here.

and $I_2 : \Omega \rightarrow \mathbb{R}$ be the moving and fixed binary images to be registered, respectively.

Definition 3.1. For any $V \subset \Omega$, the interior of V, denoted $\mathrm{Int}(V)$, is the union of all open subsets of Ω contained in V. The exterior of V, denoted $\mathrm{Ext}(V)$, is the union of all open subsets of Ω contained in $\Omega - V$. The boundary of V, denoted by ∂V, is the set of all points of Ω that are in neither $\mathrm{Int}(V)$ nor $\mathrm{Ext}(V)$ [25].

We denote subsets $V_1 \subset \Omega$ and $V_2 \subset \Omega$ as foregrounds of I_1 and I_2, respectively.

Definition 3.2. The overlap of the foreground objects V_1 and V_2 is denoted as $W = V_1 \cap V_2$ and the non-overlap region $U = U_1 \cup U_2$ where $U_1 = V_1 - V_2$ and $U_2 = V_2 - V_1$ [25].

Figure 3.10 illustrates how V_1 and V_2 relate to each other. The foreground collapse problem may occur when V_1 has an appendage which is not included in V_2

(i.e., U_1), and the growth problem (or alternatively, background collapse problem) occurs when V_2 has an appendage which is not included in V_1 (i.e., U_2).

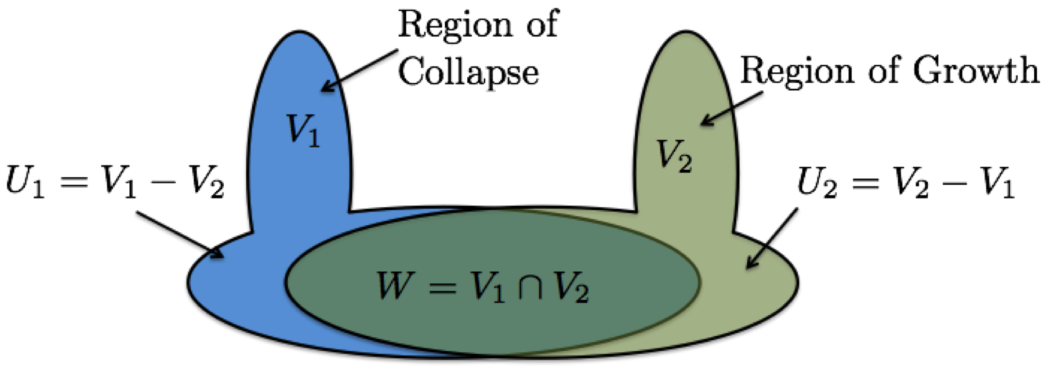


Figure 3.10. Overlap of two foregrounds, V_1 and V_2, of a pair of images being registered together. Regions U_1 and U_2 are possible collapse regions.

Next, we define the topological skeleton of the foreground as illustrated in Fig. 3.11 as the following.

Definition 3.3. Let p be a point, and V be a subset of a metric space (Ω, d). Let $B_r(p)$ denote the open metric balls $\{x \in \Omega : d(x,p) < r\}$. A closed ball $B \subset V$ is called a maximal ball of V, if for every closed ball B', $B \subseteq B' \subseteq V$, one has $B = B'$. The set $\{q \in V : \exists r > 0, B_r(q) \text{ is maximal ball of } V\}$ is defined to be the skeleton $S(V)$ of V by maximal balls [60, 25].

Figure 3.12 illustrates possible configurations of skeletal and boundary points. For every skeletal point $q \in S(V_1)$, there is a unique maximal ball $B_r(q)$ centered at q, and the set of points along the boundary ∂V_1 associated to q is $A_\partial(q) := B_r(q) \cap \partial V_1$.

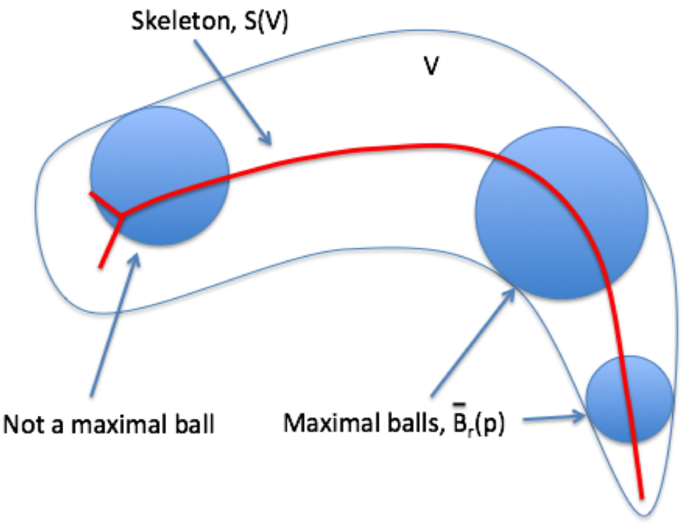


Figure 3.11. The skeleton (red line) of foreground object V is defined as the set of centers of all the maximal balls contained in V.

The association set $A_\partial(q)$ is nonempty; there usually are two (e.g., q_2) or more (e.g., q_3 or q_1) boundary points associated with the maximal ball. Only one boundary point may be associated with a maximal ball, such as the focal point of a boundary curve (e.g., p_4). It is also possible that a single boundary point is associated with more than one skeletal point if ∂V_1 is not differentiable at $p \in \partial V_1$ (e.g., p_2).

For simplicity, we will first discuss the shape collapse of a simple binary shape V_1, such as the rectangle in Fig. 3.13, registering to a zero-set $V_2 = \varnothing$ with no regularization constraint, and assume that ∂V_1 is a piecewise C^1 closed curve. For a binary image, a greedy algorithm that minimizes the similarity-cost $C = \|I_1(\phi(x)) - I_2(x)\|^2$ follows the cost gradient in the direction that decreases the area of V_1 in the fastest way. The Euclidean distance function $d : V_1 \to [0, \infty)$ to the boundary ∂V_1, is given by $d(x, \partial V_1) = \inf \{|x - y| : y \in \partial V_1\}$. The gradient of the distance function, ∇d

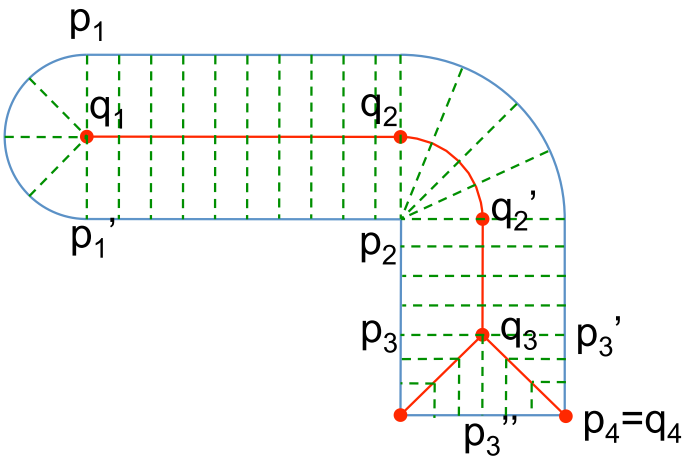


Figure 3.12. Possible configurations of the skeletal and boundary points. A skeletal point can correspond to two (q_2), three (q_3) or more (q_1) boundary points. Similarly, a boundary point can have one (p_4) or many (p_2) associated skeletal points.

(when it exists), is perpendicular to the level sets of d (as shown in different shades of blue in Fig. 3.13).

Definition 3.4. Let $L(p, q)$ denote the line segment with end points $p \in \partial V_1$ and $q \in S(V_1)$, and $L^\circ(p, q) = L(p, q) - \{p, q\}$ (shown as green dotted lines in Fig. 3.12) [25].

By Proposition 1 in [25], we have that if $a \in L^\circ(p, q)$, then p is the unique closest point of ∂V_1 to point a. Thus, the fastest area decreasing flow in the interior of V_1 is along $L^\circ(p, q)$ with unit speed until the skeleton is reached. The flow is not definable along the skeleton.

Without regularization, the area decreasing flow in the continuous space will reduce V_1 to $\varnothing$ since it will remove the skeleton along the way (as shown in Fig. 3.14) because the deforming forces come from different directions along $L(p, q)$. With reg-

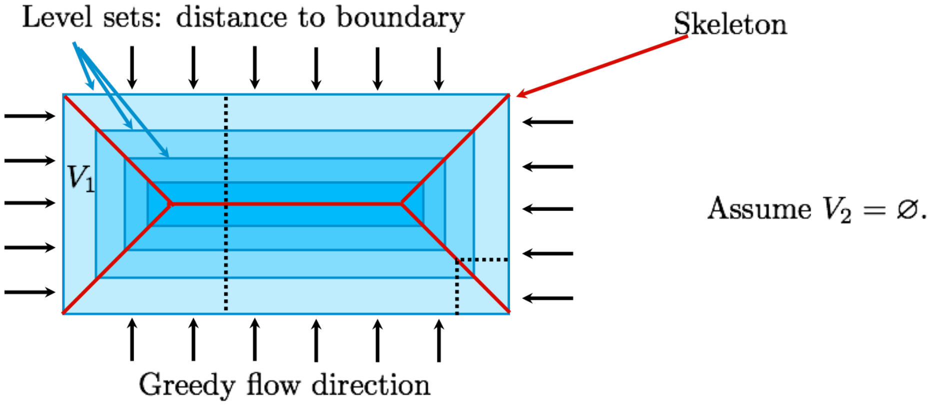


Figure 3.13. A simple binary shape collapse example where V_1 is a rectangle (blue region) and V_2 is a zero-set. The red lines in V_1 is the topological skeleton of V_1. The different shades of blue in V_1 shows the Euclidean distance-to-boundary level sets.

ularization, this effect will be reduced by averaging and slowing down the area reduction.

In the discrete case, because the skeleton has zero measure, removing the skeleton has little or no effect in minimizing the similarity cost, and thus the skeleton remains (as shown in Fig. 3.15). Generally speaking, the skeleton reached through the registration process does not necessarily be the same as $S(V_1)$, but it is a very good approximation.

Figures 3.16 and 3.17 illustrate actual shape collapse observed by registering a binary rectangle image rotated by 30° (Fig. 3.16(a)) and a binary fish image (Fig. 3.17(a)) to a $\varnothing$ image (Fig. 3.16(b)), using a diffeomorphic registration method with zero regularization, provided by the ANTs package [6, 5]. We observe that both shapes deform completely into what appears to be $\varnothing$ (Figs. 3.16(c) and 3.17(c)), further analysis of the X- and Y-displacement fields generated by the registration al-

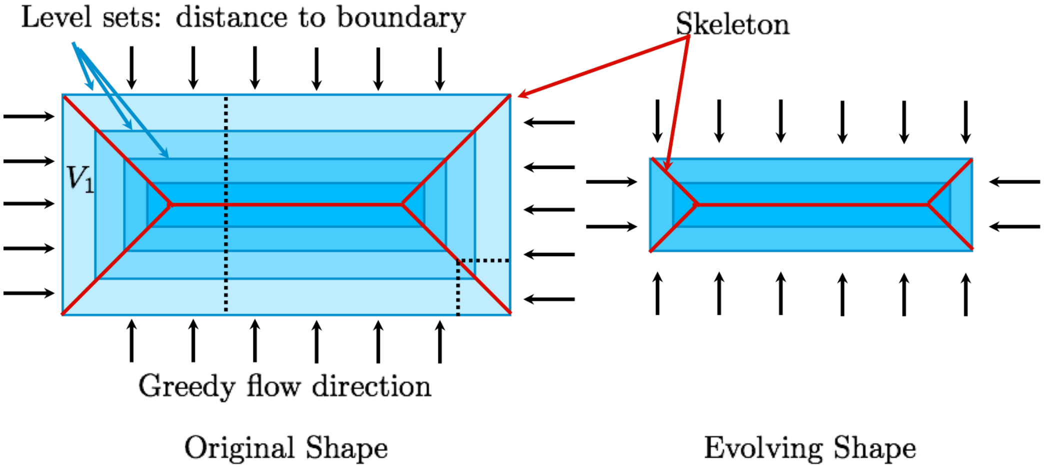


Figure 3.14. In the continuous area-reduction case, V_1 will reduce to $\varnothing$ without regularization since it will remove the skeleton along the way.

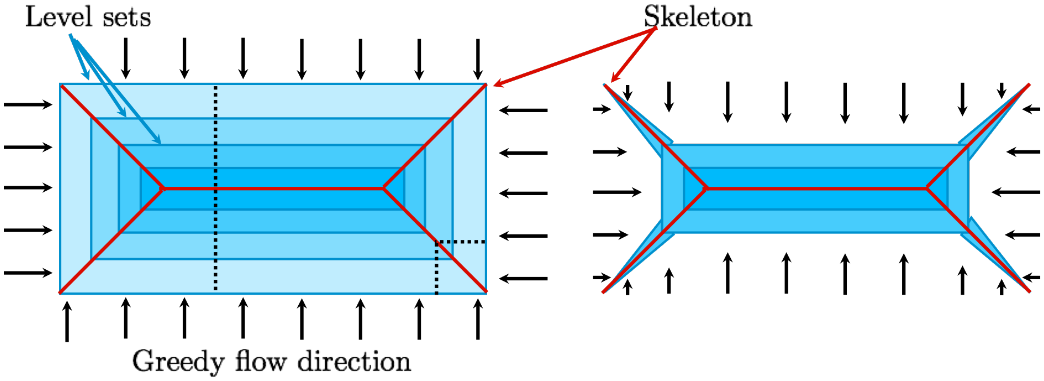


Figure 3.15. In the discrete area-reduction case, because removing the skeleton has little or no effect in minimizing the similarity cost, the skeleton remains.

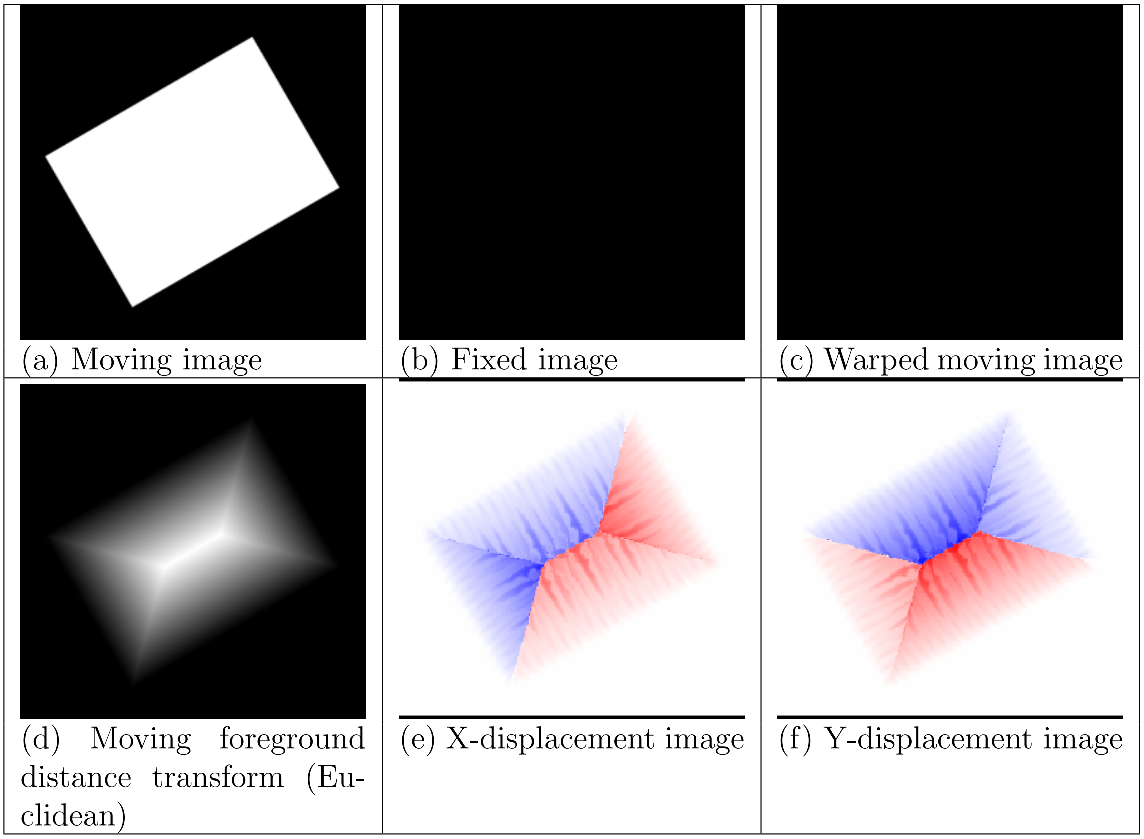
gorithm (Figs. 3.16(e)-(f) and 3.17(e)-(f)) reveals that each shape actually collapsed along the paths perpendicular to the level sets of the Maurer distance transforms [47] shown in Figs. 3.16(d) and 3.17(d). In the continuous space, the skeletons of each shape would have remained after deformation, but because of discretization, the resulting image appears to be $\varnothing$. Note that the X- and Y-displacement fields, with negative values colored in blue and positive values colored in red, shows that the gradient flow directions of the registration process are perpendicular to the level sets of the distance transforms, as predicted above.

Next, shape collapse for image pairs where both V_1 and V_2 are non-zero sets will be described.

Definition 3.5. A point $p_1 \in \partial V_1$ is a collapsing point for $q \in S(V_1)$, if $\exists p_2 \in \partial V_1$ such that $p_1 \neq p_2$, $\{p_1, p_2\} \subset A_\partial(q)$, and $L(p_1, q) \cup L(p_2, q) \subset U_1$ [25].

Figure 3.18 illustrates both collapsing and non-collapsing points for a pair of partially overlapping foregrounds V_1 and V_2. Here, p_1 is a collapsing point because there exists another point p_2 that satisfies all the conditions listed in Definition 3.5: $p_1 \neq p_2$, $\{p_1, p_2\} \subset A_\partial(q)$, and $L(p_1, q) \cup L(p_2, q) \subset U_1$. Likewise, p_2 is a collapsing point for the same reason. However, even though there exists a p_2' that satisfies $p_1' \neq p_2'$ and $\{p_1', p_2'\} \subset A_\partial(q')$, p_1' is not a collapsing point because $p_2' \notin U_1$. During registration, points p_1 and p_2 will move along $L^\circ(p_1, q)$ and $L^\circ(p_2, q)$ respectively, until they approach the skeletal point q. On the other hand, point p_2' will approach ∂V_2 provided that $L^\circ(p_2', q)$ enters V_2 at most once.

Figure 3.19 illustrates another scenario where both associated points are lo-



(a) Moving image

(b) Fixed image

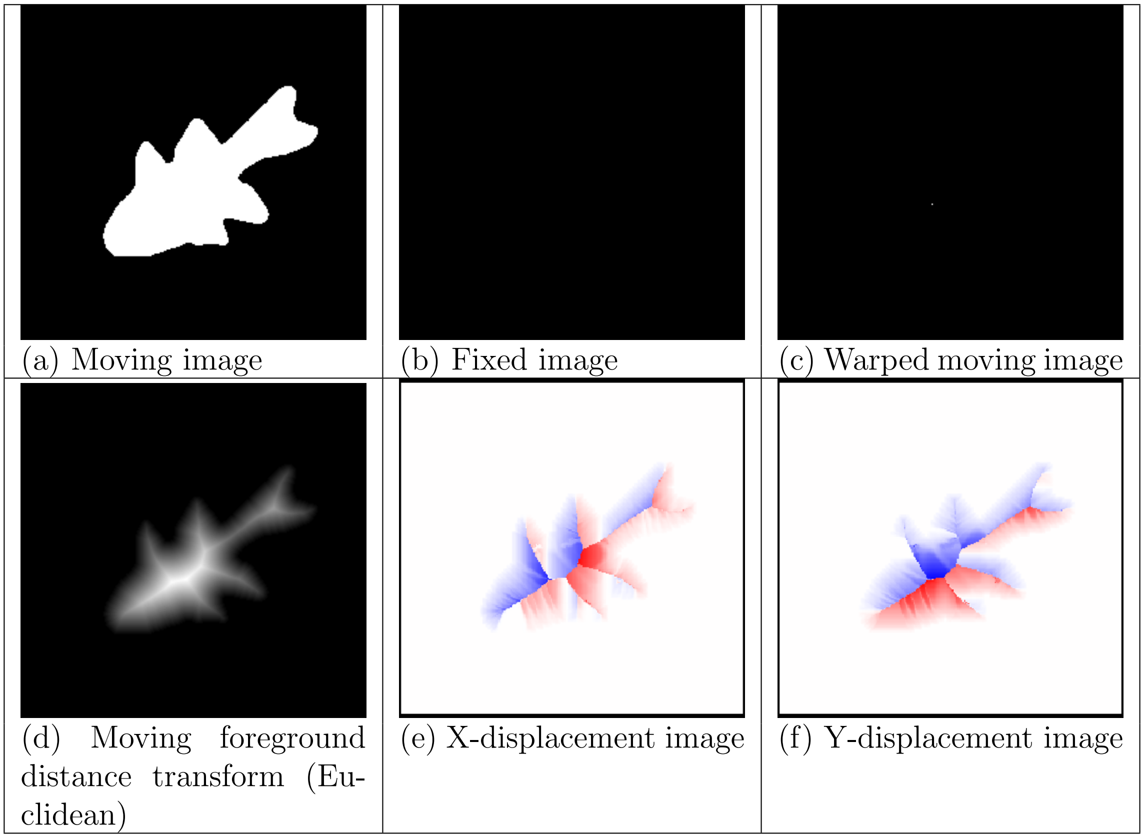
(c) Warped moving image

(d) Moving foreground distance transform (Euclidean)

(e) X-displacement image

(f) Y-displacement image

Figure 3.16. Shape collapse of a rectangle rotated by 30° registering to ∅ using time-varying velocity field transformation with no regularization. Blue in the displacement images represent negative values while red represents positive values. Observe that the gradient flow direction is perpendicular to the level sets of the distance transform, which align neither with the X- nor the Y-axis.



(a) Moving image

(b) Fixed image

(c) Warped moving image

(d) Moving foreground distance transform (Euclidean)

(e) X-displacement image

(f) Y-displacement image

Figure 3.17. Shape collapse of a fish shape registering to ∅ using time-varying velocity field transformation with no regularization. Blue in the displacement images represent negative values while red represents positive values. Notice that even with a complex skeleton, the gradient flows in the direction perpendicular to the level sets of the distance transform (d).

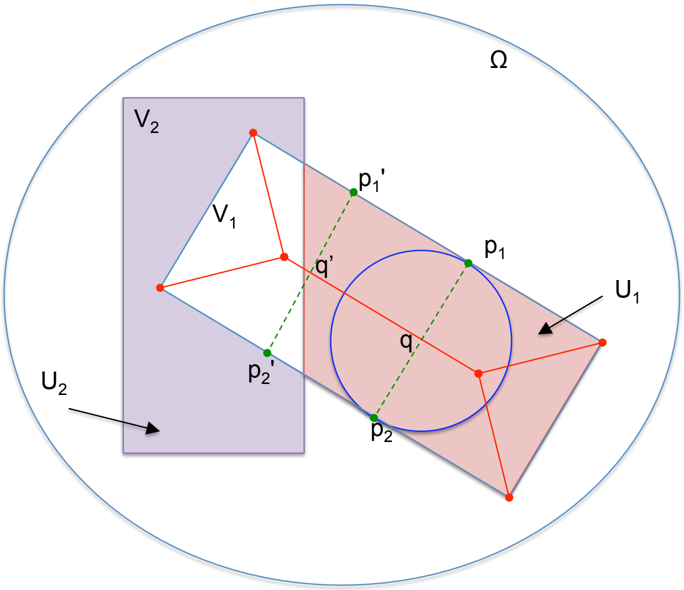
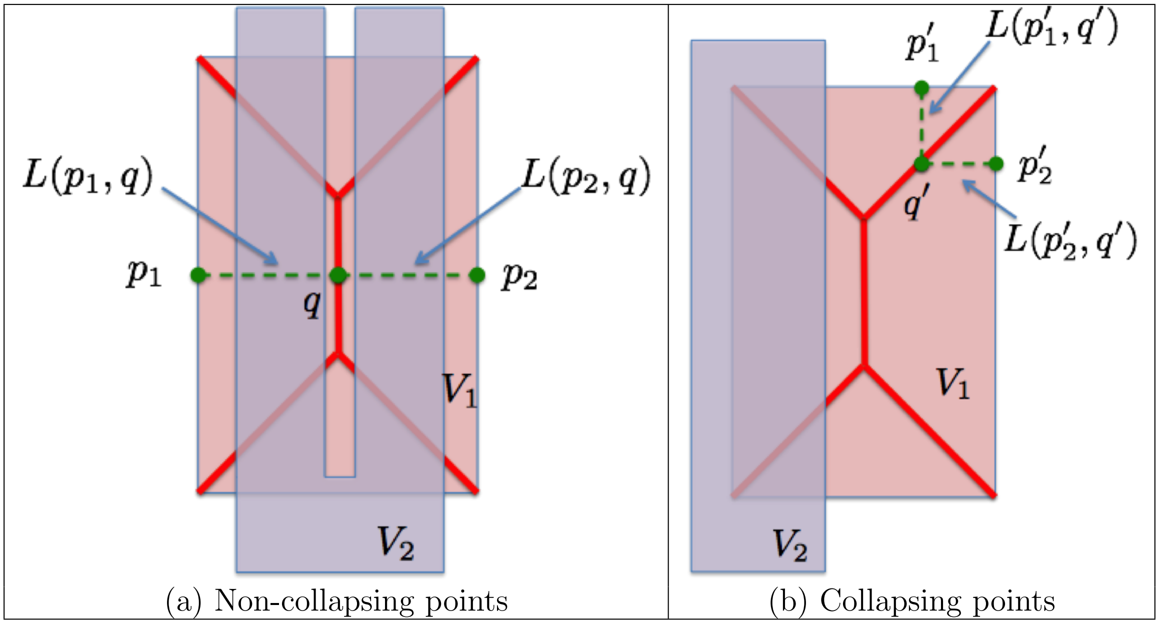


Figure 3.18. Collapsing and non-collapsing points. p_1 is a collapsing point because there exists another point p_2 that belongs to the same maximal ball $B_r(q)$, and both p_1 and p_2 are in the non-overlap region U_1. Likewise, p_2 is also a collapsing point. In contrast, p_1' is not a collapsing point because although p_2' belongs to the same maximal ball $B_r(q')$, p_2' is not in U_1.

cated in the non-overlapping region U_1, but they are non-collapsing points in one situation and collapsing points in another. Points p_1 and p_2 in Fig. 3.19(a) are non-collapsing points, because although both points are located in the non-overlap region U_1, their distance transform lines $L(p_1, q)$ and $L(p_2, q)$ are not contained in U_1. In contrast, points p_1' and p_2' are collapsing points because lines $L(p_1', q')$ and $L(p_2', q')$ (and therefore also points p_1' and p_2') are contained in U_1, satisfying all conditions listed in Definition 3.5.



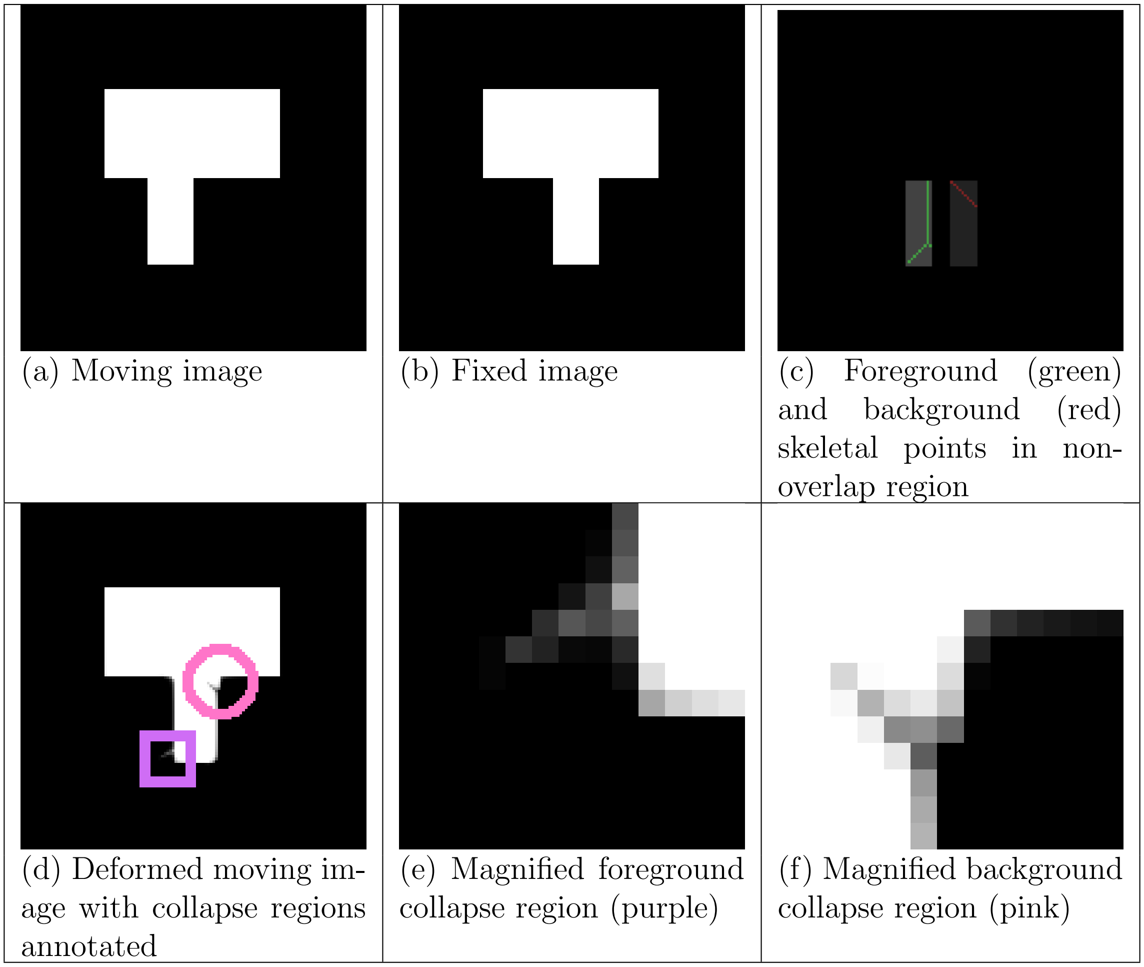
(a) Non-collapsing points

(b) Collapsing points

Figure 3.19. (a) p_1 and p_2 are not collapsing points because although both points are contained in the non-overlap region U_1, the lines $L(p_1, q)$ and $L(p_2, q)$ are not. (b) p_1' and p_2' are collapsing points because lines $L(p_1', q')$ and $L(p_2', q')$ (and therefore also points p_1' and p_2') are contained in U_1.

Finally, it can be seen that the growth in the foreground of the moving image (as demonstrated by the hand example), is really a collapsing of the background. By computing the set of skeletal points of the background $q \in S(V_1^{\complement})$ and the boundary points of the background $p \in \partial V_1^{\complement}$, the set of collapsing points in the non-overlapping region U_2 can be found in a similar way as described in Definition 3.5. Figure 3.20 illustrates a registration example where foreground and background collapse occurs simultaneously. Non-overlap region U_1 is shown in bright gray in (c), whereas non-overlap region U_2 is shown in dark gray. Within U_1 and U_2, foreground and background skeletal points about which shape collapse is likely to occur is shown in green and red, respectively. Figures (d)-(f) illustrate the actual foreground and background shape collapse after registration, and we can verify that the non-overlap regions either collapsed toward the skeletal points or deformed towards the target boundary, as predicted.

A shape collapse prediction software was developed in C++ that takes both 2D images and 3D volumes and their foreground and (if present) background skeleton data as inputs. The software determines if a shape boundary is a collapsing boundary point based on the shape collapse criterion outlined above. The software writes out foreground and background boundary collapse points and their associated skeletal points as images. Figures 3.20(c), 4.13(c), 4.17(f,g) and 4.19(f,g) are examples of predicted collapsing boundary points and their associated skeletal points generated by this software.



(a) Moving image

(b) Fixed image

(c) Foreground (green) and background (red) skeletal points in non-overlap region

(d) Deformed moving image with collapse regions annotated

(e) Magnified foreground collapse region (purple)

(f) Magnified background collapse region (pink)

Figure 3.20. A registration example where foreground and background collapse occurs simultaneously. Foreground and background skeletal points about which collapse is likely to occur is shown in (c). Actual shape collapse resulting from registration is annotated in (d)-(f).

3.3.3 Measuring Collapse After Registration

The previous section showed a method for predicting collapse points independent of registration algorithm used. In this section, a method for detecting actual shape collapse caused by registration is outlined. While predicted collapse points reveal likely locations where shape collapse may occur, actual shape collapse may occur in different locations at different levels depending on the registration algorithm used. The traditional method for measuring local expansion and contraction is the Jacobian [11, 17, 14], but the Jacobian does not differentiate if a neighborhood of points map to a single contiguous neighborhood of points, or multiple non-contiguous neighborhoods. Therefore, a new method for measuring the level of dispersion a small neighborhood of displacements has in order to detect if a shape collapse had occurred.

The underlying idea behind detecting shape collapse is that if the set of points sampled from a small neighborhood in the deformed image maps to more than one non-contiguous neighborhoods, it is very likely that a collapse had occurred. In other words, a neighborhood of contiguous points in the deformed image should originate from the same general neighborhood of contiguous points in the moving image if no collapse had occurred. K-means clustering is a popular vector quantization method which aims to partition N observations into K clusters in which each observation belongs to the cluster with the closest mean. In order to measure the level of separation a neighborhood of points has when mapped back to the moving image coordinate system, k-means clustering was used as described below.

For the sake of simplicity, collapse detection using k-means clustering for 2D case will be explained first (See Fig. 3.21). Let $P = \{p_1, p_2, \ldots, p_9\} \subset \mathbb{R}^2$ be a set of points in a 3×3 neighborhood in the deformed image coordinate system, centered around the center point p_5. Let $S = \{u(p_1), u(p_2), \ldots, u(p_9)\} \subset \mathbb{R}^2$ be the set of displacement vectors for each of the points in P. The k-means clustering seeks to partition the set S into k partitions, $S_1, S_2, \ldots, S_k$ to minimize the within-cluster sum of squares (WCSS) (sum of distance functions of each point in the cluster to the k-centers $\mu_1, \mu_2, \ldots, \mu_k$). For $k = 2$, this is achieved by solving

$$\min_S \sum_{i=1}^{2} \sum_{x \in S_i} \|x - \mu_i\|^2 \tag{3.19}$$

where

$$\mu_i = \frac{1}{|S_i|} \sum_{x \in S_i} x \tag{3.20}$$

is the k-center associated to partition S_i and $i = \{1, 2\}$.

Figure 3.21 shows an example of a collapse at point p_5. The arrows represent the Eulerian displacements in a neighborhood of the pixel p_5. An Eulerian displacement pulls back the coordinate in the moving image at the tip of the arrow to the base of the arrow in the fixed image coordinate system.[1] Fig. 3.21(b) shows that two disjoint regions from the moving image are being pulled back to a single connected neighborhood in the fixed image. This by definition is a collapse of the moving image at point p_5.

Once the displacements are partitioned into their respective clusters, the dis-

[1] The direction of the arrow can be confusing because it looks like the displacement moves in the direction opposite to the arrow. The correct way to interpret this figure is that the arrows point to the location in the moving image coordinate system that gets pulled back to the base of the arrow in the fixed image coordinate system.

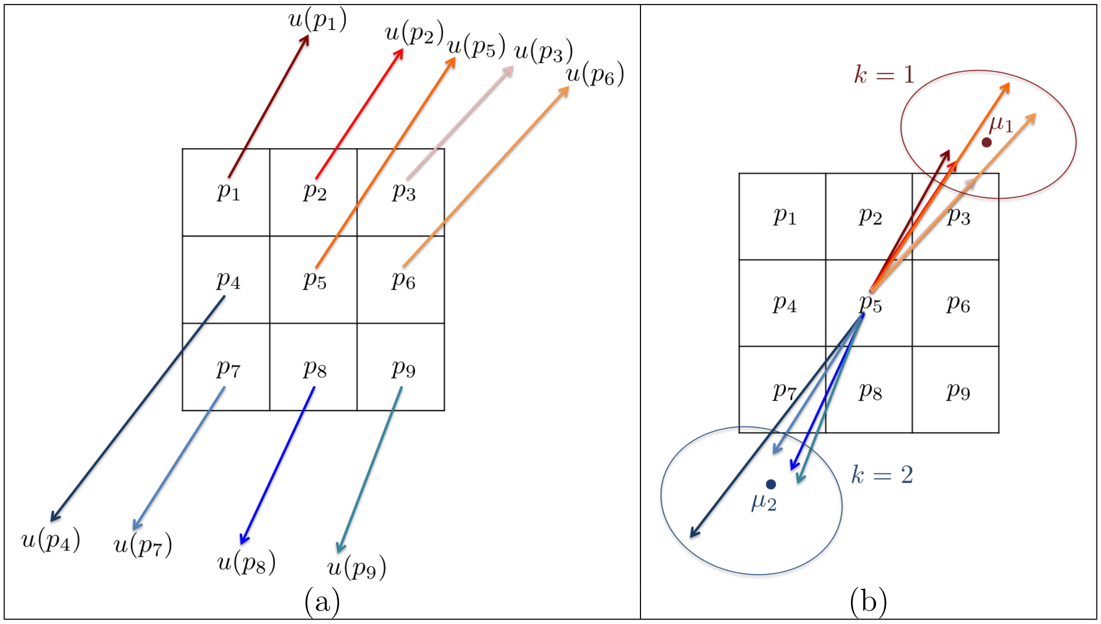


Figure 3.21. (a) A 3 × 3 neighborhood of displacements, centered around p_5. (b) After translating all the neighborhood displacements to the center, perform a k-means clustering ($k = 2$) to measure the level of dispersion. The further apart μ_1 and μ_2 are, the more likely that a shape collapse occurred in the neighborhood.

tance of the k-centers $d = \|\mu_2 - \mu_1\|$ is measured to determine whether there was a shape collapse at this neighborhood of points or not. If the means of the clusters are distant apart, this indicates that the level of dispersion of the displacement vectors at this location is high which makes this location a sort of a “sink,” where a region of wide area collapsed into a small area. Figure 3.22(a) and (c) show the predicted skeletal points of Fig. 3.16(a) and Fig. 3.17(a) being registered to $\varnothing$ about which shape collapse is likely to happen. Using the displacement fields generated by the time-varying velocity field registration method as shown in Figs. 3.16(e)-(f) and Figs. 3.17(e)-(f), the k-means clustering method was applied to detect and measure the actual shape collapse observed as shown in Figs. 3.22(b) and (d) (bright pixels).

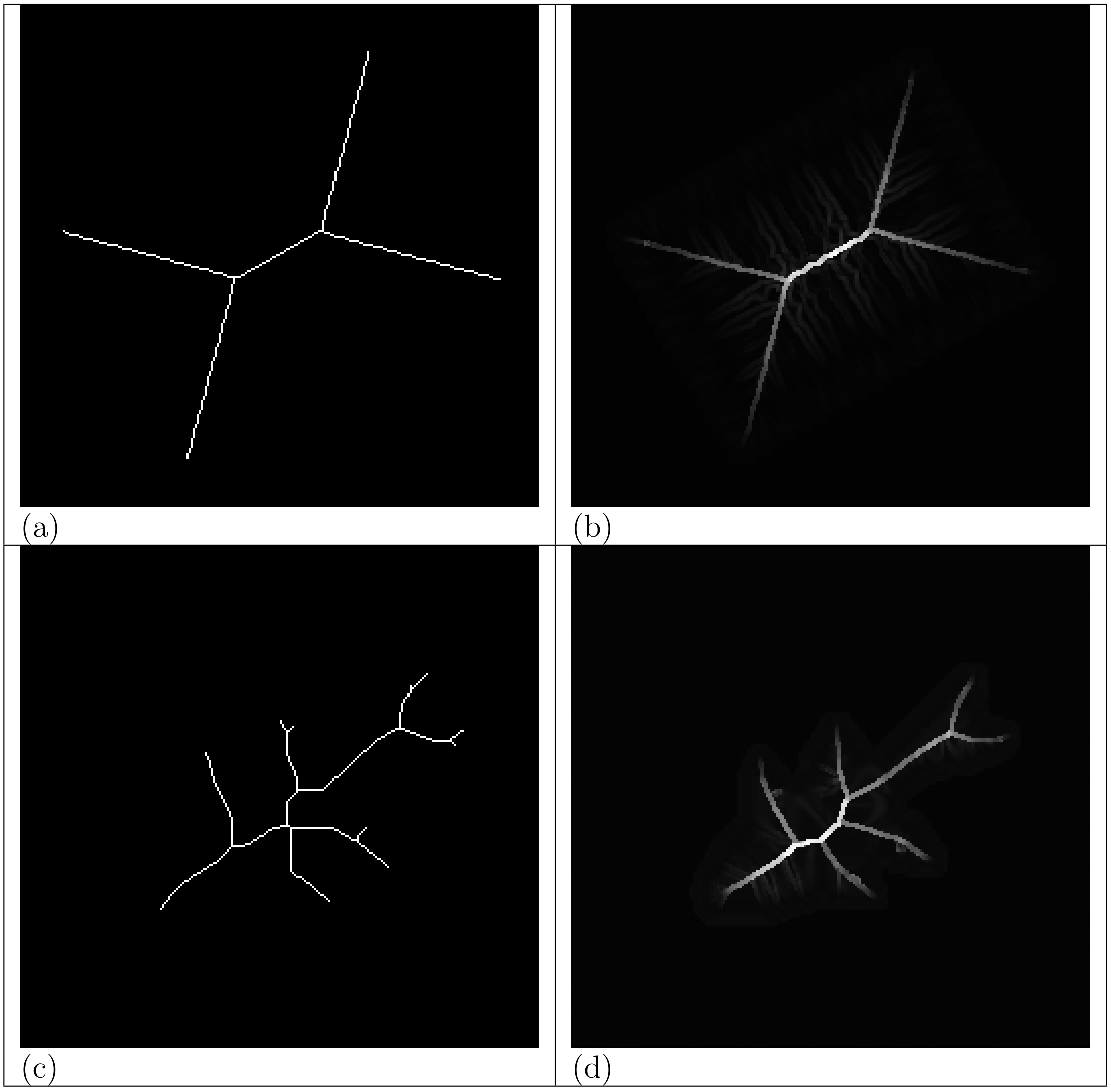


Figure 3.22. (a) & (c) Predicted skeletal points about which shape collapse is likely to occur. (b) & (d) Actual shape collapse observed using k-means clustering (bright pixels).

It can be observed from Fig. 3.22(b), the rectangle collapsed along the predicted skeletal points, with the strongest collapse happening around the center of the rectangle (brightest pixels) and the weakest collapse happening around the corners of the rectangle. This can be explained with the radius of the maximal ball associated with these locations: the maximal ball has the greatest radius at the center of the rectangle and the smallest radius around the corners of the rectangle. As for the fish example in Fig. 3.22(c-d), we can observe that shape collapse regions that the k-means clustering method shows a result that differs from the predicted skeletal points, particularly around the bottom two fins. In fact, the shape collapse pattern resembles what is shown in the distance transform in Fig. 3.17(d) more than the topological skeleton. This shows that while the prediction of collapse regions can be made independent of the registration algorithm used, actual shape collapse may differ from the prediction as demonstrated here.

Next, to verify that the k-means clustering method works both for foreground and background collapse, it was applied on the "tooth" example shown in Fig. 3.20. From comparing the predicted collapse region and the detected collapse region, we find that the method works equally well for foreground and background regions. However, there is something interesting worth noting here - the k-means clustering method also detected shape collapse along the target boundary for this particular registration example. Analysis of the X- and Y-displacement fields verify that the deformation terminated abruptly at the target boundary, which should not happen with smooth deformation. In fact, this is another kind of shape collapse that can occur due to the absence of regularization in the registration process. The nature of this type of shape collapse will be discussed in the next section.

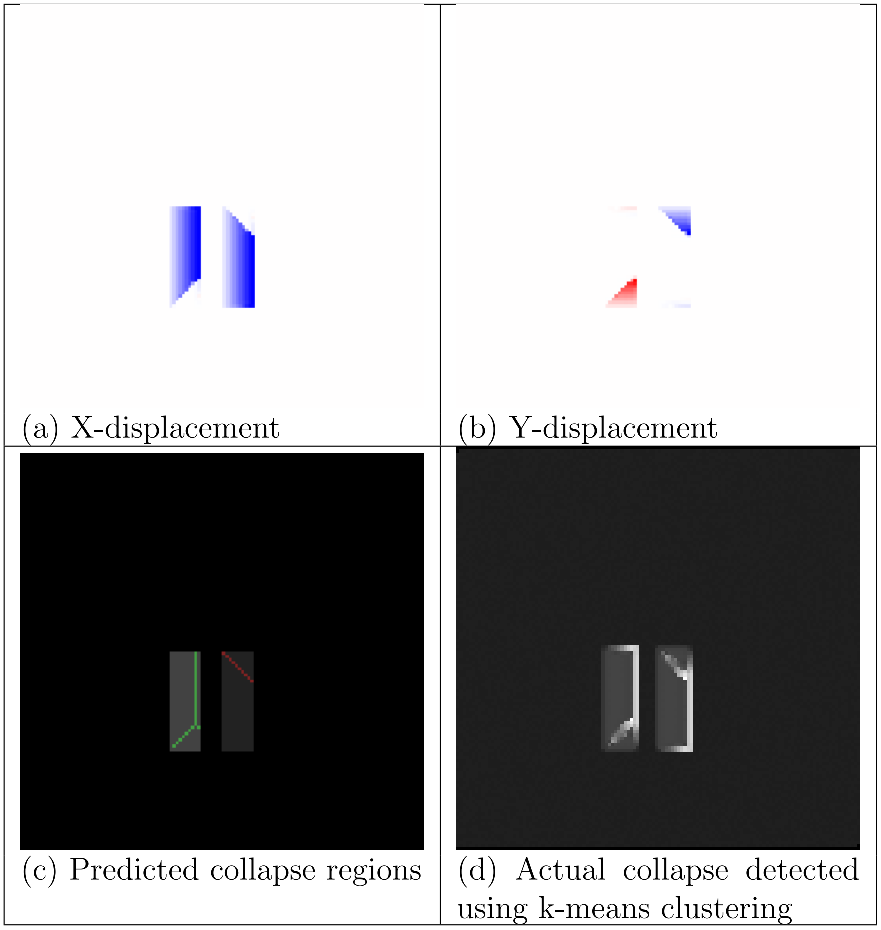

(a) X-displacement
(b) Y-displacement
(c) Predicted collapse regions
(d) Actual collapse detected using k-means clustering

Figure 3.23. K-means clustering applied on the “tooth” registration example. The method works for foreground and background collapse equally well. Notice that shape collapse is also detected in the target boundary with this particular registration algorithm.

The 2D k-means clustering method can easily be extended to 3D volumes by performing the same $k = 2$ clustering for a $3 \times 3 \times 3$ neighborhood. The choice of a

3×3 neighborhood (likewise, $3 \times 3 \times 3$ for volumes) showed to be sufficient to measure the level of dispersion on the neighborhood. Comparison between 3×3, 5×5 and 7×7 neighborhoods were made to see if larger neighborhoods made improvements to the measurement of dispersion as shown in Fig. 3.24. In general, it was found that the 3×3 neighborhood was sufficient to measure the level of dispersion, and located the collapsing points more precisely.

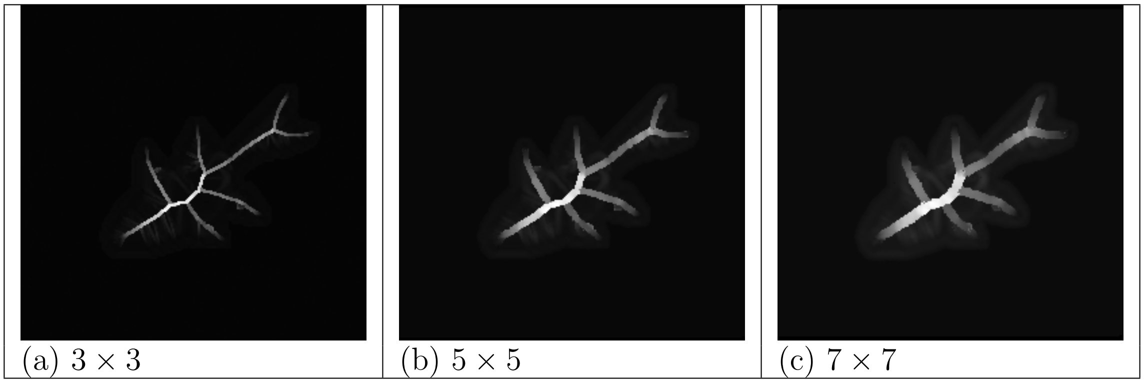

(a) 3×3 (b) 5×5 (c) 7×7

Figure 3.24. K-means clustering results using 3×3, 5×5 and 7×7 neighborhoods. 3×3 neighborhood was found to give the most precise location of the collapsing points.

CHAPTER 4

EXPERIMENTS

4.1 Evaluation Using Brain MRI Databases

Alignment of brain structures between different subjects is one of the many commonly employed applications of image registration, useful for mapping physiology, functionality and connectivity between subjects or to an atlas. To demonstrate how the evaluation methods described in this work can be used to evaluate image registration algorithms on real-life applications, the following experiments were conducted.

4.1.1 Data

The NA0 evaluation database consists of 16 brain MR volumes from 8 male and 8 female normal right-handed adults, drawn at random from a population of 240 normal subjects. These data sets were selected from a database of healthy individuals from the Human Neuroanatomy and Neuroimaging Laboratory (HNNL) at the University of Iowa. The demographics of the subjects are shown in Table 4.1. All 16 MR volumes were aligned along the AC-PC points prior to applying the non-rigid image registration algorithms. To evaluate structural alignment, segmentations of 32 grey matter regions of interest (ROIs) made and reviewed by experts at HNNL. The original segmentations were conducted in 2D plane, resulting in smooth edges in the plane of segmentation but rough when viewing oblique slices. Thus the segmentations were smoothed and verified by reviewers at HNNL. Some of these data sets and their segmentations appear in a new atlas by Hannah Damasio that illustrates the normal neuroanatomy of the human brain, and have been used in several publications about the morphometric analysis of the normal human brain [21]. Table 4.2 shows each ROI label and the average volume of the corresponding ROI in the NA0 database. Figure 4.2 shows an example of an NA0 image data overlaid with its ROI segmentations, viewed from three orthogonal views.

Table 4.1. Clinical demographic characteristics of the study population of NA0.

Label	Age	Gender	Race	Ethnic Category	Handedness
na01	43	Male	White	Non Hispanic	+95
na02	48	Male	White	Non Hispanic	+95
na03	28	Male	White	Non Hispanic	+85
na04	28	Male	Asian	Non Hispanic	+100
na05	32	Male	Unknown	Hispanic	+100
na06	27	Male	White	Non Hispanic	+80
na07	29	Male	White	Non Hispanic	+65
na08	25	Male	White	Non Hispanic	+100
na09	26	Female	White	Non Hispanic	+100
na10	27	Female	Asian	Non Hispanic	+100
na11	36	Female	White	Non Hispanic	+95
na12	26	Female	White	Non Hispanic	+85
na13	24	Female	Unknown	Hispanic	+100
na14	28	Female	White	Non Hispanic	+80
na15	30	Female	Black	Non Hispanic	+100
na16	41	Female	White	Non Hispanic	+100

4.1.2 Methods

Seven image registration algorithms were used to perform pairwise registrations of the NA0 dataset, and their names are as follows: SLE, SICLE, SyN, Diffeomorphic Demons, Log Symmetric Demons, elastix and NiftyReg. Default parameters for each registration software were used, since optimizing registration performance of

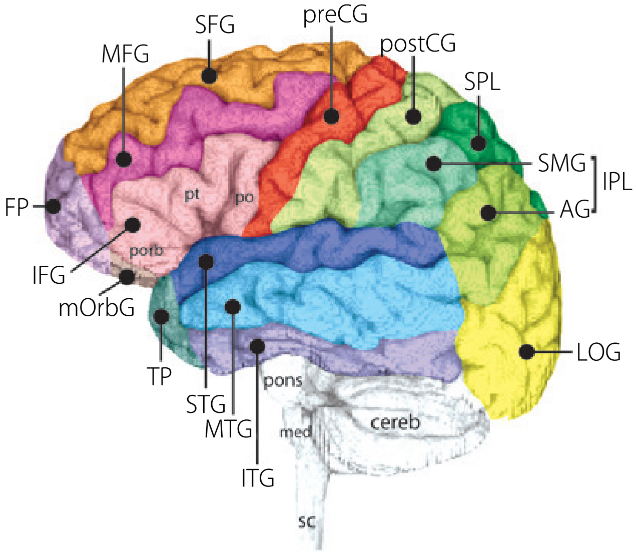
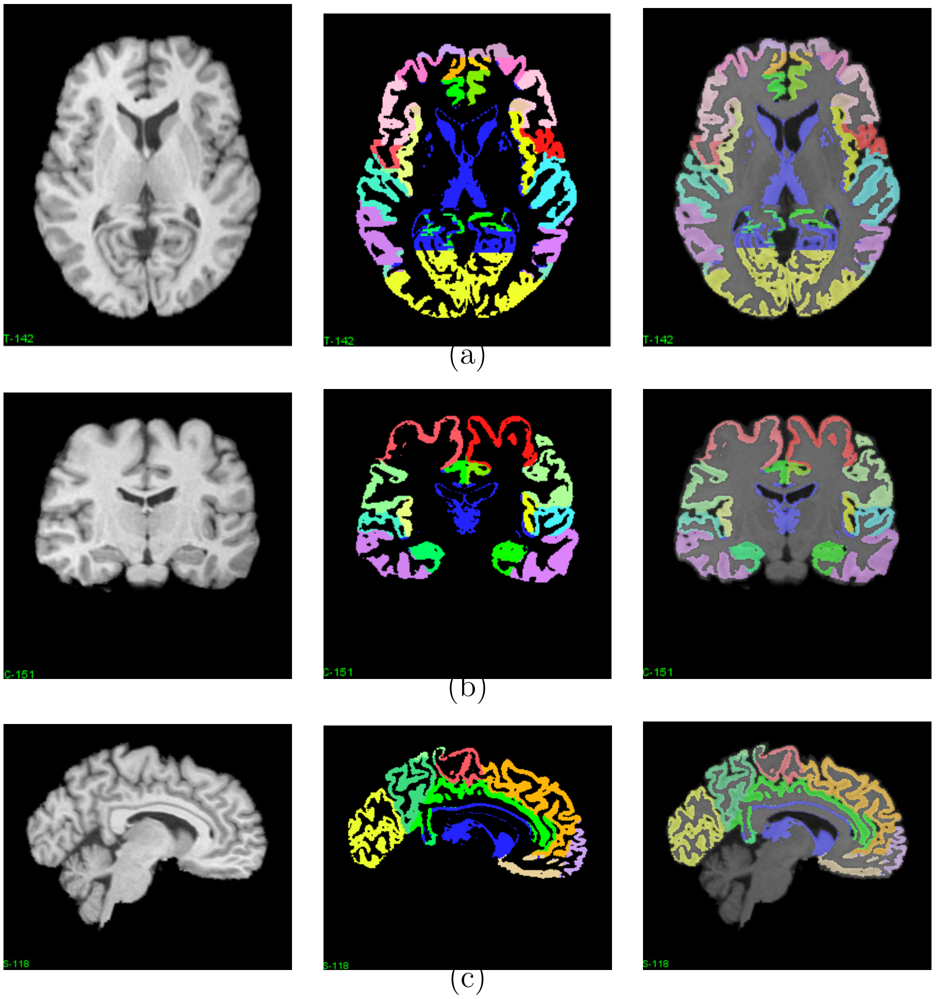


Figure 4.1. The segmentations available in NIREP NA0 (Detail of figure 2 of chapter 2, H. Damasio, “Human Brain Anatomy in Computerized Images,” 2^{nd} ed., 2004, Oxford University Press, In press). Shown are typical segmentations available to this project and include: Cerebrum: The cerebellum, hypothalamus, and brain stem are not segmented; Left and right hemispheres: Frontal Lobe: Frontal Pole, Superior Frontal Gyrus, Middle Frontal Gyrus, Inferior Frontal Gyrus, Orbital Frontal Gyrus, Precentral Gyrus; Parietal Lobe: Postcentral Gyrus, Superior Parietal Lobule, Inferior Parietal Lobule; Temporal Lobe: Temporal Pole, Superior Temporal Gyrus (including Heschl’s Gyrus (Primary Auditory Cortex) and Planum Temporale), Infero-Temporal Region, Parahippocampal Gyrus (including the Amygdala and Hippocampus); Occipital Lobe; Cingulate Gyrus; Insula.



(a)

(b)

(c)

Figure 4.2. An MR data set from NA0 database. Shown are the T1 image (left column), the segmentation image associated with the na01 data set (middle column), and the segmentation overlapped on the T1 image (right column) viewed in (a) transverse, (b) coronal and (c) sagittal slices.

Table 4.2. Regions of Interest (ROI) in the NA0 evaluation database. The average volume for each ROI is reported in units of voxels.

	ROI	ave volume $\times 10^5$		ROI	ave volume $\times 10^5$
1	L occipital lobe	0.87	2	R occipital lobe	0.93
3	L cingulate gyrus	0.42	4	R cingulate gyrus	0.45
5	L insula gyrus	0.22	6	R insula gyrus	0.21
7	L temporal pole	0.28	8	R temporal pole	0.31
9	L superior temporal gyrus	0.45	10	R superior temporal gyrus	0.39
11	L infero temporal region	1.0	12	R infero temporal region	1.0
13	L parahippocampal gyrus	0.35	14	R parahippocampal gyrus	0.34
15	L frontal pole	0.17	16	R frontal pole	0.18
17	L superior frontal gyrus	0.79	18	R superior frontal gyrus	0.78
19	L middle frontal gyrus	0.67	20	R middle frontal gyrus	0.64
21	L inferior gyrus	0.30	22	R inferior gyrus	0.32
23	L orbital frontal gyrus	0.46	24	R orbital frontal gyrus	0.4
25	L precentral gyrus	0.62	26	R precentral gyrus	0.62
27	L superior parietal lobule	0.64	28	R superior parietal lobule	0.60
29	L inferior parietal lobule	0.78	30	R inferior parietal lobule	0.82
31	L postcentral gyrus	0.45	32	R postcentral gyrus	0.42

each algorithm is not within the scope of this work (therefore the results reported in this work is not representative of the full potential of the registration software used). All pairwise combinations of 16 brain MR images resulted in 240 pairwise registrations for each algorithm. All registrations were performed on a Intel® Xeon® X5670 @ 2.93GHz (24 cores) with 47 GB of memory.

Evaluations of image registration performance were carried out using the following criteria: 1) Target Overlap, 2) Mean Overlap, 3) Union Overlap, 4) Volume Similarity, 5) False Positive, and 6) False Negative.

4.1.3 Results and Discussion

Evaluation of overlap measures were conducted using Target Overlap, Mean Overlap, Union Overlap, Volume Similarity, False Positive, and False Negative [65]. The overlap measures were computed for each individual regions of interest listed in Table 4.2, and also for all regions. Table 4.3 shows the Target Overlap, Mean Overlap, and Union Overlap measures of all regions for all 240 pairwise registrations for each registration algorithm. The overlap measures before registration was included in the first column to illustrate how much improvement there is after registration. Overall, it can be seen that NiftyReg and Diffeomorphic Demons are the best performing algorithms.

4.2 Shape Collapse Experiments

In order to study the effect of shape collapse in greedy registration algorithms, time-varying velocity field registration algorithm provided by ANTs [6, 5], which is based on the LDDMM algorithm [8]. The choice of this algorithm is based on the popularity of this group of diffeomorphic registration algorithms today, and how well they performed in evaluation works such as the EMPIRE10 challenge [51, 52] and Klein's evaluation of brain MRI registration algorithms [41, 42]. While these diffeomorphic algorithms boast of good registration performance, it will be shown in this work that these algorithms also suffer from the shape collapse problem due to their greedy nature. Unless specified otherwise, all of the registrations in this chapter were generated using the ANTs registration suite using the time-varying velocity field method with $numberOfTimeIndices$ set to 4, $updateFieldVarianceInVoxelSpace$

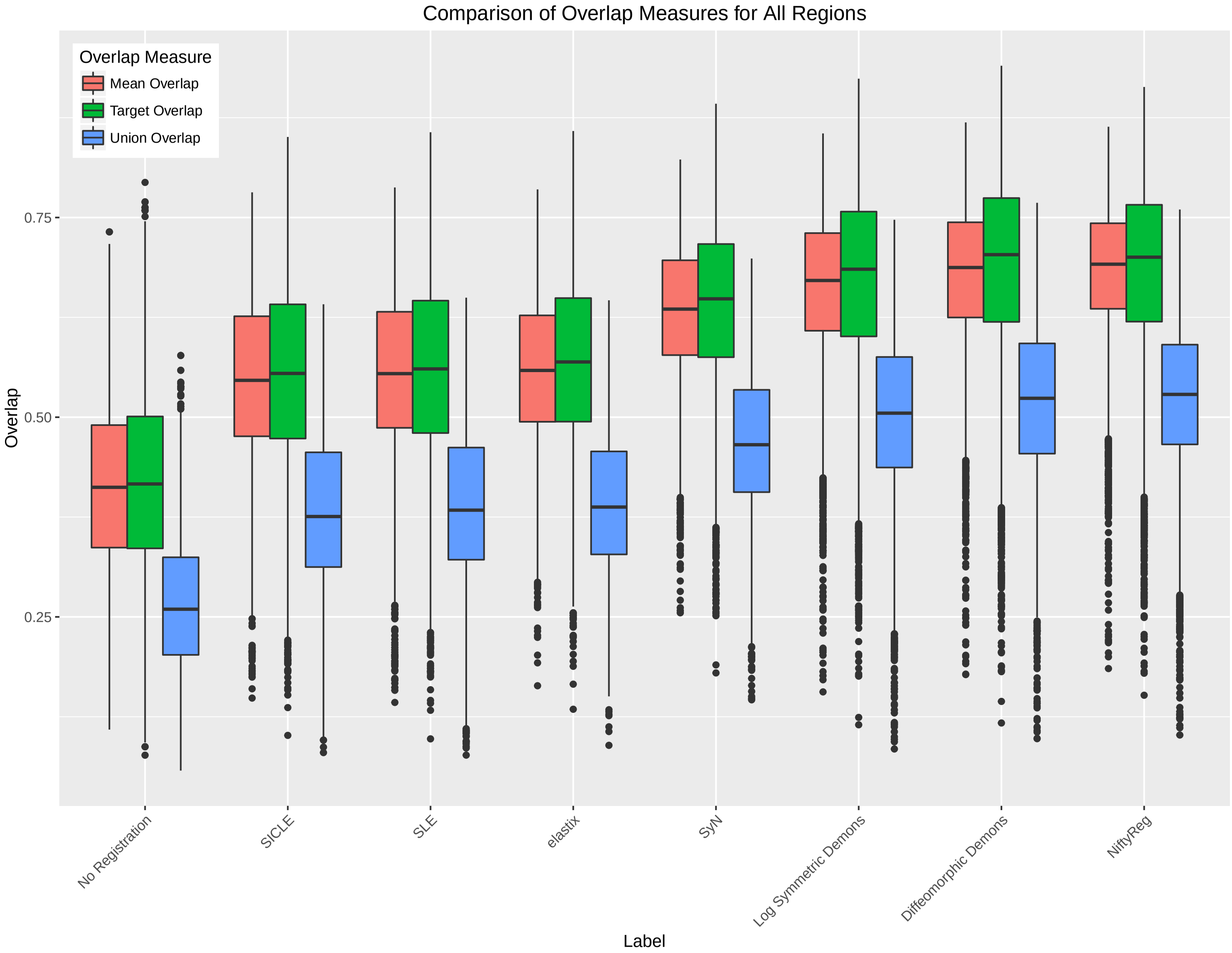


Figure 4.3. A boxplot of Target Overlap (red), Mean Overlap (green), and Union Overlap (blue) of all regions for all 240 pairwise registrations for each registration algorithm. The first column shows the overlap measures before registration.

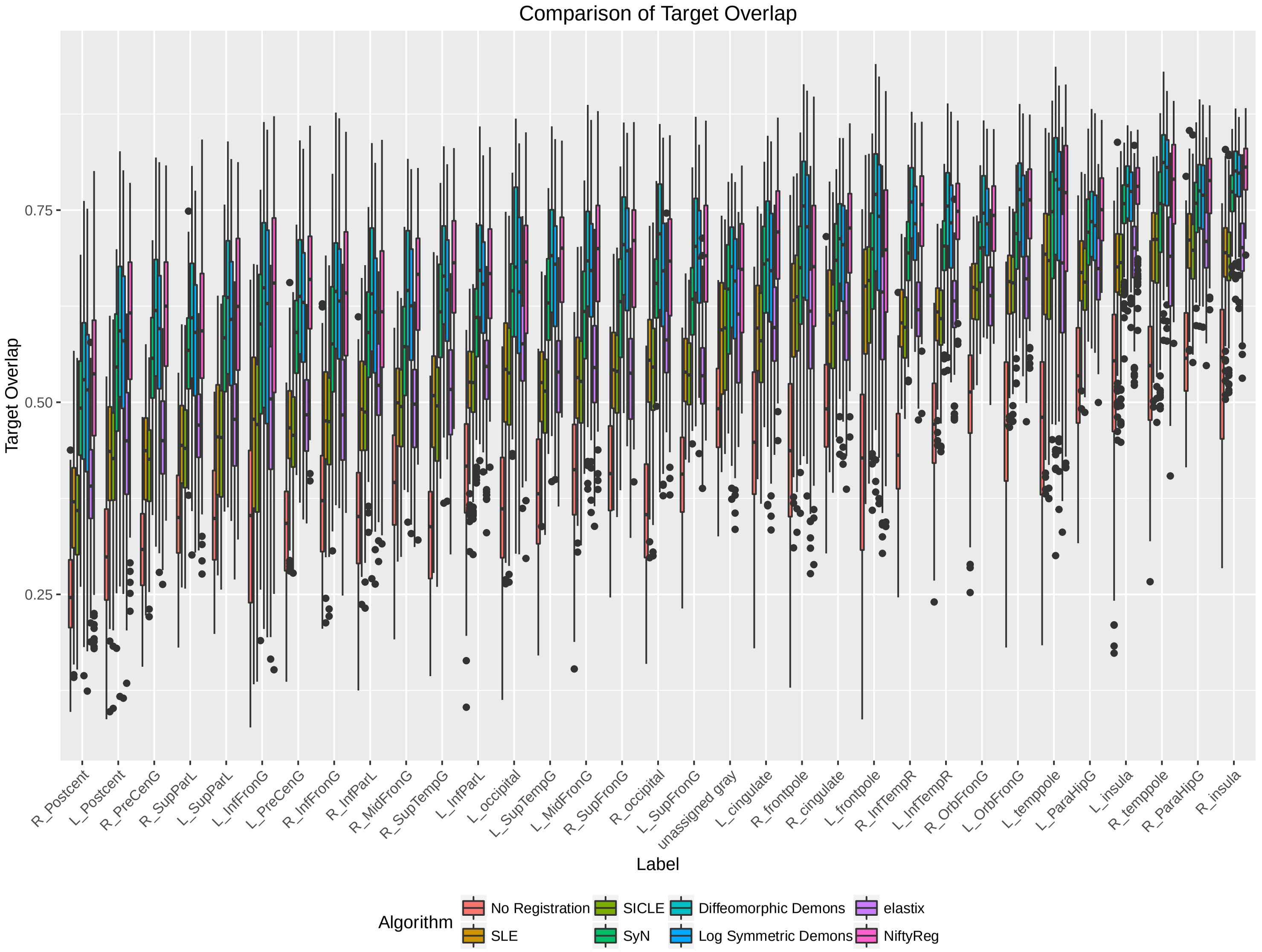


Figure 4.4. NA0 Target Overlap comparison

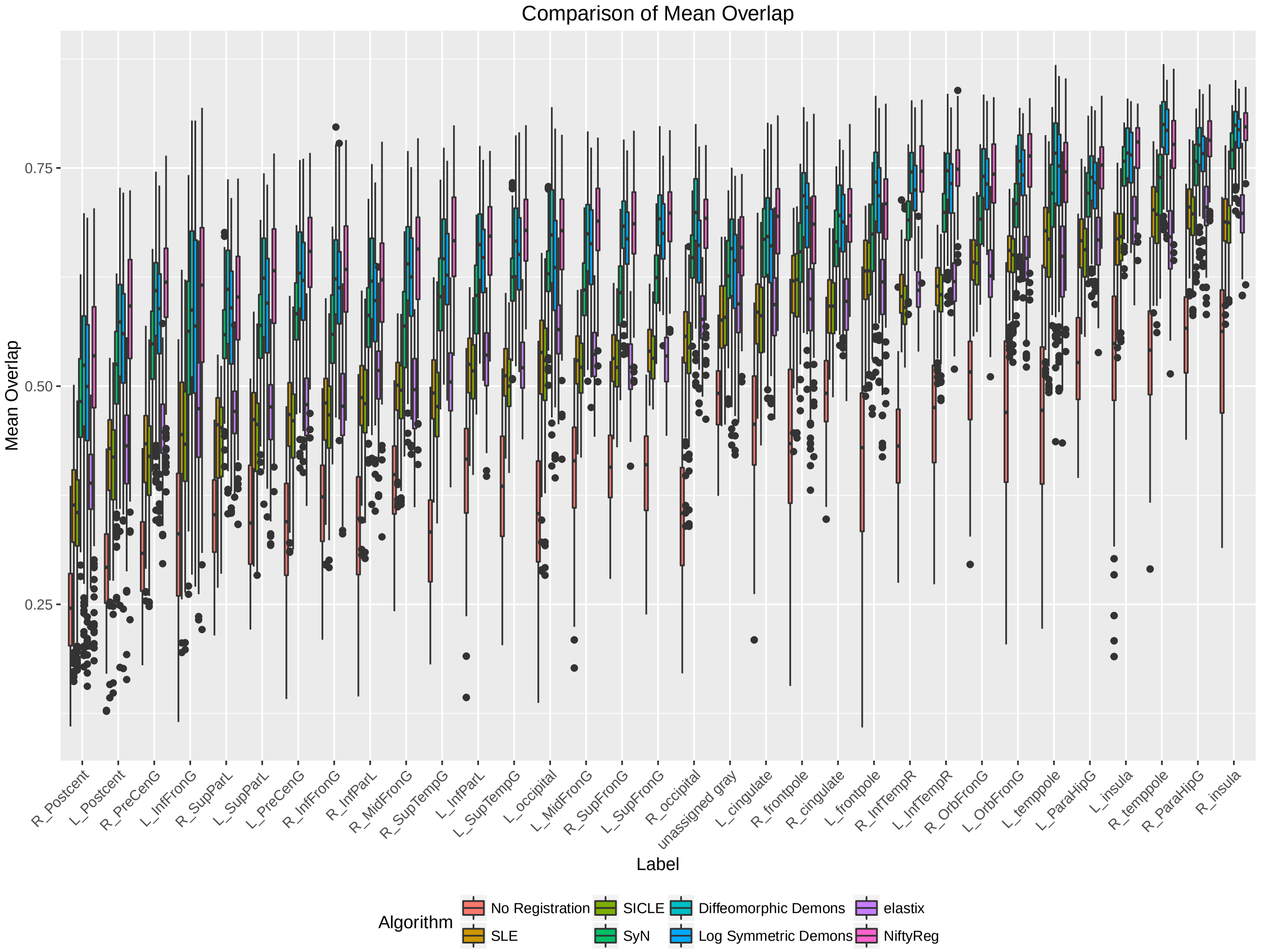


Figure 4.5. NA0 Mean Overlap comparison

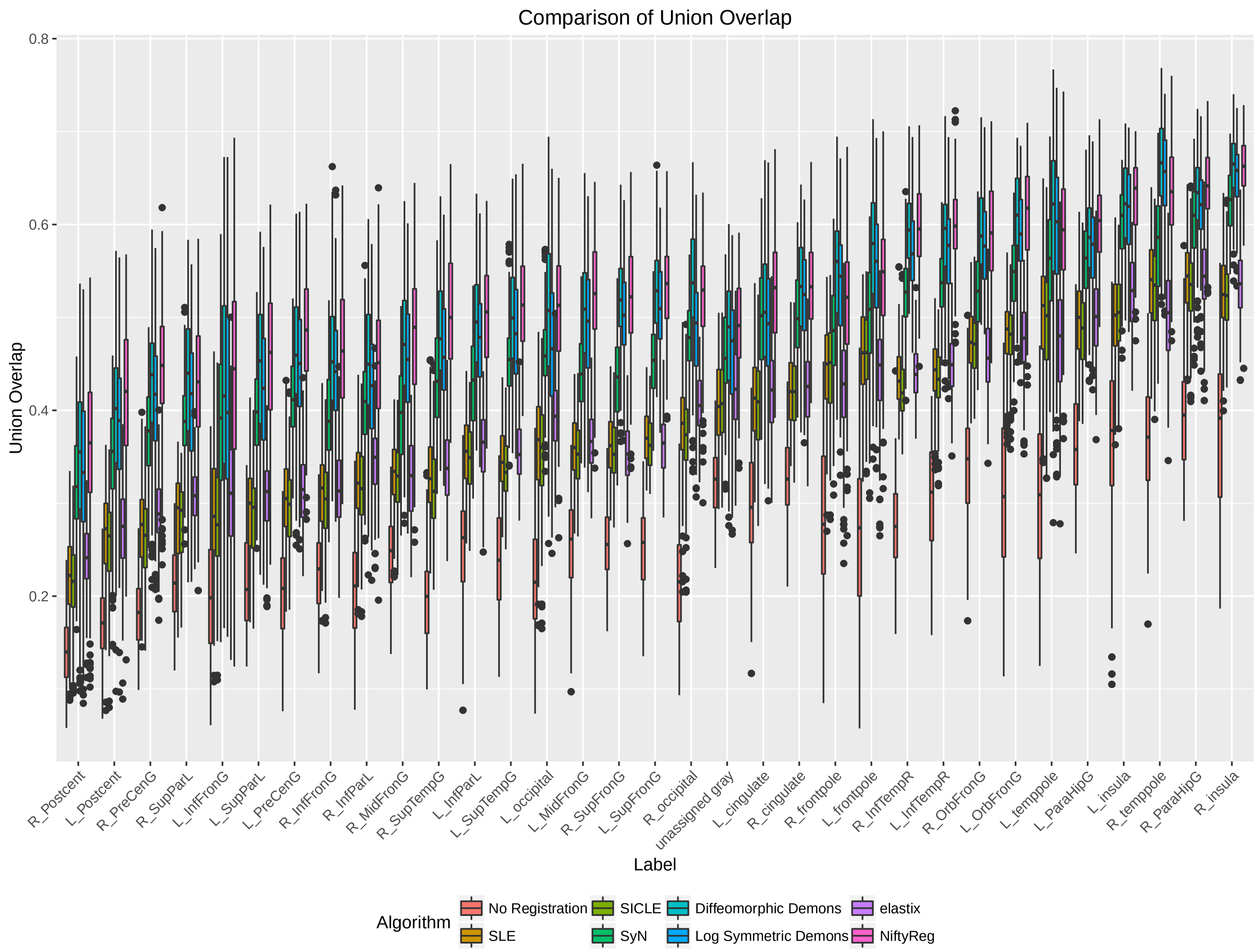


Figure 4.6. NA0 Union Overlap comparison

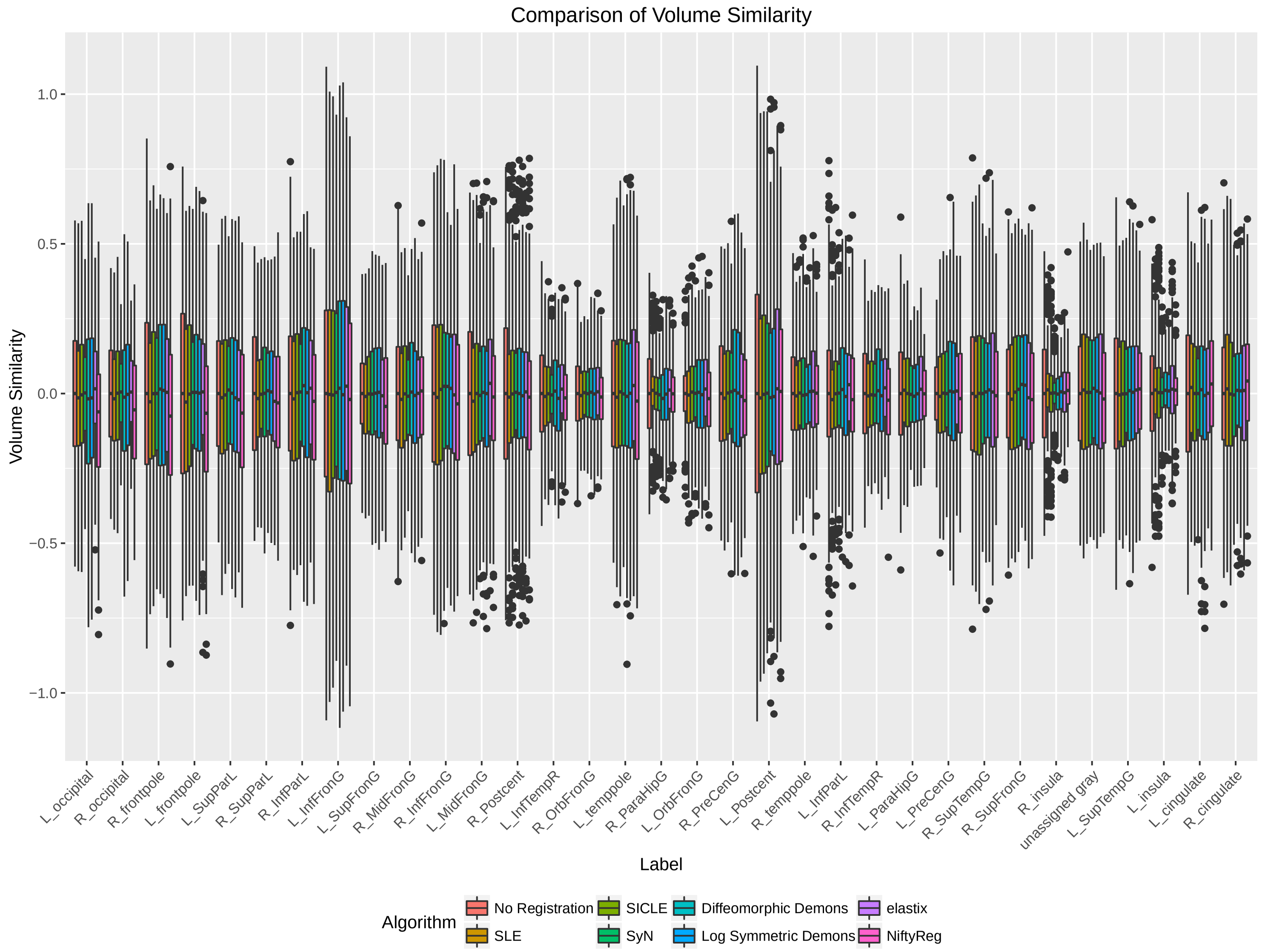


Figure 4.7. NA0 Volume Similarity comparison

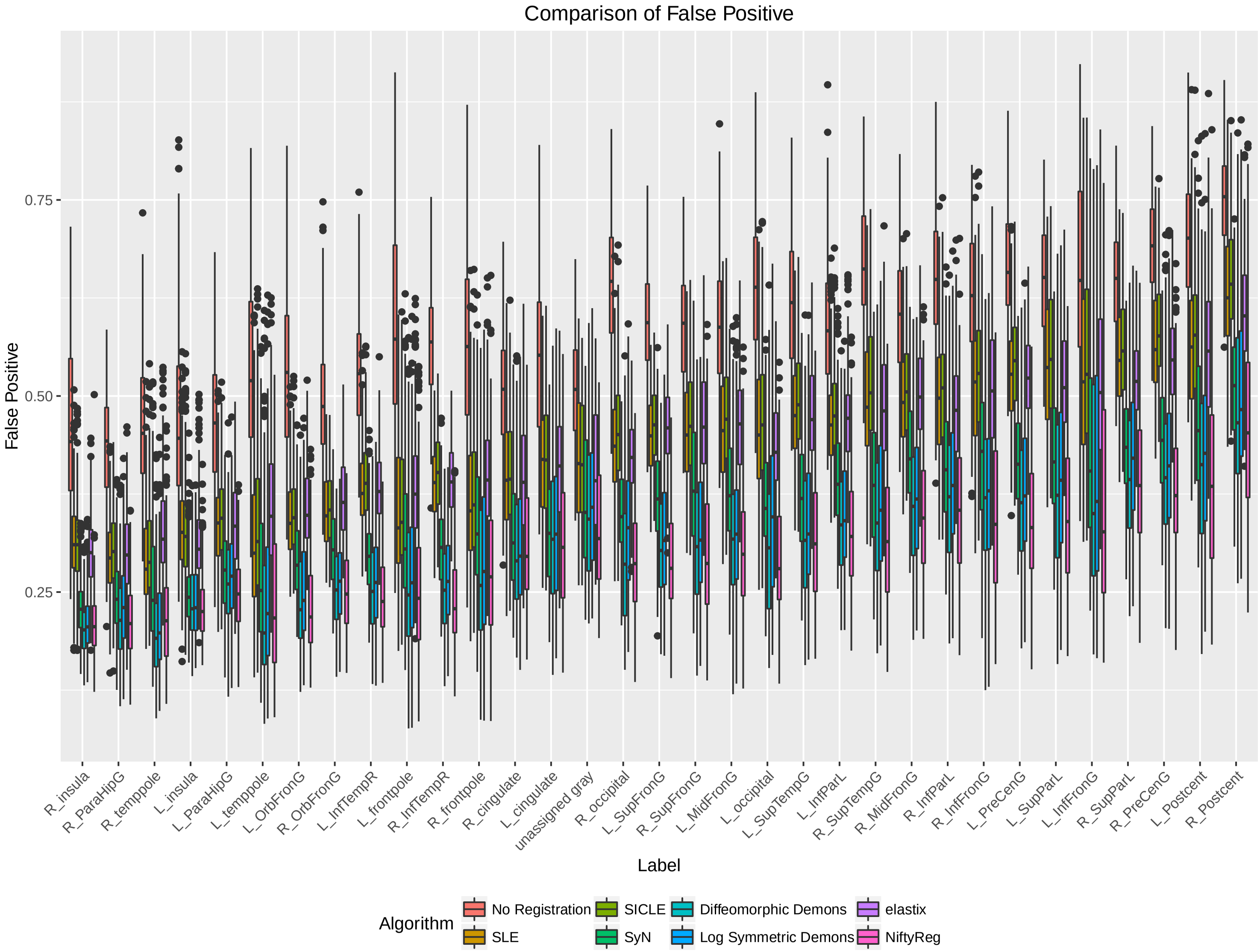


Figure 4.8. NA0 False Positive comparison

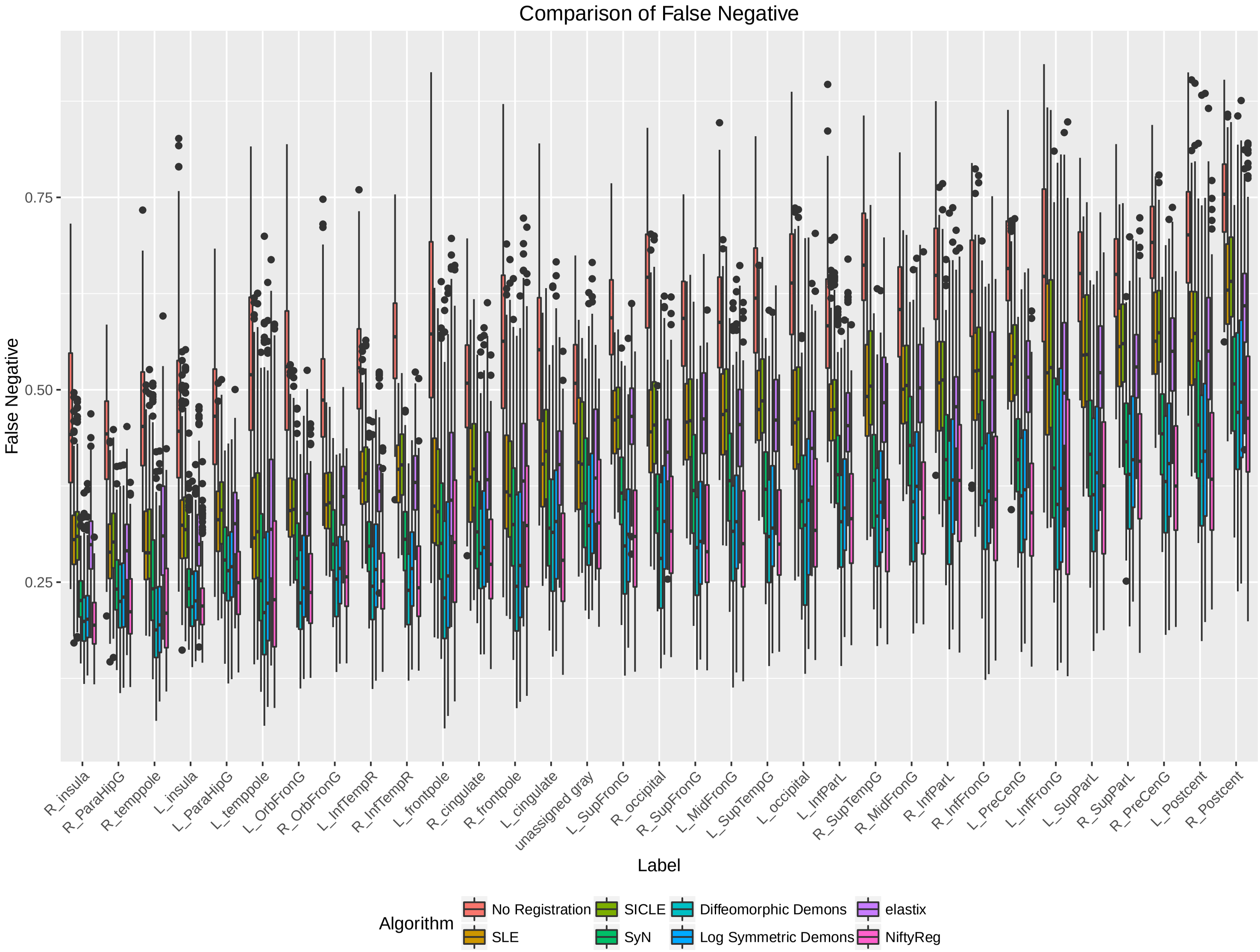


Figure 4.9. NA0 False Negative comparison

set to 0 and the remaining variance terms set to 0. Tools for predicting and detecting shape collapse points were implemented in C++ and MATLAB.

4.2.1 3D Binary Shape Collapse: 3D Objects Deformed to Their Skeletons

The shape collapse of 2D images to the empty set has been explored in the previous sections and has shown that the greedy registration algorithm's cost gradient flows along the fastest area-decreasing path (which is the path perpendicular to the level sets of the distance transform), ultimately collapsing to the skeleton of the shape. Similarly, we repeat the same experiment with simple 3D shapes to see if the same shape collapse behavior can be observed. Figure 4.10 illustrates a simple 3D example of a cylinder registering to the empty set. Detailed view of the displacement fields generated is shown in Fig. 4.11. Using the same k-means clustering method to measure the level of shape collapse of the displacement in 3D space, and generating a volume using the distance of the means for each voxel location, we obtain Fig. 4.10(d). Notice how the detected shape collapse region resembles the skeleton of the cylinder. Taking a detailed look at the Maurer distance transform of the cylinder (Figs. 4.11(a)-(c)) and comparing it to the displacement field generated by the registration algorithm (Figs. 4.11(d)-(l)), we observe that the registration process caused the cylinder to deform along the path perpendicular to the level sets of the distance transform, ultimately terminating at the skeleton of the cylinder. It is interesting to note that the displacements to the top and bottom of the "ears" of the skeleton (i.e., the cone-shaped skeleton) are purely in the Z-axis direction.

Likewise, the same experiment was carried out using a more complex shape

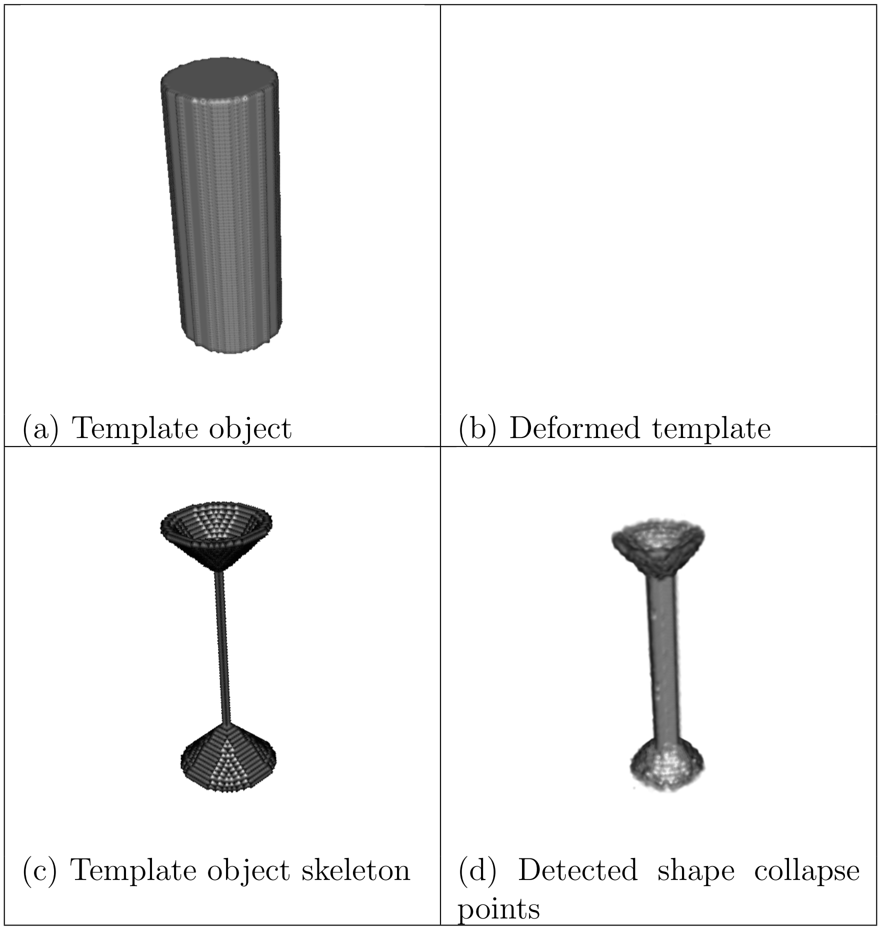
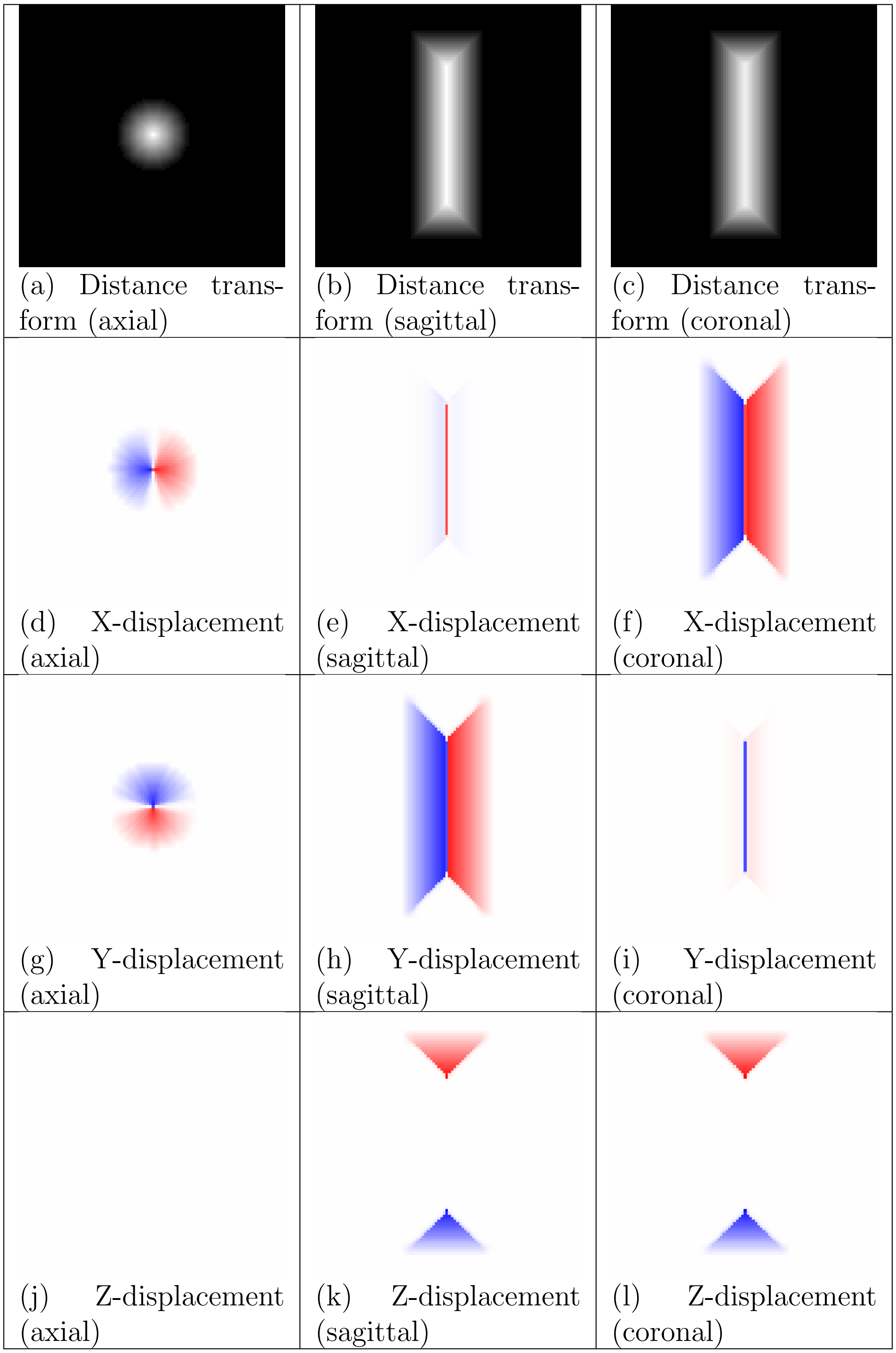

(a) Template object
(b) Deformed template
(c) Template object skeleton
(d) Detected shape collapse points

Figure 4.10. Shape collapse resulting from registering a cylinder to the empty set. The detected shape collapse points using k-means clustering reveals the cylinder collapsed into its skeleton.



(a) Distance transform (axial)

(b) Distance transform (sagittal)

(c) Distance transform (coronal)

(d) X-displacement (axial)

(e) X-displacement (sagittal)

(f) X-displacement (coronal)

(g) Y-displacement (axial)

(h) Y-displacement (sagittal)

(i) Y-displacement (coronal)

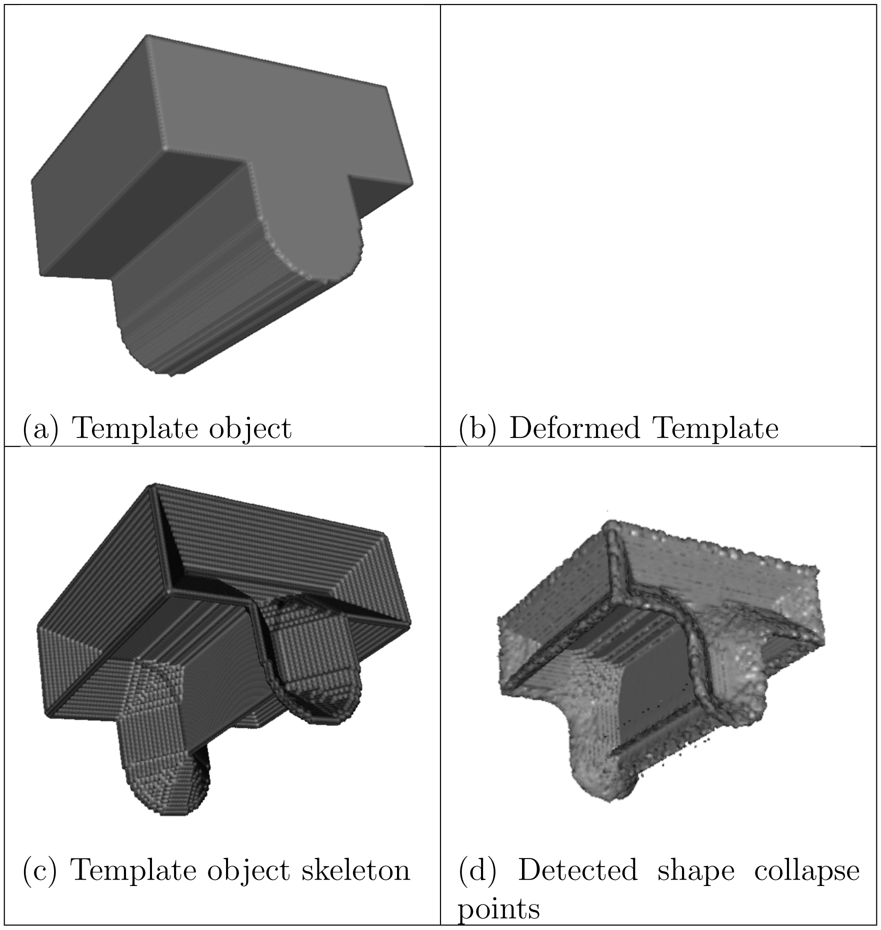
(j) Z-displacement (axial)

(k) Z-displacement (sagittal)

(l) Z-displacement (coronal)

Figure 4.11. Orthogonal views of the Maurer distance transform of the cylinder and displacement fields. The displacement fields reveal that the cylinder followed the path perpendicular to the level sets of the distance transform and collapsed into the skeleton.

like the 3D “tooth” shown in Fig. 4.12.



(a) Template object

(b) Deformed Template

(c) Template object skeleton

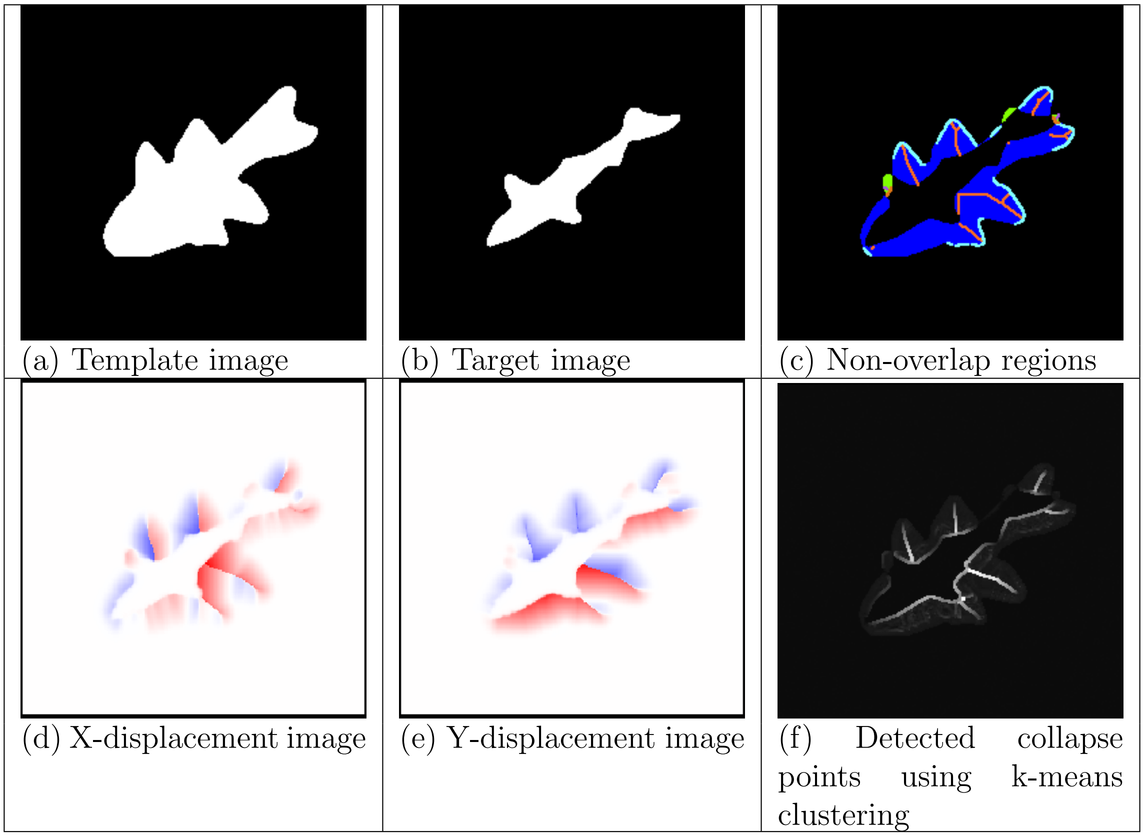
(d) Detected shape collapse points

Figure 4.12. Shape collapse of a 3D tooth registering to the empty set. The detected shape collapse points using k-means clustering reveals the shape collapsing into its skeleton.

4.2.2 One-sided Binary Shape Collapse: 2D Fish Example

Next, experiments were carried out with image pairs were the target image is non-zero. Figure 4.13 illustrates an example of a big fish being registered to a small fish. Fig. 4.13(c) shows the non-overlap regions where blue indicates non-overlap region U_1, green indicates non-overlap region U_2, red indicates foreground skeleton in U_1, purple indicates background skeleton in U_2, cyan indicates collapsing foreground boundary, and orange indicates collapsing background boundary. Figs. 4.13(d) and (e) show the X- and Y-displacement fields generated by the registration algorithm, and Fig. 4.13(f) shows the detected shape collapse points using k-means clustering. Notice that apart from the predicted shape collapse about the foreground and background skeletons in the non-overlap region, we also see shape collapse happening around the boundary of the target image. As mentioned in the previous section, this is due to the fact that this particular registration algorithm does not smooth out the transformation (i.e., the displacement field does not taper off gradually around the boundary, but terminates abruptly). A way to interpret this phenomenon is that a region in the template image that had non-zero area collapsed into a set of zero measure around the boundary of the target image, while there was zero deformation in the overlapping region W of the image pair. In other words, instead of smoothly rescaling to the target image, the non-overlap region "crashes into" the target boundary as it deforms.

To aid understanding how the registration algorithm deforms the template image during registration, a segmentation of the fins of the fish was made as shown in



(a) Template image

(b) Target image

(c) Non-overlap regions

(d) X-displacement image

(e) Y-displacement image

(f) Detected collapse points using k-means clustering

Figure 4.13. Shape collapse example of big fish being registered to a small fish. (c) Blue: U_1; Green: U_2; Red: Foreground skeleton in U_1; Purple: Background skeleton in U_2; Cyan: Collapsing foreground boundary; Orange: Collapsing background boundary.

Fig. 4.14(a). Then the progressive change of the template image shape at each iteration of the registration process is shown in Fig. 4.14. As it can be observed from the mid-stage deformed images, there is no deformation happening in the overlapping region of the template and target images, as the segmentation in this region is preserved throughout the registration process. As it was suspected that the absence of regularization allowed the deformation to have abrupt discontinuities, the registration was repeated with various regularization values (i.e., $updateFieldVarianceInVoxelSpace$).

4.2.3 Mitigating Binary Shape Collapse: 2D SyN Registration

Figure 4.15 shows the registration results for different regularization factors. As it can be seen from the displacement and k-means images, higher regularization factors do indeed smooth out the deformation field, reducing the effect of shape collapse both at the target boundary and the non-overlap region. This is an expected behavior, since smoothing out the deformation with regularization slows down the rate of area-decreasing flow. However, it can also be seen that higher regularization factors limit the degree of freedom the deformation can have and results in registration errors as shown by the squared difference images.

The full ramification of the shape collapse problem can be observed by the alignment of the segmentations (as shown in Fig. 4.16) using the transformation generated by the registration algorithm registering the two binary images. Although the similarity cost of the registered binary images may have been minimized by the registration algorithm, the pointwise correspondence within the iso-intensity region may not be the desired result. Figs. 4.16(a) and (b) illustrate how the fins of the big and

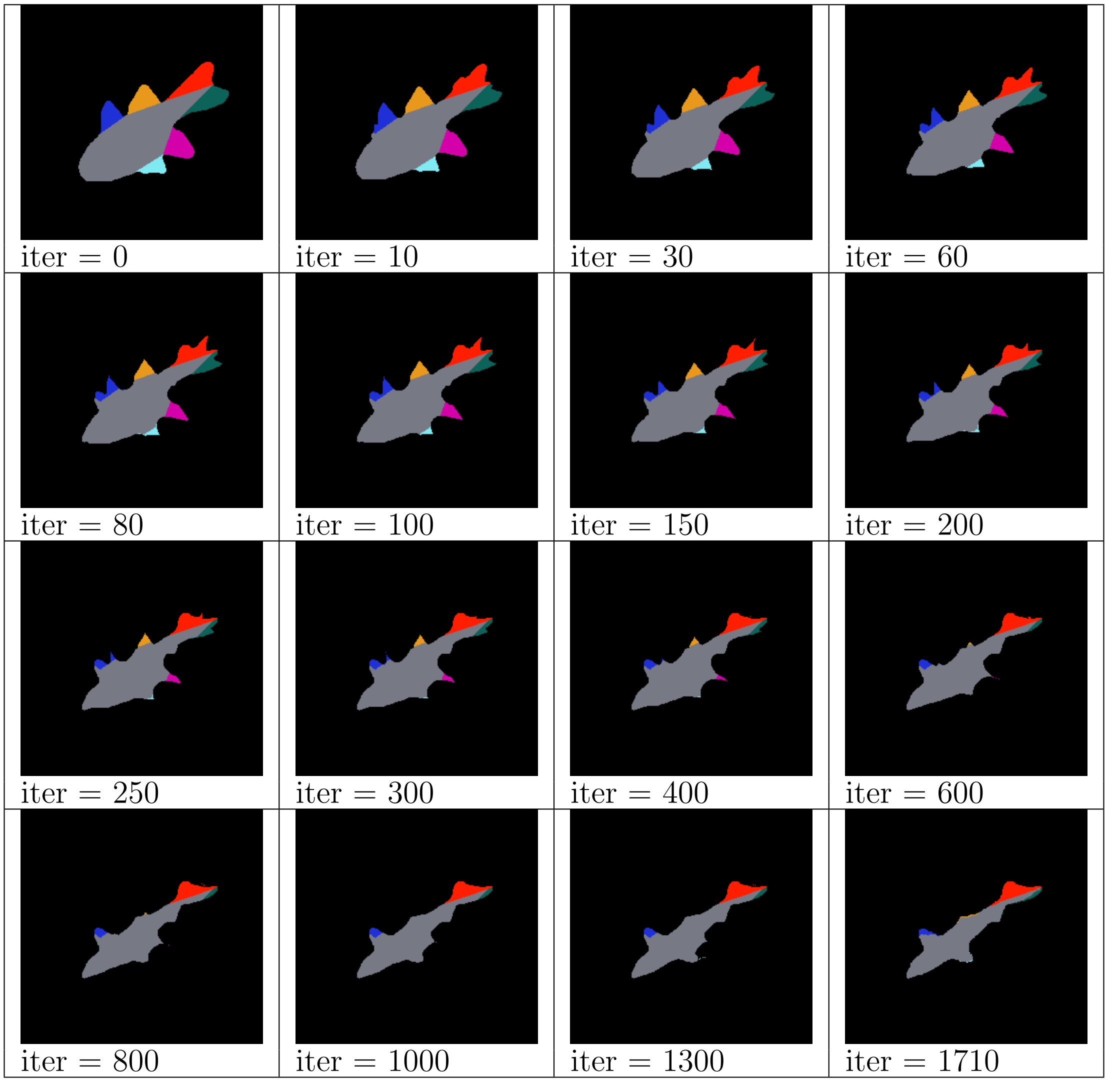


Figure 4.14. Deformed template image mask at each iteration during registration.

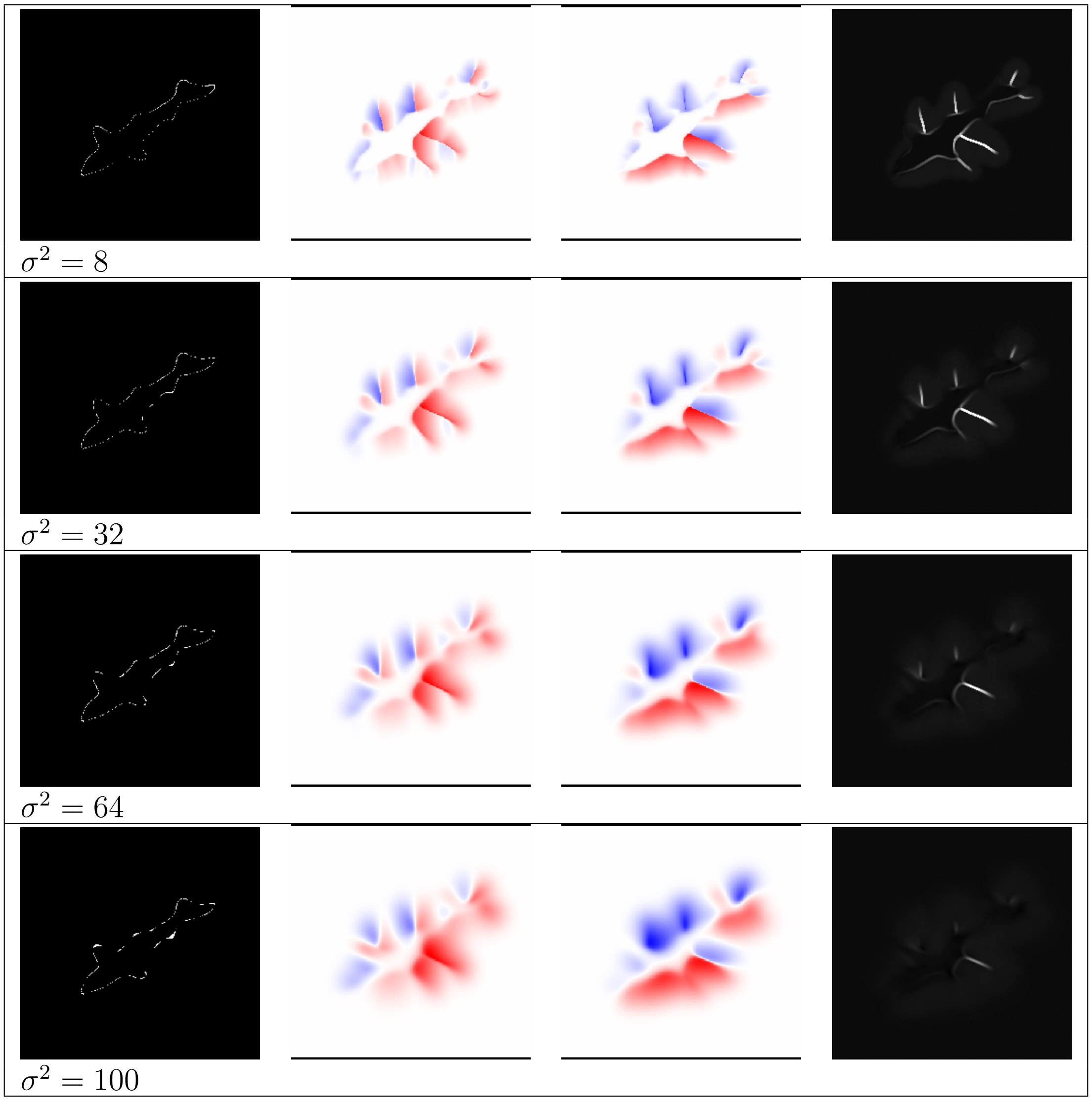
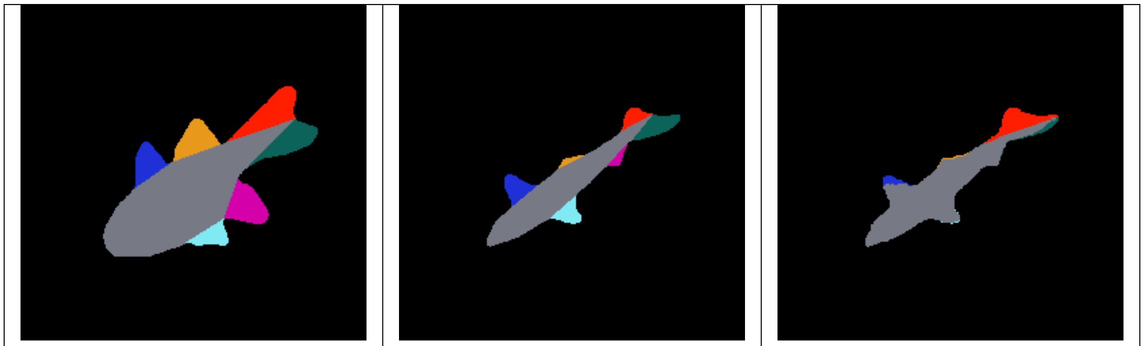


Figure 4.15. Squared difference after registration (first column), X- & Y-displacements (second and third column), and detected collapse regions (last column) for different regularization values (σ^2).

small fish can be segmented, respectively. The desired registration result is to deform image (a) to (b), aligning the fins and torso. However, the deformed template image mask shown in (c) shows that instead of matching the fins, the registration algorithm instead collapsed the non-overlap regions to match the target shape. Consequently, we see that the purple fin present in (a) is missing in (c) due to complete collapse, and only small remnants of the cyan, orange, green and blue fins remain in the deformed image. What this tells us is that had the images been from real anatomy instead of fish phantoms, we may encounter situations where a structure with non-zero volume may map to a zero-measure strip of skeleton, which is an undesirable result.



(a) Template mask

(b) Target mask

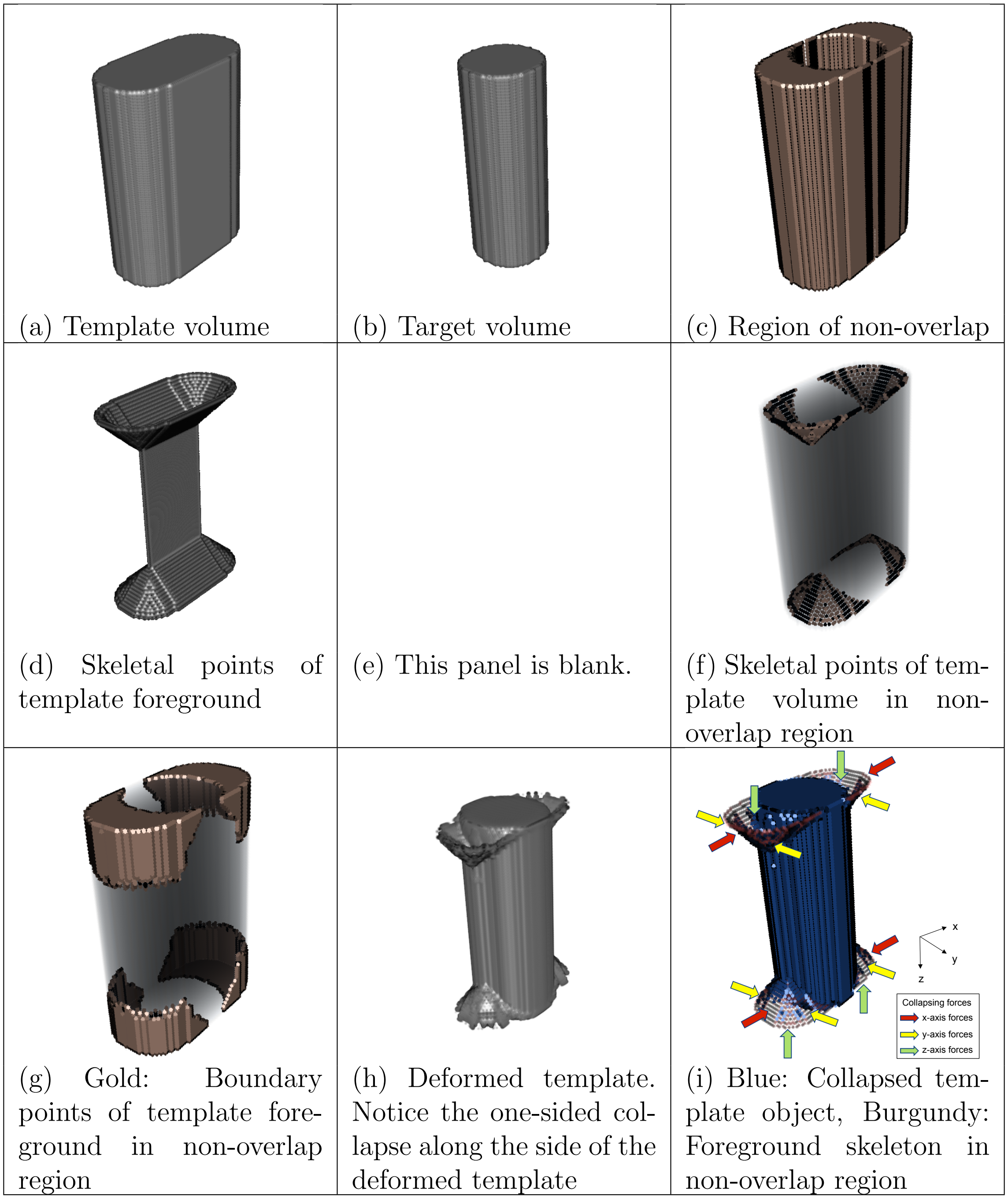
(c) Deformed template mask

Figure 4.16. Segmentation of the fins of the fish shows that despite zero sum of squared difference error after registration, the internal correspondence may not be the desired result.

4.2.4 3D Binary Shape Collapse: Flat Cylinder Registered to a Round Cylinder

The next example showing the shape collapse of a flat-cylinder (i.e., a cylinder with its two sides flattened) registering to a cylinder may illustrate a possible scenario in which a flattened vessel registers to a cylindrical vessel by shape collapse. Figures 4.17(a) and (b) are 3D renderings of the flat-cylinder and cylinder to be registered. Both shapes have the same height and width on the short axis. The 3D rendering in Fig. 4.17(c) shows the region of non-overlap, which looks like the flat-cylinder with a hole the size of the target cylinder drilled through the center. Because the region of non-overlap only consists of U_1 and $U_2 = \varnothing$, we can predict that only shape collapse in the foreground will occur, as shown in Figs. 4.17(d)-(g). The gold region in Fig. 4.17(g) shows the boundary points in the template volume that is likely to collapse, and the gold region in Fig. 4.17(f) shows the skeletal points about which shape collapse is expected to occur. In this example, the non-overlap region contains both collapsing and non-collapsing boundary points. The boundary points associated with the ear-shaped skeletal points (as shown in Fig. 4.17(f)) in the non-overlap region are likely to collapse during registration whereas boundary points associated with the skeletal points in the overlap region are likely to deform towards the boundary of the cylinder. The deformed template volume in the mid-stages of the registration process shown in Fig. 4.17(h) confirms the prediction with the flat-cylinder collapsing around the “ears” of the skeleton. The arrows in Fig. 4.17(i) show the direction of the gradient flow during registration.

Using the k-means clustering method, the actual shape collapse points result-



(a) Template volume

(b) Target volume

(c) Region of non-overlap

(d) Skeletal points of template foreground

(e) This panel is blank.

(f) Skeletal points of template volume in non-overlap region

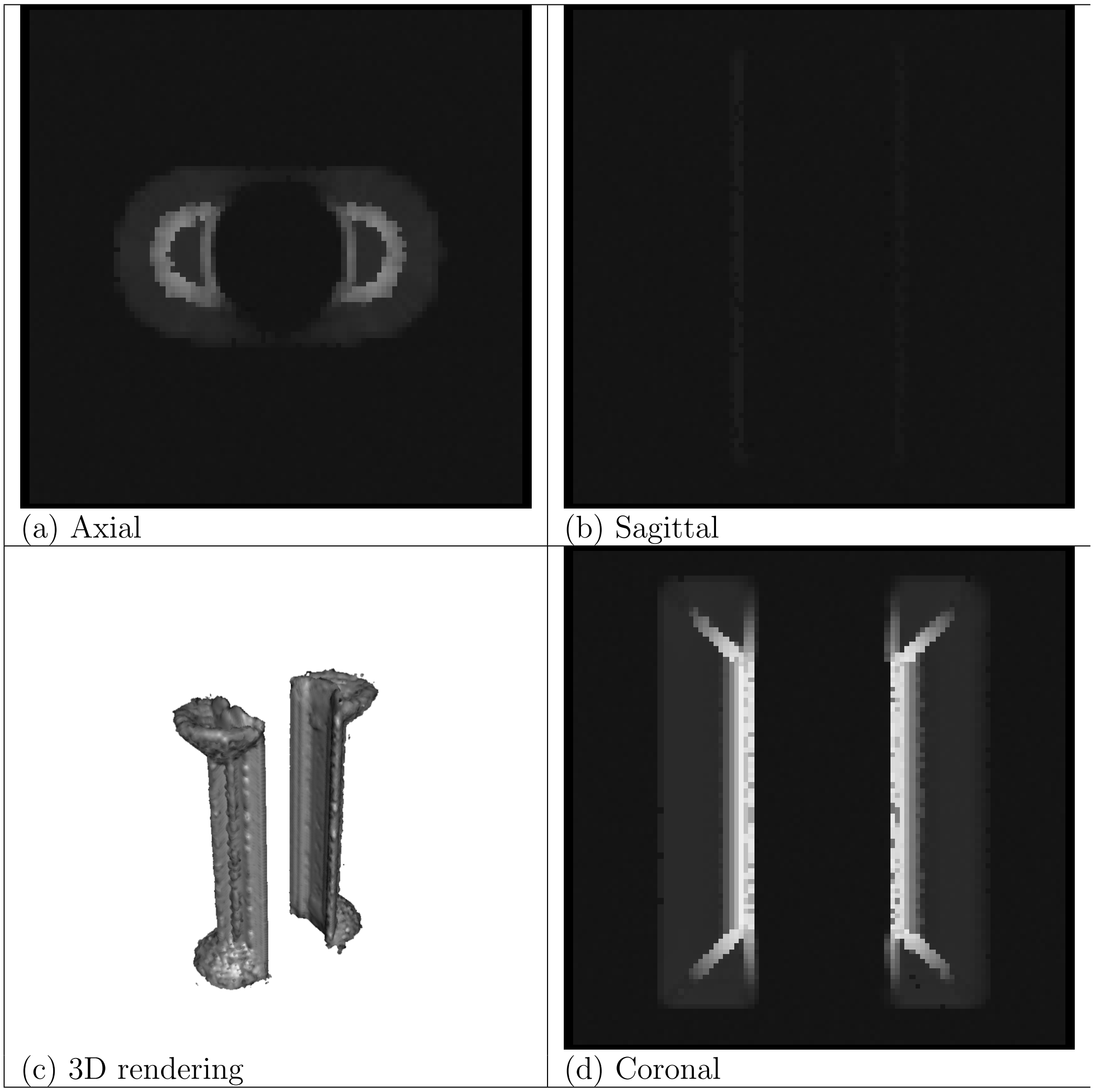
(g) Gold: Boundary points of template foreground in non-overlap region

(h) Deformed template. Notice the one-sided collapse along the side of the deformed template

(i) Blue: Collapsed template object, Burgundy: Foreground skeleton in non-overlap region

Figure 4.17. Shape collapse example of a flat-cylinder registering to a round cylinder of same height and short-axis width. Because the region of non-overlap contains only U_1 and $U_2 = \varnothing$, only foreground collapse occurs. The deformed flat cylinder in mid-stage registration shown in (h) shows two-sided shape collapse around the "ears" of the skeleton and the one-sided collapse along the sides.

ing from the registration algorithm are shown in Fig. 4.18. Similarly to the big fish to small fish registration example, not only do we observe the two-sided shape collapse in the predicted region, but we also see one-sided shape collapse around the boundary of the target shape.


(a) Axial

(b) Sagittal

(c) 3D rendering

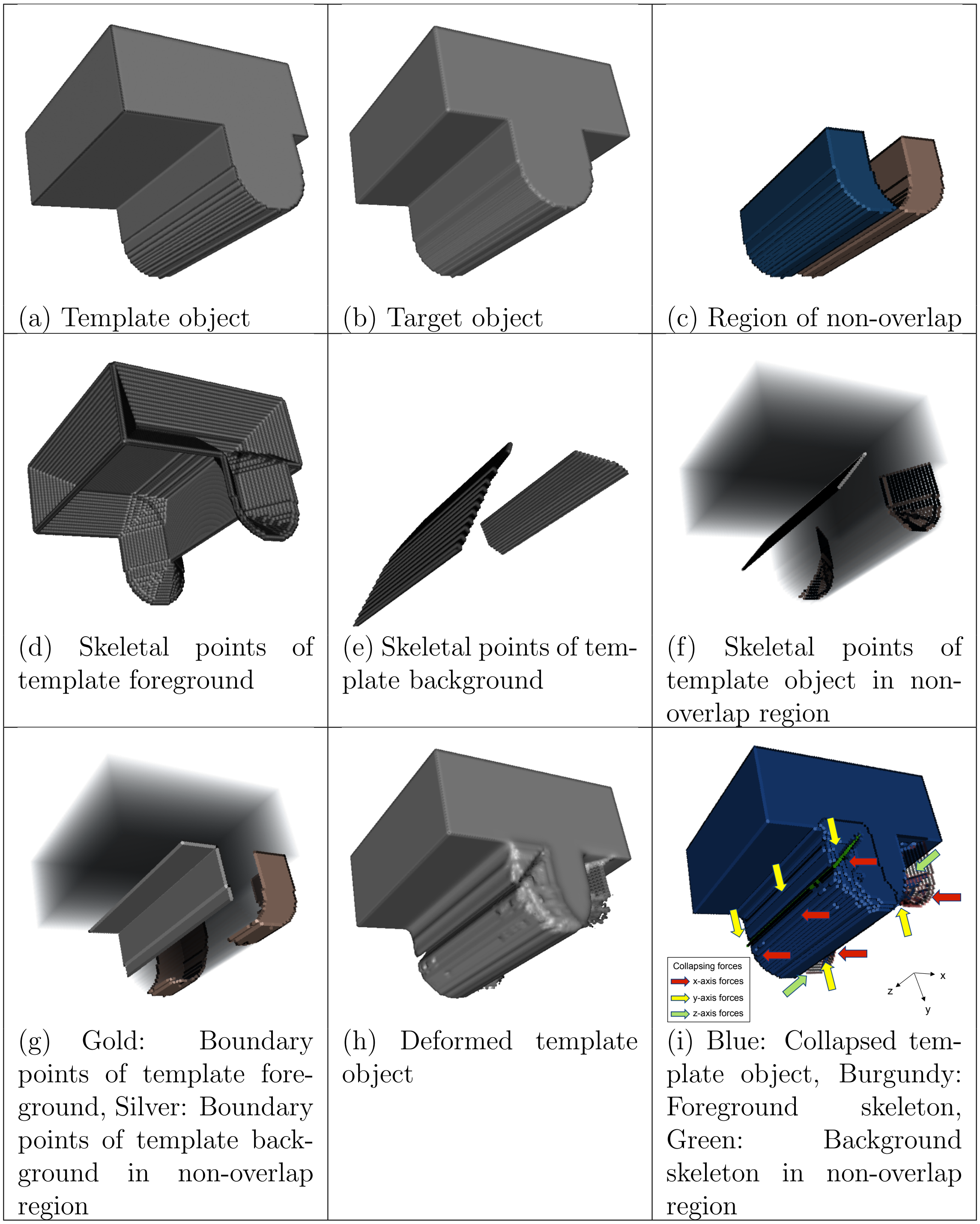
(d) Coronal

Figure 4.18. Orthogonal views and 3D rendering of the detected shape collapse regions of the flat-cylinder registering to a cylinder using k-means clustering. Notice the one-sided shape collapse along the wall of the cylinder.

4.2.5 3D Foreground and Background Binary Shape Collapse: Sliding Appendage

The next example demonstrates both foreground and background shape collapse in 3D space. Extending the 2D tooth example in Fig. 3.20 into 3D, we have the template and target objects as shown in Figs. 4.19(a) and (b). The non-overlap region shown in Fig. 4.19(c) contains both U_1 and U_2, which means that both foreground and background collapse is likely to occur. The skeletal points illustrated in Fig. 4.19(f) show where foreground and background shape collapse is going to take place. Notice that the boundaries highlighted in Fig. 4.19(g) collapses into skeletal points that are 2D planes, which has co-dimension one with the 3D volume. Fig. 4.19(i) illustrates the gradient flow direction taken to collapse the foreground and the background.

Figure 4.20 shows the orthogonal views and the 3D rendering of the shape collapse image generated using the k-means clustering method. We can confirm that the registration algorithm generated displacements that are highly dispersed around the skeletal region of the collapse region, as predicted.



(a) Template object

(b) Target object

(c) Region of non-overlap

(d) Skeletal points of template foreground

(e) Skeletal points of template background

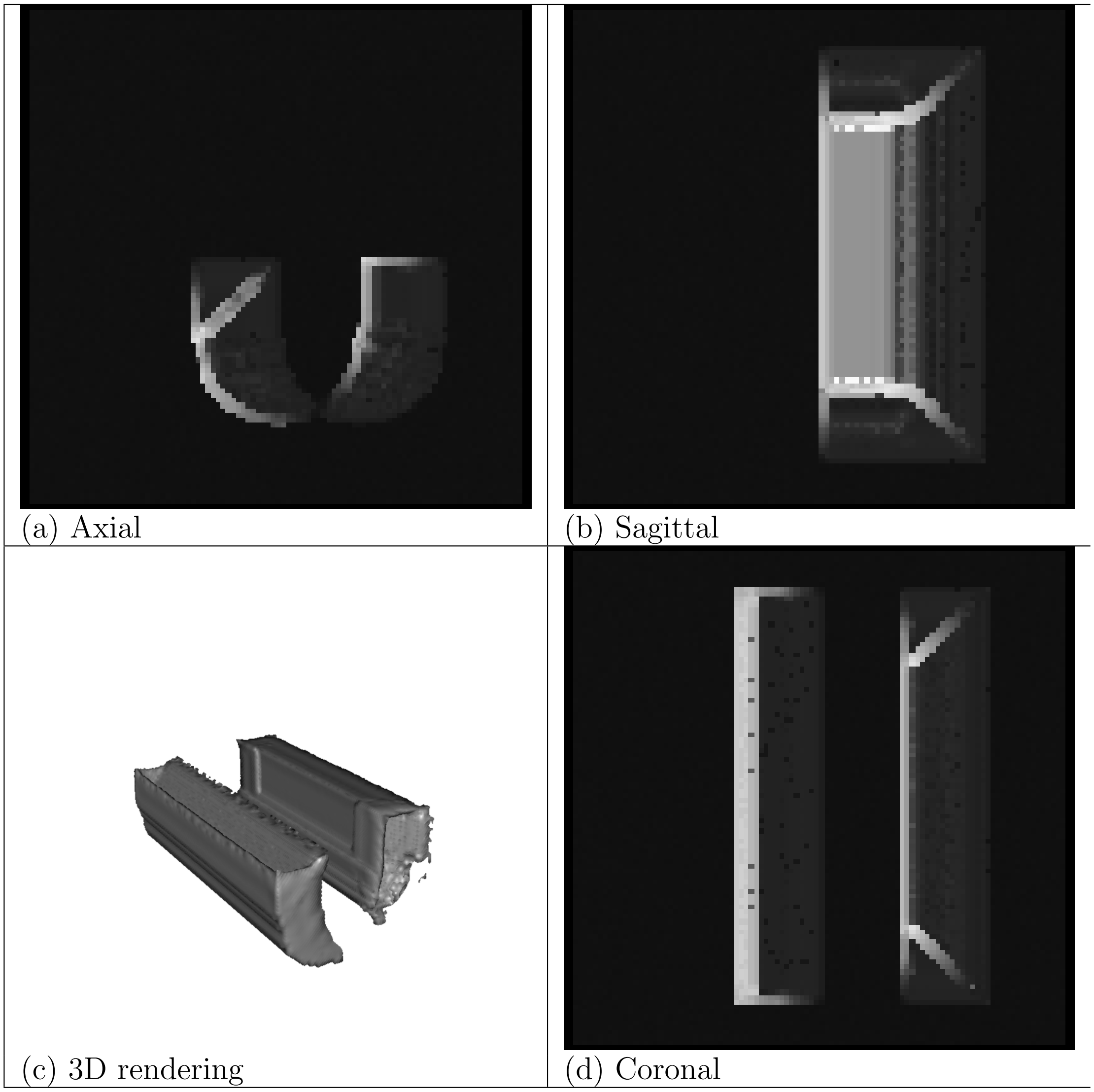
(f) Skeletal points of template object in non-overlap region

(g) Gold: Boundary points of template foreground, Silver: Boundary points of template background in non-overlap region

(h) Deformed template object

(i) Blue: Collapsed template object, Burgundy: Foreground skeleton, Green: Background skeleton in non-overlap region

Figure 4.19. 3D foreground and background collapse using time-varying velocity field transformation with no regularization. Notice that the foreground collapses into the skeleton and the background encroaches into the foreground along the skeleton.



(a) Axial

(b) Sagittal

(c) 3D rendering

(d) Coronal

Figure 4.20. Orthogonal views and 3D rendering of the detected shape collapse regions of the teeth registration using k-means clustering.

4.2.6 Mitigating Binary Shape Collapse: 3D SyN Registration

Figure 4.21 shows how different levels of regularization affects the level of shape collapse in 3D. In each column, the template objects (flat-cylinder and tooth) were deformed using transformations generated by registration with various smoothing variances. The gradual diminishing of shape collapse levels can be observed visually in each column as regularization levels are increased. A quantitative analysis of the level of shape collapse at different regularization levels can be made by taking the mean summation of Jacobian of transformation at points in the deformed image coordinate system that are mapped to collapsing boundary points in the template image coordinate system. Figure 4.22 shows the plot of sum of Jacobian at collapsing boundary points versus regularization levels.

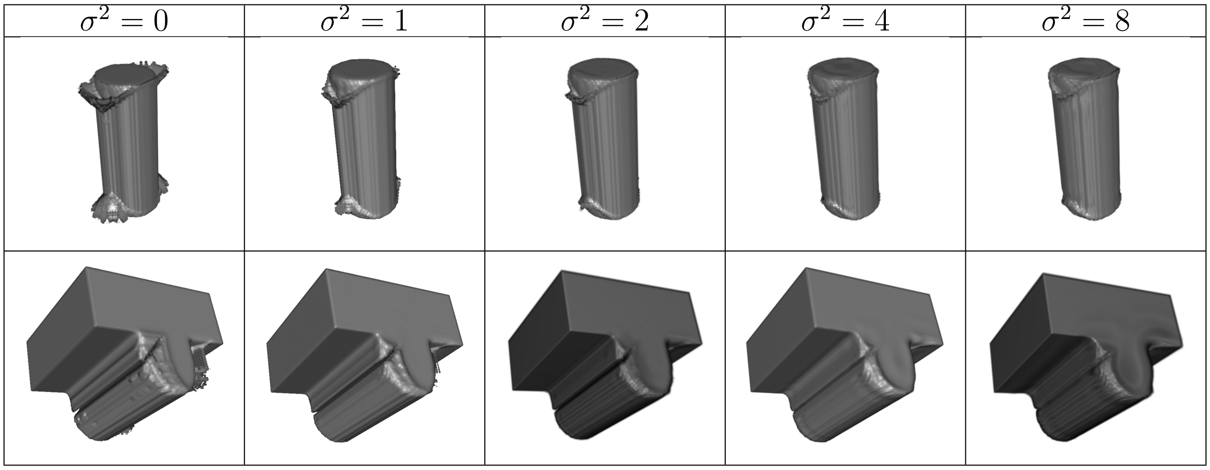
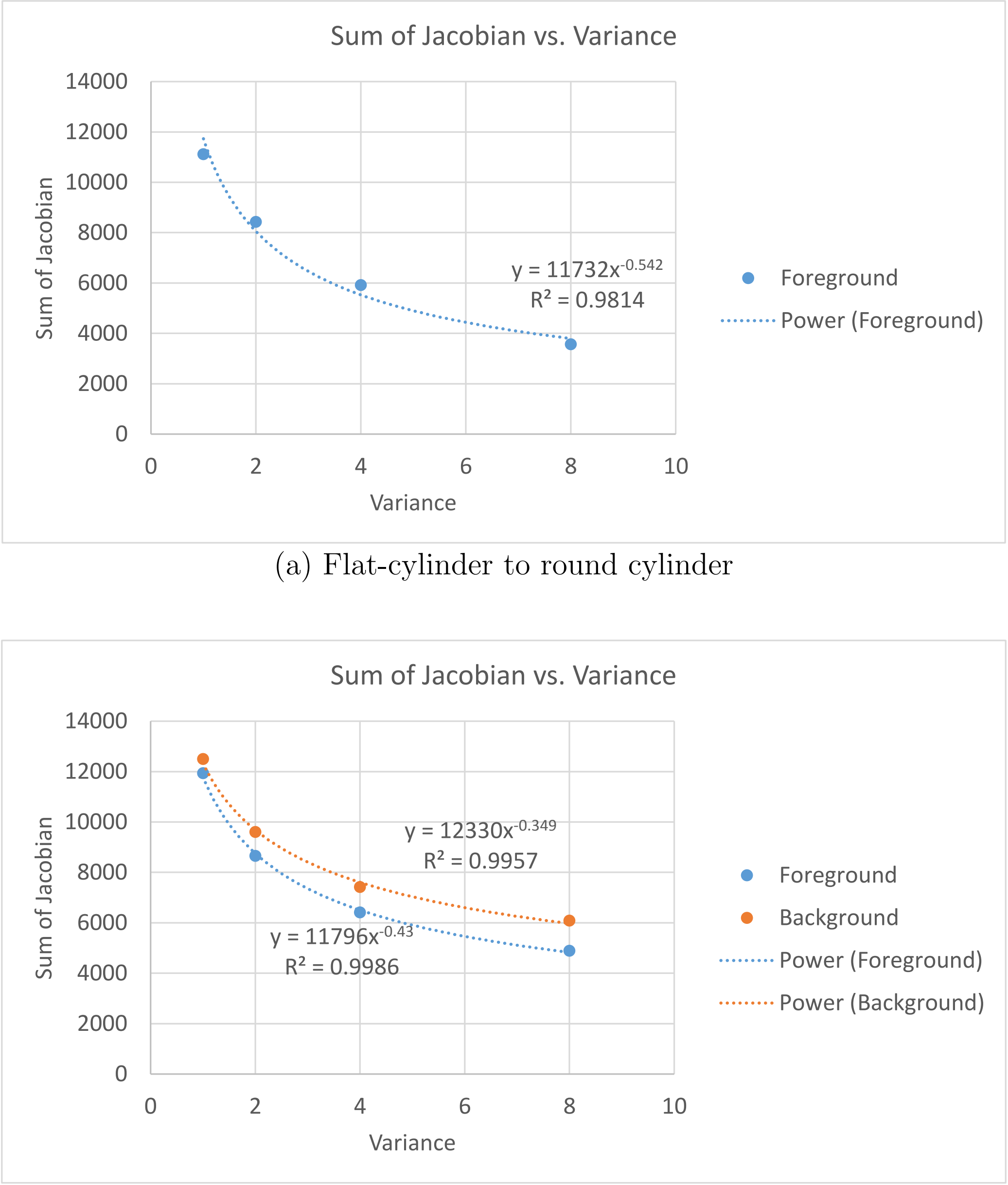


Figure 4.21. Demonstration of how different regularization parameters affect the level of collapse.



(a) Flat-cylinder to round cylinder

(b) Sliding appendage

Figure 4.22. Plot of sum of Jacobian values at collapsing foreground and background boundaries at different regularization levels.

4.2.7 3D Shape Collapse: MRI Brain Images

To demonstrate that the shape collapse problem is a real problem that can be found in real life situations, brain MR images from the NA1 database were registered using the ANTs registration package. Real brain data is far more complex than the simple shape examples that were used in earlier sections, added on to the fact that we're now dealing with grayscale image, not binary images. The added complexity made prediction of the collapse region very difficult, but the collapse detection mechanism has worked well. Time-varying velocity field transformation model of the ANTs package was used for all registrations.

NA1 consists of 18 brain MR volumes from 9 male and 9 female normal right-handed adults, also from the Human Neuroanatomy and Neuroimaging Laboratory. The demographics of the subjects are shown in Table 4.3. All 18 MR volumes were aligned along the AC-PC points prior to applying the non-rigid image registration algorithms. Segmentations of 57 grey matter ROIs acquired from the output of FreeSurfer using the Desikan-Killiany atlas and edited by Joel Bruss to correct mistakes will be used to evaluate registration performance.

The study population was a consecutive series of participants recruited from: (1) The University Hospital setting, recruited through local advertising; (2) Local communities in Iowa, recruited through newspaper advertising (and with special targeting of minority populations); (3) A dataset of approximately 80 normal control subjects who agree to be contacted for future studies under IRB# 2000303002 - Mechanisms of Perimetric Variability - Michael Wall, PI. and (4) College students at the

Table 4.3. Clinical demographic characteristics of the study population of NA1.

Label	Age	Gender	Race	Handedness
3065	35	Male	White	+100
3362	23	Male	White	N/A
3368	26	Male	White	N/A
3402	23	Male	Hispanic	+100
3403	32	Male	African Am.	+95
3407	28	Female	White	+100
3413	26	Female	White	+100
3414	31	Male	African Am.	+90
3424	24	Female	White	+100
3425	23	Male	White	+95
3463	24	Male	White	+90
3468	36	Female	White	+100
3474	30	Female	White	+100
3479	29	Female	White	+80
3481	39	Male	White	+95
3489	56	Female	White	+100
3491	27	Female	Asian/Pacific Islander	+100
3493	22	Female	African Am.	+95

University of Iowa, through local advertising; (5) Referral from colleague - Patients of Michael Wall, MD. Dept. of Neurology, who is a co-investigator on this project; (6) Other-Word of mouth. The inclusion/exclusion criteria for MRI included in this database are: (1) $Age > 18years$; (2) No history of developmental, neurological or psychiatric disease; (3) Normal visual acuity (can read text in a book at arms length without glasses); (4) No history of medical disease or medication which might affect cerebral blood flow; (5) No factor which contraindicates MR scanning, including pacemaker, pacemaker wires, implanted cardiac defibrillator, Neurostimulator, aneurysm clip, or any electronic implant, inner ear surgery, weight over 136 kg (300lb), metal embedded in soft tissue or in the eye, prosthetic eye, or claustrophobics; (6) Not pregnant; (7) Not non-English speaker; (8) Do not smoke tobacco; (9) No non-removable body piercing; (10) No neurological conditions of any of the following: stroke, severe head trauma (motor vehicle accident, loss of consciousness, alteration of consciousness or memory loss), tumor, meningitis, encephalitis, seizure disorder, severe migraine, dementia, epilepsy, any other neurological condition which may contribute to cognitive impairment; (11) No developmental disabilities, including dyslexia and a learning disability; (12) No medical conditions of the following: severe hypertension, severe thyroid dysfunction, severe anemia and/or sickle cell disease, renal failure, heart disease, diabetes; (13) Not use anti-depressants; (14) No history of depression; (15) No regular use of antihistamine; (16) Not an employee of the Neurology Department or a medical students rotating in Neurology; (17) No conditions of any of the following: employee of the PI or employee of a research team member, individual supervised by PI or supervised by member of research team, individual subordinate to the PI or subordinate to any member of the research team, student or trainee under the direction of the PI or under the direction of a member of the research team; (18) Not incompetent or have limited decision-making capacity on initial enrollment into the study; (19) No change of capacity to consent over the course of the study; (20) Not prisoner.

Table 4.4 associates a label with each ROI in NA1 database.

Table 4.4. Regions of Interest (ROI) in the NA1 evaluation database.

ROI	ROI	ROI	ROI
number	name	number	name
0	Background	1	Still unassigned regions
2	Left Cuneus	3	Right Cuneus
4	Left Lateral Occipital Gyrus	5	Right Lateral Occipital Gyrus
6	Left Lingual Gyrus	7	Right Lingual Gyrus
8	Left Calcarine (pericalcarine) Region	9	Right Calcarine (pericalcarine) Region
10	Left Cingulate Gyrus	11	Right Cingulate Gyrus
12	Left Insular Cortex	13	Right Insular Cortex
14	Left Temporal Pole	15	Right Temporal Pole
16	Left Superior Temporal Gyrus	17	Right Superior Temporal Gyrus
18	Left Heschl's Gyrus	19	Right Heschl's Gyrus
20	Left Middle Temporal Gyrus	21	Right Middle Temporal Gyrus
22	Left Inferior Temporal Gyrus	23	Right Inferior Temporal Gyrus
24	Left Fusiform Gyrus	25	Right Fusiform Gyrus
26	Left Entorhinal Cortex	27	Right Entorhinal Cortex
28	Left Parahippocampal Gyrus	29	Right Parahippocampal Gyrus
30	Left Frontal Pole	31	Right Frontal Pole
32	Left Superior Frontal Gyrus	33	Right Superior Frontal Gyrus
34	Left Middle Frontal Gyrus	35	Right Middle Frontal Gyrus
36	Left Inferior Frontal Gyrus	37	Right Inferior Frontal Gyrus
38	Left Orbital Gyri	39	Right Orbital Gyri
40	Left Paracentral Lobule	41	Right Paracentral Lobule
42	Left Precentral Gyrus	43	Right Precentral Gyrus
44	Left Postcentral Gyrus	45	Right Postcentral Gyrus
46	Left Superior Parietal Lobule	47	Right Superior Parietal Lobule
48	Left Inferior Parietal Lobule	49	Right Inferior Parietal Lobule
50	Left Precuneus	51	Right Precuneus
52	Caudate.	53	Putamen.
54	Globus Pallidus.	55	Nucleus Accumbens
56	Thalamus	57	Lateral Ventricles

4.2.7.1 Collapse Example 1: Collapse of the Thalamus

Figure 4.23 shows the template (patient ID: 3065) and target (patient ID: 3424) brain MR images before registration. The columns show the axial, sagittal and coronal views of the template, target, and difference images. Green in the difference image indicates regions where only the template contains image data; red indicates regions where only the target contains image data; and yellow indicates regions where both the template and target contain image data. The original 3065 and 3424 images had the brain located too close to the superior end of the image, causing the transformation to be affected by the boundary condition (this will be discussed in detail in Section 4.2.8). So, both images were shifted downward towards the center of the image before registration.

Figure 4.24 shows the orthogonal views of the deformed segmentation masks after registration, and also the collapse magnitude image. In both the template and target masks, the thalami at the center (dark orange) can clearly be seen. However, after registration, as indicated by the red arrows in the third row, we can see that the thalamus has collapsed. The collapse magnitude image in the fourth row confirms the collapse, with bright regions in the center indicate regions of large collapse. Figure 4.25 3D rendering of the template thalamus, target thalamus, and the collapsed template thalamus after registration. Also, a 3D rendering (from three different angles) of the bright region of the collapse magnitude image is shown in Fig. 4.26.

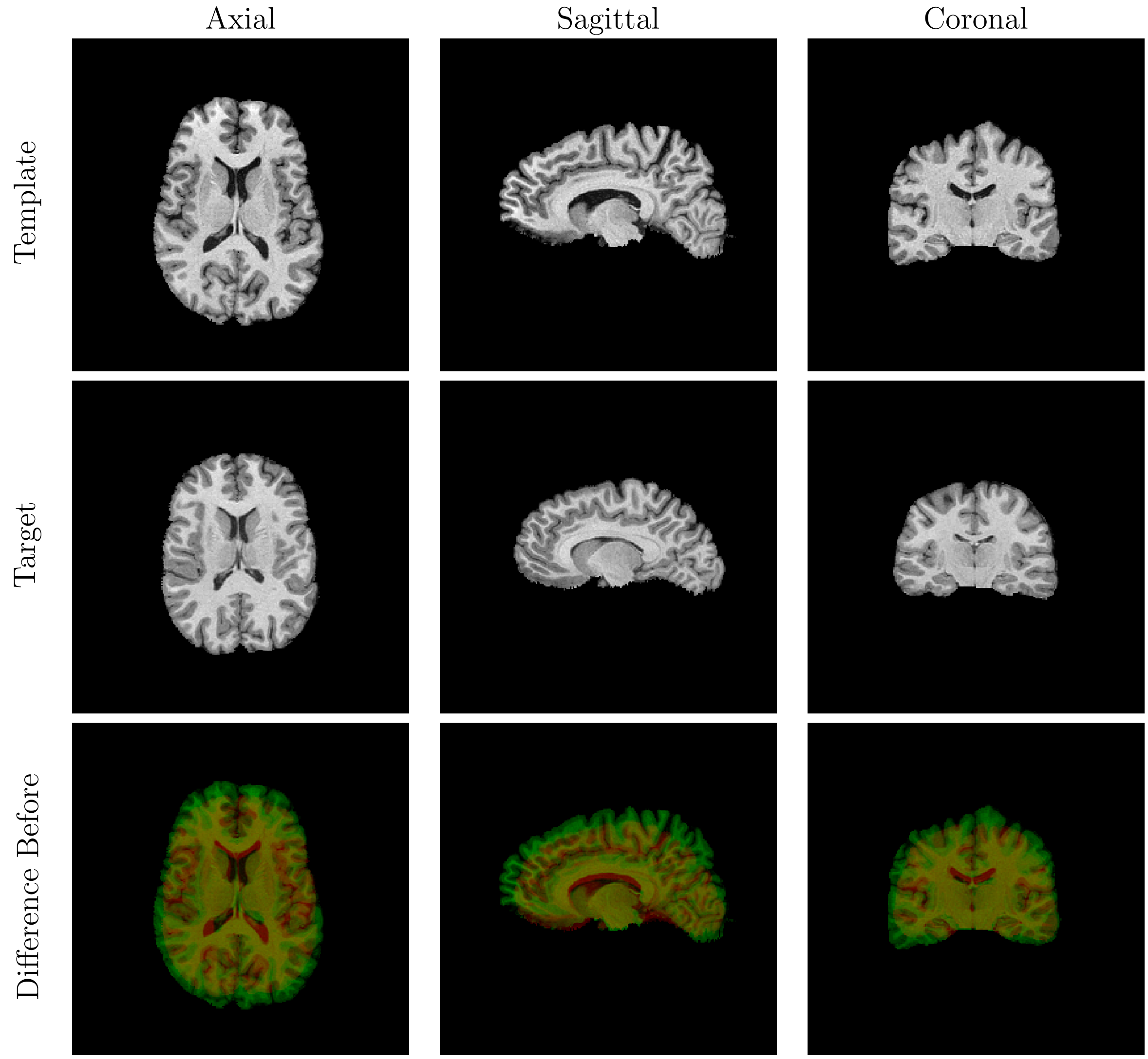


Figure 4.23. Template (patient ID: 3065) and target (patient ID: 3424) image pair and their difference before registration. The first two rows show the three orthogonal views of the template and target data. The last row shows the difference of the template and target, with green indicating regions where only the template contains image data; red indicating regions where only the target contains image data; and yellow indicating regions where both the template and target contain image data.

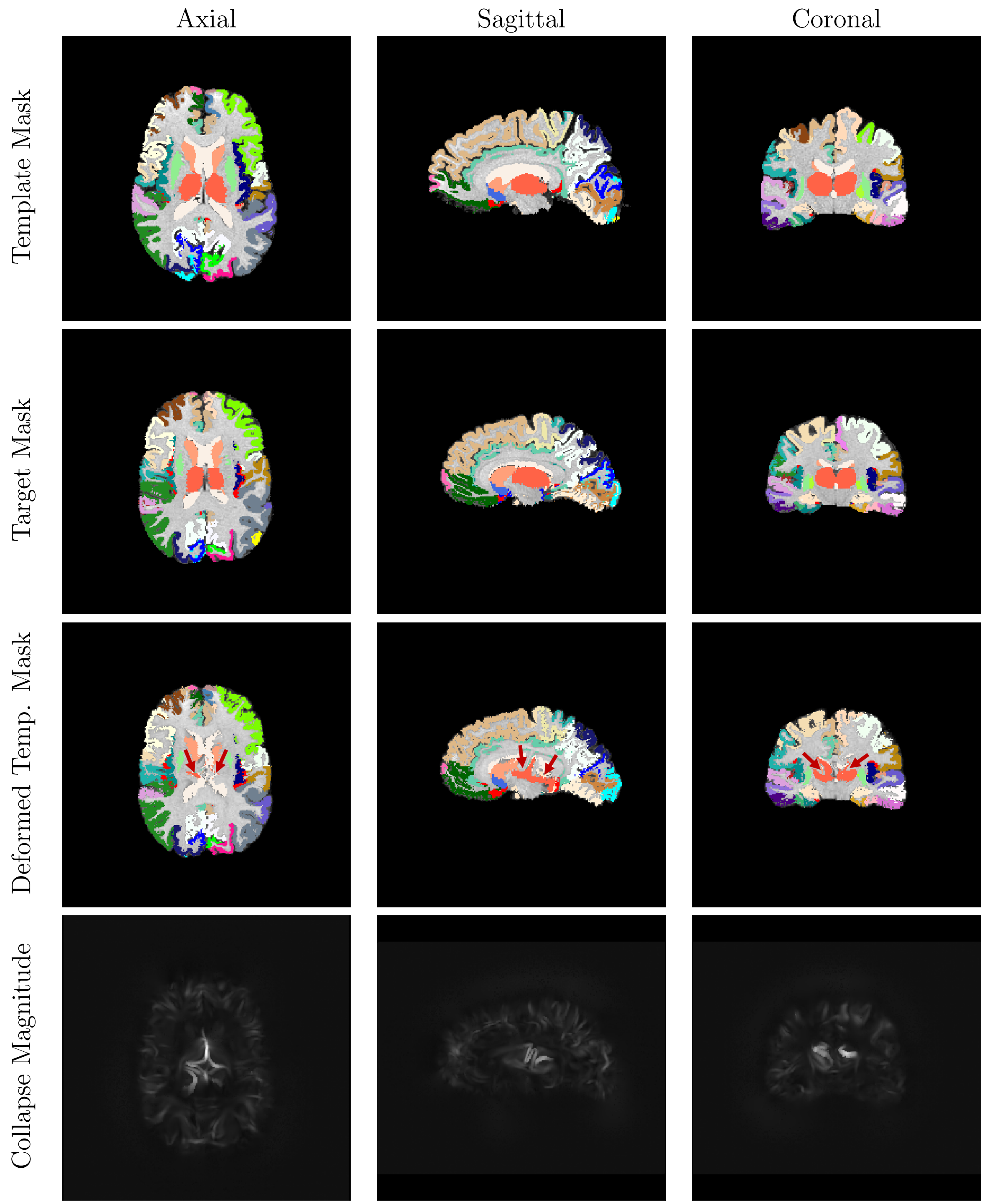


Figure 4.24. Orthogonal views of the template, target, and deformed template mask after registration. The images in the last row shows the collapse magnitude image, where bright areas indicate areas of large collapse. Collapse of the thalamus (red arrows in the third row) is observed after image registration.

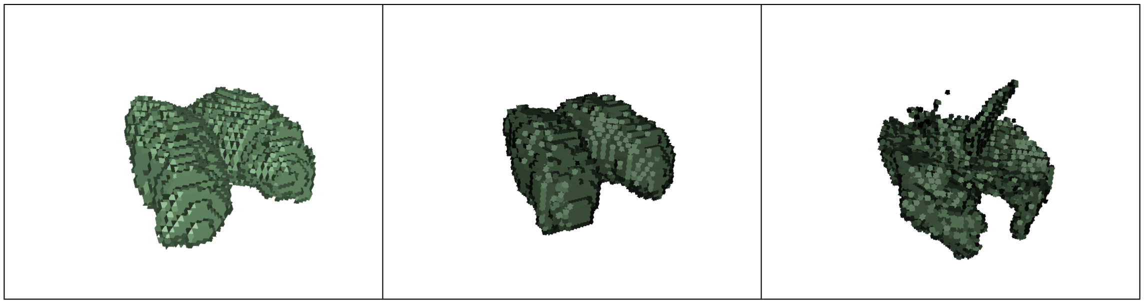


Figure 4.25. A 3D rendering of the template thalamus, target thalamus, and the collapsed template thalamus after registration. Notice how the template thalamus collapsed and does not resemble the shape of the template thalamus.

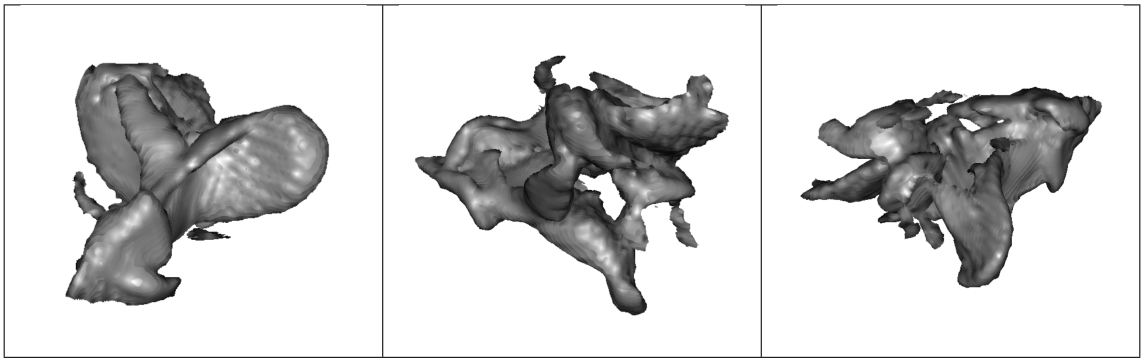


Figure 4.26. Three different 3D rendering from different angles of the collapse magnitude image in the thalamus region. This is not the rendering of the actual collapsed thalamus (as shown in Fig. 4.25, but instead, the rendering of the regions of greatest collapse magnitude.

4.2.7.2 Collapse Example 2: Collapse of the Cortex

It was also observed that shape collapse occurred in the cortical region. Figure 4.27 shows the same brain MR data of patient 3065 and 3424 as shown in Fig. 4.23, except that it highlights the orthogonal plane where the mismatch between the cortical regions are large.

Figure 4.28 shows the orthogonal views of the template, target and deformed template segmentation masks, and the collapse magnitude image after registration. The red arrows in the third row indicate where shape collapse likely has occurred. Compared to the collapse of the thalamus shown in the previous example, the collapse of the cortex is not immediately obvious. However, the bright highlights near this region in the collapse magnitude image (particularly visible in the sagittal and coronal views) suggest that a collapse has occurred. This serves at explanation as to how the white cortical region flowed into (or, collapsed into) the brown cortical region.

This example demonstrates that it is very difficult to detect shape collapse by analyzing the deformed image or the segmentation mask. The collapse magnitude image serves as an indicator to highlight regions where shape collapse may have occurred. While not all bright points in the collapse magnitude image indicate shape collapse, but areas with high collapse magnitude strongly suggest that a shape collapse may have occurred and demand further analysis. There is not yet a clear-cut way to determine whether a collapse of certain magnitude is a result of an actual shape collapse or just large deformation. Simple thresholding of the collapse magnitude image does not clearly isolate real collapsing points from the false ones. At this

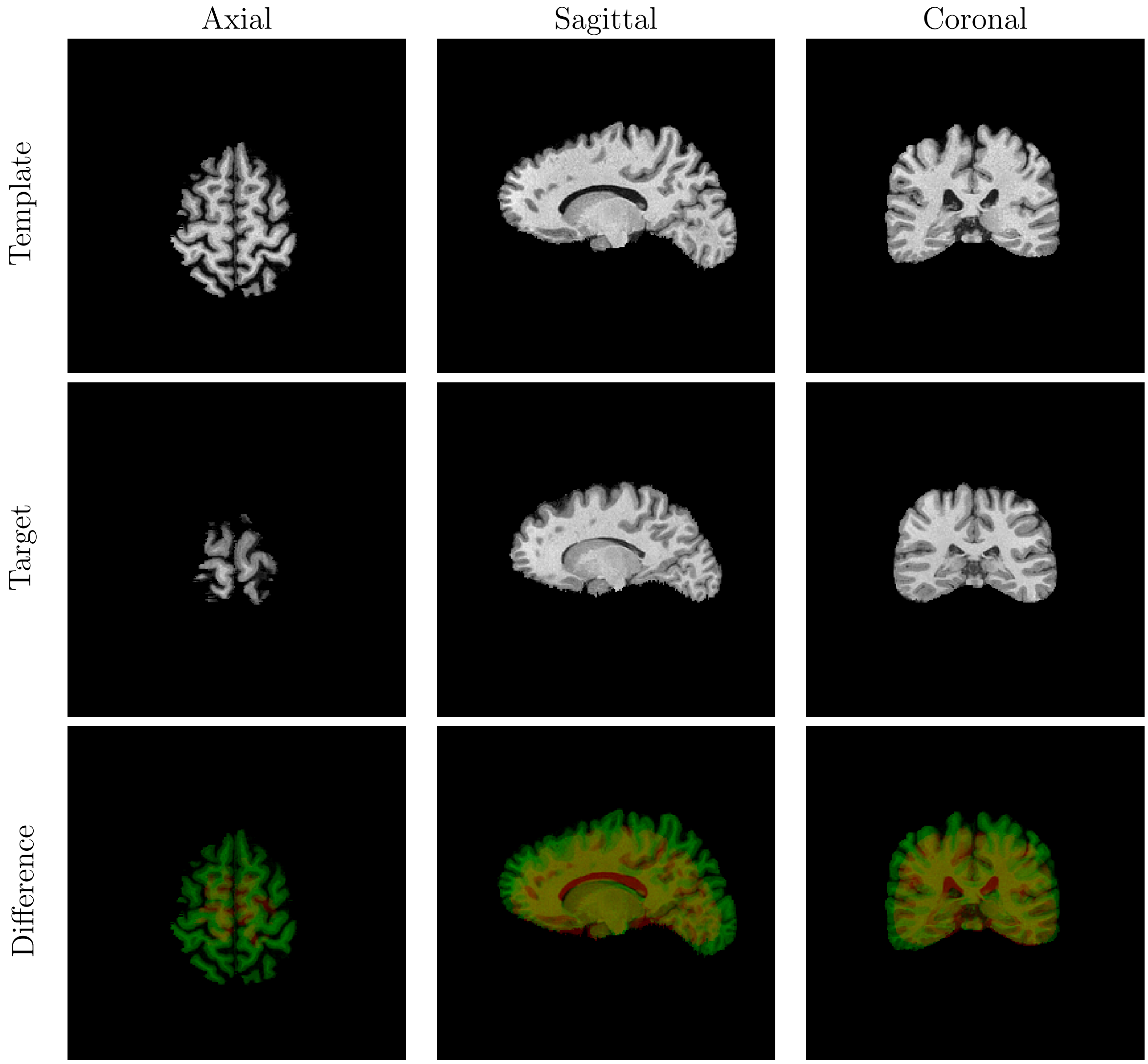


Figure 4.27. Template (patient ID: 3065) and target (patient ID: 3424) image pair and their difference before registration. These are the same exact brain MR images as shown in Fig. 4.23, except that the orthogonal views show where the mismatch between the cortical regions are large.

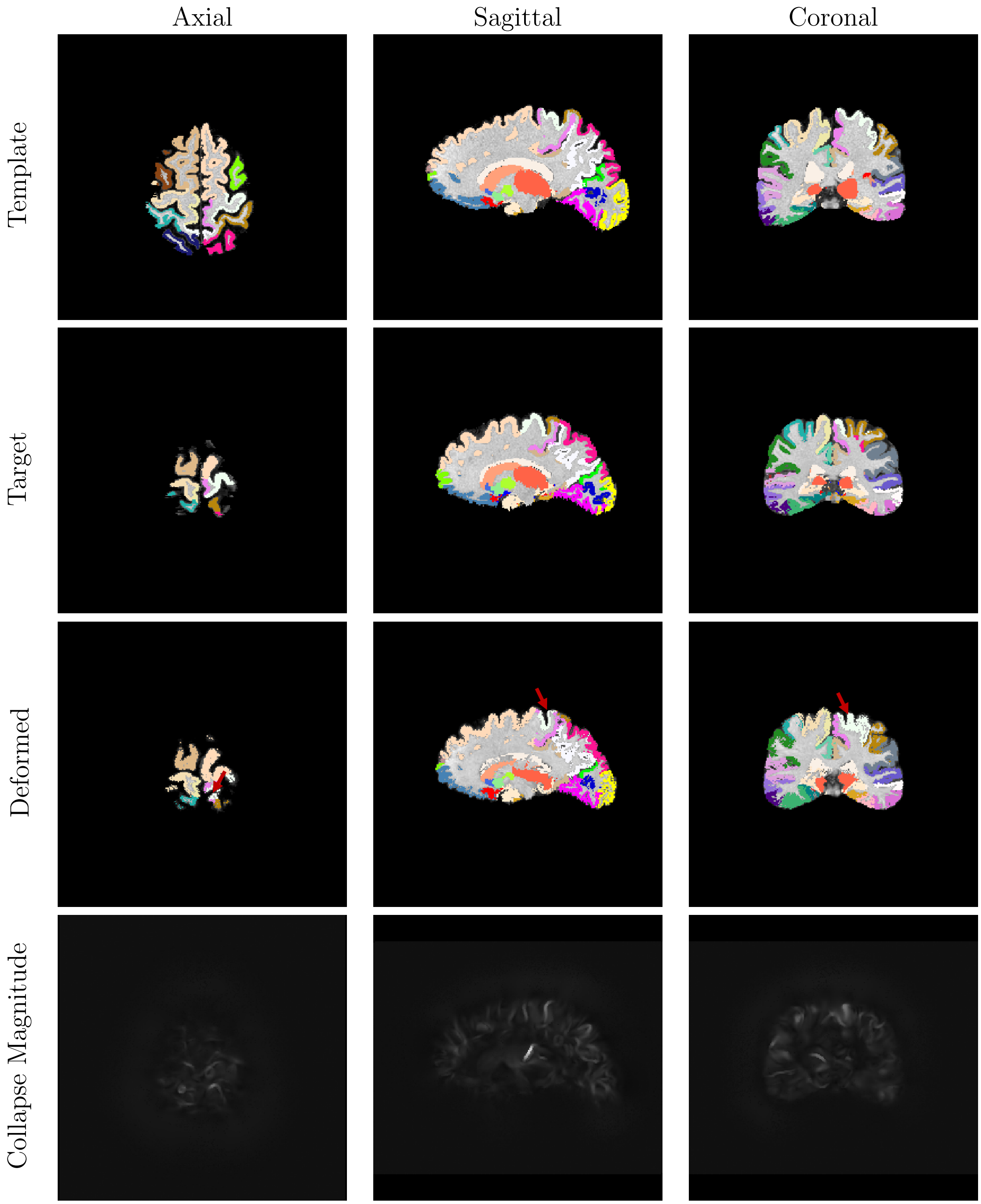


Figure 4.28. Orthogonal views of the template, target, and deformed template mask after registration. The images in the last row shows the collapse magnitude image, where bright areas indicate areas of large collapse. Collapse of the cortical region (red arrows in the third row) is observed after image registration.

point, only visual analysis of suspected regions is the recommended approach, but a more formal definition of collapsing points based on collapse magnitude is left for future work.

4.2.8 Boundary Conditions and Shape Collapse

In the previous section, it was noted that the original images for patient 3065 and 3424 had to be shifted downward to be centered for registration. In this section, what happens to the shape collapse if the images are not shifted (i.e., when the deformation occurs near the image boundary). Figure 4.29 shows the uncentered version of the images in Fig. 4.23. It is clear from the sagittal and coronal views that the brain is located near the superior boundary.

Figure 4.30 shows the collapse magnitude image after registration. Notice the bright highlights near the superior boundary. The shape of the collapse magnitude near the boundary suggests that objects are being "squished" near the image boundary. Figure 4.31 shows the 3D representation of the collapse magnitude images of the same registration pair generated by registering (a) uncentered; and (b) centered images. In Fig. 4.31(a), the collapse magnitude image shows deformations happening within the brain boundary. However in Fig. 4.31(b), there is large collapse magnitude observed near the superior boundary, which flattens out as it gets closer to the boundary. The possible explanation of this phenomenon is that the transformation is parametrized in such a way that zero deformation is allowed near the image boundary. Thus, all deformations between the image boundary and the brain are concentrated near the upper boundary of the brain, creating the "squished" effect. If the registra-

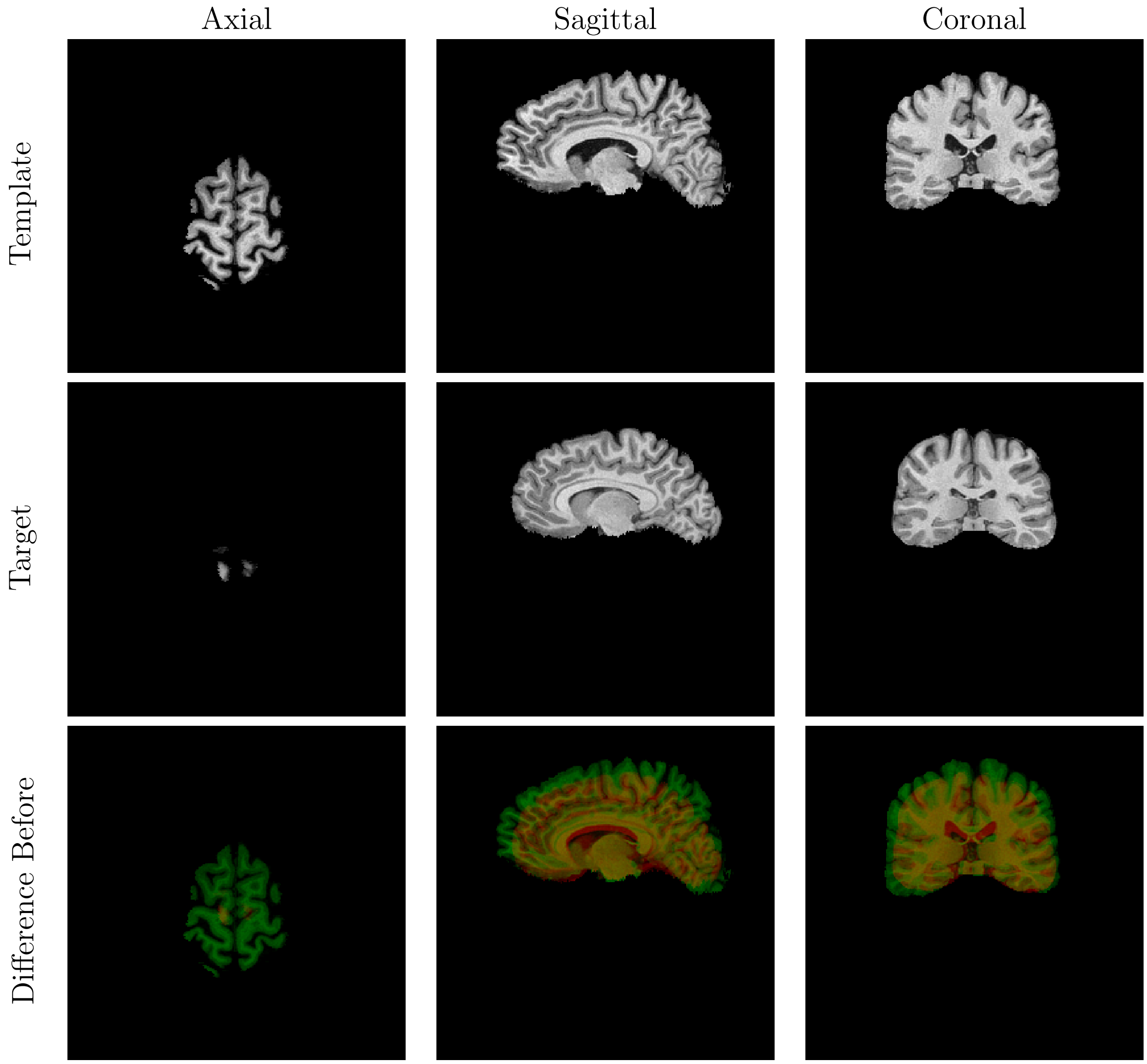


Figure 4.29. Template (patient ID: 3065) and target (patient ID: 3424) image pair and their difference before registration. These are the same exact brain MR images as shown in Fig. 4.23, except that the images are not centered. As it can be seen from the sagittal and coronal views, the brain is located near the superior boundary.

tion algorithm does not suffer from this boundary condition (i.e., deformations are allowed at the boundary), this kind of squishing would not occur.

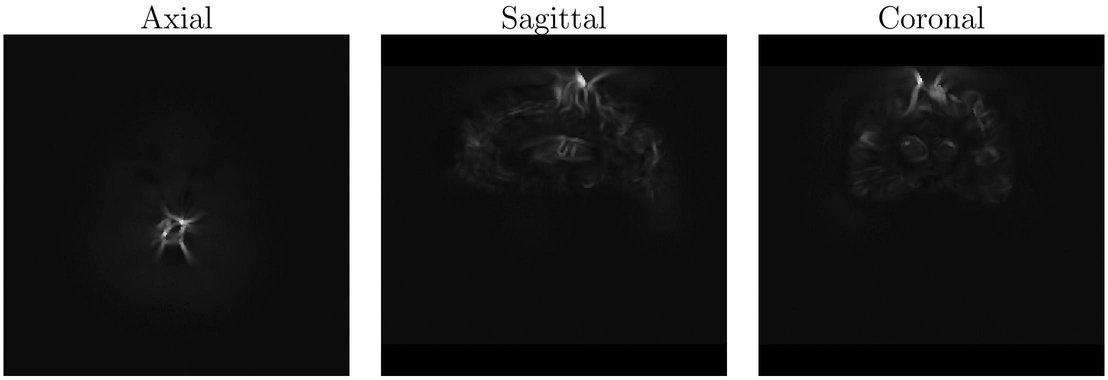


Figure 4.30. Orthogonal view near the superior boundary of the collapse magnitude image after registration. Notice the bright highlights near the boundary of the image, which means that objects are being “squished” near the boundary.

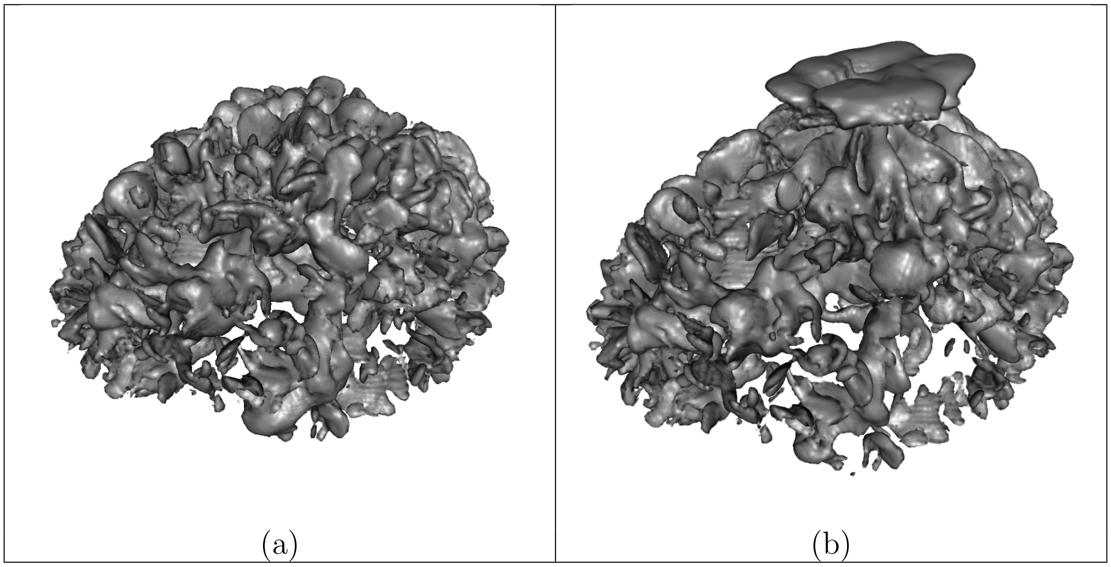


Figure 4.31. Collapse magnitude images of the same registration pair generated by registering (a) uncentered images; and (b) centered images.

CHAPTER 5

SUMMARY AND CONCLUSIONS

The goal of this thesis was to create an evaluation framework for evaluating the performance of nonrigid image registration algorithms. This thesis is a product of the nonrigid image registration evaluation project (NIREP) [18]. This thesis gives guidance to which image registration algorithms work best for certain registration problems and why. It also provides insights as to how image registration algorithm may perform on other types image data sets that were not evaluated in this text.

A major challenge of evaluating image registration performance is the lack of a "Gold Standard," i.e., the true correspondence between two arbitrary images is unknown. A second major challenge is there are many image registration problems in which there may be many different correspondence maps that give the same result, i.e., there is no unique correspondence map. An example of this occurs for generating an automatic segmentation of a target image using an atlas. In this task, the atlas is registered to a target image and then the estimated transformation is used to map the atlas segmentation to the target image. In this case, there are an infinite number of transformations that can produce the same automatic segmentation.

The approach taken in this thesis to overcome these problems and evaluate image registration performance was to (1) break each image registration algorithm into its basic components (transformation model, matching criteria, and regularization) in order to isolate and understand how each component affects registration performance; (2) compare image registration algorithms on common evaluation databases; and (3) evaluate image registration algorithm performance using many evaluation tests. The advantage of using multiple evaluation tests is that one can characterize different aspects of image registration algorithm performance. Characterizing registration algorithm performance can be used to predict performance of a particular image registration algorithm on similar types of images and image registration problems.

5.1 Image Registration Algorithm Component Analysis

One of the contributions of this thesis is to examine image registration algorithm performance at the component level. An image registration algorithm can be decomposed into four basic components: transformation model, similarity cost function, regularization cost function and optimization. This thesis discussed the impact that the first three of these components have on image registration performance. The impact that optimization approaches have on image registration performance was not addressed in this thesis. Some of the reasons why the optimization component was excluded from this work were: (1) changing the optimization method can reduce computation time but does not necessarily provide better correspondence; and (2) reimplementing a particular image registration algorithm multiple times with different optimization methods requires a lot of programming effort and was hence beyond the scope of this project.

This thesis catalogs many of the common small and large deformation image registration transformation models. The advantages, limitations and degrees of freedom of each transformation model were discussed. In general, the degrees of freedom is a good way to characterize the complexity of a transformation model with respect to computation needs, storage needs, and flexibility to represent complex shape deformations. Issues related to transformation models comprised of basis functions with infinite support vs. local support were discussed. This thesis catalogs many of the common image registration similarity cost functions and regularization cost functions. In each case, the pros and cons of each cost function were discussed. The importance of regularization in an image registration algorithm was demonstrated first hand in this thesis when it was used to mitigate shape collapse.

5.2 Image Registration Algorithm Evaluation

One of the contributions of this thesis is to catalog the benefits and limitations of many of the most commonly used image registration evaluation approaches. For example, one of the best approaches to evaluate image registration accuracy is to measure the distance between a transformed landmark and it's corresponding landmark. Unfortunately, corresponding landmark evaluation has many limitations including: (1) it only measures the accuracy of a transformation at the landmarks; (2) landmarking is time consuming; (3) there is error in selecting landmarks; (4) landmarking often has to be performed by trained professionals such as doctors; and (5) not every object of interest has landmarks (e.g., a smooth surface). Another example is that inverse consistency is a necessary condition for good correspondence but it is not sufficient. For example, Section 3.1.8 demonstrated that although the SyN image registration algorithm had zero inverse consistency error by construction, it still had midpoint landmark error, i.e., correspondence error.

Evaluation tests provide a means for quantifying image registration performance and allows us to rank image registration algorithms based on how well they perform specific tasks. This thesis did not attempt to analyze every single existing image registration algorithm with every single evaluation approach, but rather provided examples and guidance of how image registration algorithms can be evaluated and ranked. To this end, the performance of many commonly used image registration algorithms were evaluated and ranked based on different evaluation criteria.

Finally, one incremental contribution of this thesis was to demonstrate how existing evaluation methods can be applied in the midpoint coordinate system to evaluate some symmetric image registration algorithms such as the SyN registration algorithm.

5.3 Detecting, Visualizing, and Mitigating Shape Collapse

A major contribution of this thesis was to develop tools to evaluate and visualize 2D and 3D image registration shape collapse. This thesis demonstrates that many current diffeomorphic image registration algorithms suffer from the collapse problem. The collapse problem occurs when a set with some finite area/volume collapses to a set of near zero measure. We demonstrated that both the foreground and the background of an image can collapse. The growth problem can be considered a dual problem of the collapse problem. A growth occurs when a region of the template image expands to fill in a region in the target image that is missing in the template image (See Fig. 3.8).

This thesis provides the first visualizations of the collapse problem in 3D for simple shapes and real human brain MR images. To do this, we extended tools from 2D to 3D for predicting and visualizing where shape collapse may occur based on the overlap of binary template and target images. The extension from 2D to 3D was not trivial. Generating the skeleton of an object in 2D is an easier problem than generating a skeleton of a 3D object. In addition, most skeletonization algorithms only generate skeletons of the foreground object but our application requires skeletonization of both the foreground and background. Skeletonizing the image background is different than skeletonizing the foreground since the boundary of the image is not considered part of the background object. To overcome these issuses, we worked with Dakai Jin to modify his novel skeletonization method [38] to skeletonize both the foreground and background of a 3D binary image. Further, we developed software to identify the surface collapse points and visualize them.

The tools developed to visualize predicted shape collapse and actual shape collapse lead to the discovery of one-sided shape collapse. This thesis is the first to characterize one-sided shape collapse and it provides the first 2D and 3D visualizations of one-sided collapse. This discovery also required us to refine our notion of shape collapse as either one-sided or two-sided.

This thesis provides the first experiments that demonstrate how adjusting image registration parameters can mitigate the collapse problem to some extent. In these experiments, we showed how adjusting the appropriate parameters in the SyN algorithm could reduce shape collapse. However, the cost for mitigating the collapse was that the SyN algorithm produced worse registration with respect to foreground overlap.

5.4 Future Work

It is our hope that the evaluation framework presented in this thesis will be continued to be expanded upon in the future. We also hope that the insights gained by understanding the benefits, drawbacks and limitations of the individual components that make up an image registration algorithm will help produce the next generation of improved image registration algorithms.

REFERENCES

[1] Y. Amit. A non-linear variational problem for image matching. *SIAM Journal on Scientific Computation*, 15(1):207–224, January 1994.

[2] Vincent Arsigny, Olivier Commowick, Xavier Pennec, and Nicholas Ayache. A log-euclidean framework for statistics on diffeomorphisms. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2006*, pages 924–931. Springer, 2006.

[3] John Ashburner. A fast diffeomorphic image registration algorithm. *Neuroimage*, 38(1):95–113, 2007.

[4] B. B. Avants, C. L. Epstein, M. Grossman, and J. C. Gee. Symmetric diffeomorphic image registration with cross-correlation: evaluating automated labeling of elderly and neurodegenerative brain. *Med Image Anal*, 12(1):26–41, Feb 2008.

[5] Brian Avants, Nicholas James Tustison, michael stauffer, Gang Song, Baohua Wu, and James Gee. The insight toolkit image registration framework. *Frontiers in Neuroinformatics*, 8(44), 2014.

[6] Brian B Avants, Nicholas J Tustison, Gang Song, Philip A Cook, Arno Klein, and James C Gee. A reproducible evaluation of ants similarity metric performance in brain image registration. *Neuroimage*, 54(3):2033–2044, 2011.

[7] R. Bajcsy and S. Kovacic. Multiresolution Elastic Matching. *Computer Vision, Graphics, and Image Processing*, 46:1–21, 1989.

[8] M. Faisal Beg, Michael I. Miller, Alain Trouvé, and Laurent Younes. Computing large deformation metric mappings via geodesic flows of diffeomorphisms. *International Journal of Computer Vision*, 61(2):139–157, 2005.

[9] P. J. Besl and N. D. McKay. A method for registration of 3-D shapes. *IEEE Trans. Pattern Anal. and Machine Intelligence*, 14(2):239–256, 1992.

[10] Matias Bossa, Ernesto Zacur, and Salvador Olmos. Algorithms for computing the group exponential of diffeomorphisms: Performance evaluation. In *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference on*, pages 1–8. IEEE, 2008.

[11] R. C. Buck. *Advanced Calculus.* McGraw-Hill Book Company, St. Louis, 3 edition, 1978.

[12] G. E. Christensen. *Deformable Shape Models for Anatomy.* PhD thesis, Department of Electrical Engineering, Sever Institute of Technology, Washington University, St. Louis, MO. 63130, Aug. 1994.

[13] G. E. Christensen. Consistent linear-elastic transformations for image matching. In A. Kuba and M. Samal, editors, *Information Processing in Medical Imaging*, LCNS 1613, pages 224–237, Berlin, June 1999. Springer-Verlag.

[14] G. E. Christensen and H. J. Johnson. Consistent image registration. *IEEE Trans. Med. Imaging*, 20(7):568–582, July 2001.

[15] G. E. Christensen and H. J. Johnson. Invertibility and transitivity analysis for nonrigid image registration. *Journal of Electronic Imaging*, 12(1):106–117, Jan. 2003.

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

[17] G. E. Christensen, R. D. Rabbitt, M. I. Miller, S.C. Joshi, U. Grenander, T.A. Coogan, and D.C. Van Essen. Topological properties of smooth anatomic maps. In Y. Bizais, C. Braillot, and R. Di Paola, editors, *Information Processing in Medical Imaging*, volume 3, pages 101–112, June 1995.

[18] Gary E. Christensen, Xiujuan Geng, Jon G. Kuhl, Joel Bruss, Thomas J. Grabowski, John S. Allen, Imran A. Pirwani, Michael W. Vannier, and Hanna Damasio. Introduction to the non-rigid image registration evaluation project (nirep). In *3rd International Workshop on Biomedical Image Registration*, LCNS 4057, pages 128–135. Springer-Verlag, July 2006.

[19] A. Collignon, F. Maes, D. Delaere, D. Vandermeulen, P. Suetens, and G. Marchal. Automated multi-modality image registration based on information theory. In Y. Bizais, C. Braillot, and R. Di Paola, editors, *Information Processing in Medical Imaging*, volume 3, pages 263–274. Kluwer Academic Publishers, Boston, June 1995.

[20] W.R. Crum, T. Hartkens, and D.L.G. Hill. Non-rigid imgag registration: theory and practice. *The British Journal of Radiology*, 77:140–153, 2004.

[21] R.S. Desikan, F. Segonne, B. Fischl, B.T. Quinn, B.C. Dickerson, D. Blacker, R.L. Buckner, A.M. Dale, R.P. Maguire, B.T. Hyman, M.S. Albert, and R.J. Killiany. An automated labeling system for subdividing the human cerebral cortex on mri scans into gyral based regions of interest. *NeuroImage*, 31:968–80, 2006.

[22] Paul Dupuis, Ulf Grenander, and Michael I. Miller. Variational problems on flows of diffeomorphisms for image matching. *Quarterly of Applied Mathematics*, 56(3):587–600, 1998.

[23] Stanley Durrleman. *Statistical models of currents for measuring the variability of anatomical curves, surfaces and their evolution.* PhD thesis, Université Nice Sophia Antipolis, 2010.

[24] Stanley Durrleman, Marcel Prastawa, Nicolas Charon, Julie R Korenberg, Sarang Joshi, Guido Gerig, and Alain Trouvé. Morphometry of anatomical shape complexes with dense deformations and sparse parameters. *NeuroImage*, 101:35–49, 2014.

[25] Oguz C Durumeric, Ipek Oguz, and Gary E Christensen. The shape collapse problem in image registration. In *Mathematical Foundations of Computational Anatomy*, pages 95–106.

[26] J. Michael Fitzpatrick. The Retrospective Image Registration Evaluation Project. http://www.insight-journal.org/RIRE/, March 2007.

[27] J. Michael Fitzpatrick. Fiducial registration error and target registration error are uncorrelated, 2009.

[28] J.M. Fitzpatrick, D.L.G. Hill, and C.R. Maurer. Image registration. In M. Sonka and J.M. Fitzpatrick, editors, *Handbook of Medical Imaging*, volume 2, chapter 8, pages 447–513. SPIE Press, San Diego, 2000.

[29] Mark Foskey, Brad Davis, Lav Goyal, Sha Chang, Ed Chaney, Nathalie Strehl, Sandrine Tomei, Julian Rosenman, and Sarang Joshi. Large deformation three-dimensional image registration in image-guided radiation therapy. *Physics in Medicine and Biology*, 50(24):5869, 2005.

[30] X. Geng, D. Kumar, and G. E. Christensen. Transitive inverse-consistent manifold registration. In *Information Processing in Medical Imaging*, volume LNCS 3564, pages 468–479, Berlin, July 2005. Springer-Verlag.

[31] U. Grenander and M. I. Miller. Computational anatomy: An emerging discipline. *Quarterly of Applied Mathematics*, LVI(4):617–694, December 1998.

[32] Xuejun Gu, Hubert Pan, Yun Liang, Richard Castillo, Deshan Yang, Dongju Choi, Edward Castillo, Amitava Majumdar, Thomas Guerrero, and Steve B Jiang. Implementation and evaluation of various demons deformable image registration algorithms on a gpu. *Physics in Medicine and Biology*, 55(1):207, 2010.

[33] J. He and G. E. Christensen. Large deformation inverse consistent elastic image registration. In C. Taylor and A. Noble, editors, *Information Processing in Medical Imaging*, LCNS 2732, pages 438–449, Berlin, July 2003. Springer-Verlag.

[34] P. Hellier, C. Barillot, L. Corouge, B. Gibaud, G. Le Goualher, D. L. Collins, A. Evans, G. Malandain, N. Ayache, G. E. Christensen, and H. J. Johnson. Retrospective evaluation of inter-subject brain registration. *IEEE Transactions on Medical Imaging*, 22(9):1120–1130, 2003.

[35] Monica Hernandez, Matias N Bossa, and Salvador Olmos. Registration of anatomical images using paths of diffeomorphisms parameterized with stationary vector field flows. *International Journal of Computer Vision*, 85(3):291–306, 2009.

[36] W.M. Wells III, P. Viola, H. Atsumi, S. Nakajima, and R. Kikinis. Multi-modal volume registration by maximization of mutual information. *Medical Image Analysis*, 1(1):35–51, 1996.

[37] Pierre Jannin, Christophe Grova, and Jr. Calvin R. Maurer. Model for defining and reporting reference-based validation protocols in medical image processing. *Int Journ Comput Assisted Radiol and Surg*, 1(2):1001–1115, 2006.

[38] Dakai Jin and Punam K. Saha. *A New Fuzzy Skeletonization Algorithm and Its Applications to Medical Imaging*, pages 662–671. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[39] H. J. Johnson and G. E. Christensen. Consistent landmark and intensity-based image registration. *IEEE Trans. Med. Imaging*, 21(5):450–461, 2002.

[40] S. Kabus, T. Klinder, K. Murphy, B. van Ginneken, C. Lorenz, and J. P. W. Pluim. Evaluation of 4D-CT lung registration. In *MICCAI*, pages 747–754, 2009.

[41] Arno Klein, Jesper Andersson, Babak A. Ardekani, John Ashburner, Brian Avants, Ming-Chang Chiang, Gary E. Christensen, D. Louis Collins, James Gee, Pierre Hellier, Joo Hyun Song, Mark Jenkinson, Claude Lepage, Daniel Rueckert, Paul Thompson, Tom Vercauteren, Roger P. Woods, J. John Mann, and Ramin V. Parsey. Evaluation of 14 nonlinear deformation algorithms applied to human brain MRI registration. *NeuroImage*, 46(3):786–802, July 2009.

[42] Arno Klein, Satrajit S. Ghosh, Brian Avants, B.T.T. Yeo, Bruce Fischl, Babak Ardekani, James C. Gee, J.J. Mann, and Ramin V. Parsey. Evaluation of volume-based and surface-based brain image registration methods. *NeuroImage*, 51(1):214 – 220, 2010.

[43] F. Maes, A. Collignon, D. Vandermeulen, G. Marchal, and P. Suetens. Multimodality image registration by maximization of mutual information. *IEEE Proceedings of Mathematical Methods in Biomedical Image Analysis*, pages 309–318, June 1996.

[44] F. Maes, A. Collignon, D. Vandermeulen, G. Marchal, and P. Suetens. Multimodality image registration by maximization of mutual information. *IEEE Transactions on Medical Imaging*, 16(2):187–198, April 1997.

[45] Frederik Maes, Dirk Vandermeulen, and Paul Suetens. Medical image registration using mutual information. *Proceedings of IEEE*, 91(10):1699–1722, October 2003.

[46] J. Maintz and M. Viergever. A survey of medical image registration. *Medical Image Analysis*, 1998.

[47] Calvin R. Maurer, Jr., Rensheng Qi, and Vijay Raghavan. A linear time algorithm for computing exact euclidean distance transforms of binary images in arbitrary dimensions. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(2):265–270, February 2003.

[48] M. I. Miller, A. Banerjee, G. E. Christensen, S. C. Joshi, N. Khaneja, U. Grenander, and L. Matejic. Statistical methods in computational anatomy. *Statistical Methods in Medical Research*, 6:267–299, 1997.

[49] M. I. Miller, G. E. Christensen, Y. Amit, and U. Grenander. Mathematical textbook of deformable neuroanatomies. *Proceedings of the National Academy of Sciences*, 90(24):11944–48, December 1993.

[50] Michael I Miller, Alain Trouvé, and Laurent Younes. Geodesic shooting for computational anatomy. *Journal of mathematical imaging and vision*, 24(2):209–228, 2006.

[51] Keelin Murphy, Bram Van Ginneken, J Reinhardt, Sven Kabus, Kai Ding, Xiang Deng, and J Pluim. Evaluation of methods for pulmonary image registration: The EMPIRE10 study. *Grand Challenges in Medical Image Analysis*, 2010:11–22, 2010.

[52] Keelin Murphy, Bram Van Ginneken, Joseph M Reinhardt, Sven Kabus, Kai Ding, Xiang Deng, Kunlin Cao, Kaifang Du, Gary E Christensen, Vincent Garcia, et al. Evaluation of registration methods on thoracic ct: the EMPIRE10 challenge. *Medical Imaging, IEEE Transactions on*, 30(11):1901–1920, 2011.

[53] Xavier Pennec, Pascal Cachier, and Nicholas Ayache. *Understanding the “Demon’s Algorithm”: 3D Non-rigid Registration by Gradient Descent*, pages 597–605. Springer Berlin Heidelberg, Berlin, Heidelberg, 1999.

[54] J.P.W. Pluim, J.B.A. Maintz, and M.A. Viergever. Mutual-information-based registration of medical images: a survey. *IEEE Transactions on Medical Imaging*, 22(8):986–1004, 2003.

[55] 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. *Medical Imaging, IEEE Transactions on*, 18(8):712 –721, aug. 1999.

[56] 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, 1999.

[57] D. Shen and C. Davatzikos. Hammer: hierarchical attribute matching mechanism for elastic registration. *IEEE Trans. on Medical Imaging*, 21(11):1421–1439, Dec 2002.

[58] D. Shen and C. Davatzikos. Very high-resolution morphometry using mass-preserving deformations and hammer elastic registration. *NeuroImage*, 18(1):28–41, Jan 2003.

[59] Joo Song, Gary Christensen, Jeffrey Hawley, Ying Wei, and Jon Kuhl. Evaluating image registration using nirep. In Bernd Fischer, Benot Dawant, and Cristian Lorenz, editors, *Biomedical Image Registration*, volume 6204 of *Lecture Notes in Computer Science*, pages 140–150. Springer Berlin / Heidelberg, 2010.

[60] Milan Sonka, Vaclav Hlavac, and Roger Boyle, editors. *Image Processing, Analysis, and Machine Vision*, volume 1. Brooks/Cole Publishing, Pacific Grove, CA, 2 edition, 1998.

[61] C. Studholme, D.L.G. Hill, and D.J. Hawkes. Incorporating connected region labelling into automated image registration using mutual information. *IEEE Proceedings of Mathematical Methods in Biomedical Image Analysis*, pages 23–30, June 1996.

[62] Colin Studholme, Derek L. G. Hill, and David J. Hawkes. Automated three-dimensional registration of magnetic resonance and positron emission tomography brain images by multiresolution optimization of voxel similarity measures. *Med. Phys.*, 24(1):25–35, 1997.

[63] J.P. Thirion. Image matching as a diffusion process: an analogy with maxwell's demons. *Medical Image Analysis*, 2:243–260, 1998.

[64] Alain Trouvé. An inifinite dimensional group approach for physics based models in patterns recognition. *Int. J. Comput. Vis*, 28(3):213–221, 1995.

[65] N. Tustison and J. Gee. Introducing dice, jaccard, and other label overlap measures to itk. 12 2009.

[66] Tom Vercauteren, Xavier Pennec, Aymeric Perchant, and Nicholas Ayache. Non-parametric diffeomorphic image registration with the demons algorithm. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2007*, pages 319–326. Springer, 2007.

[67] Tom Vercauteren, Xavier Pennec, Aymeric Perchant, and Nicholas Ayache. Symmetric log-domain diffeomorphic registration: A demons-based approach. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2008*, pages 754–761. Springer, 2008.

[68] Tom Vercauteren, Xavier Pennec, Aymeric Perchant, and Nicholas Ayache. Diffeomorphic demons: Efficient non-parametric image registration. *NeuroImage*, 45(1, Supplement 1):S61 – S72, 2009. Mathematics in Brain Imaging.

[69] P. Viola and W. M. Wells III. Alignment by maximization of mutual information. *International Journal of Computer Vision*, 24(2):137–154, 1997.

[70] Ying Wei, Gary E. Christensen, Joo Hyun Song, David Rudrauf, Joel Bruss, Jon G. Kuhl, and Thomas J. Grabowski. Evaluation of five non-rigid image registration algorithms using the nirep framework. *Medical Imaging 2010: Image Processing*, 7623(1):76232L, 2010.

[71] Jay West, J. Michael Fitzpatrick, et al. Comparison and evaluation of retrospective intermodality brain image registration techniques. *J. Comp. Asst. Tomog.*, 21(4):554–566, 1997.

[72] 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(4):620–633, July/August 1992.

[73] 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(1):139–152, 1998.

[74] R.P. Woods, S.T Grafton, J.D. Watson, N.L. Sicotte, and J.C. Mazziotta. Automated Image Registration: II. Intersubject Validation of Linear and Nonlinear Models. *Journal of Computer Assisted Tomography*, 22(1):153–165, 1998.

[75] R.P. Woods, J.C. Mazziotta, and S.R. Cherry. MRI–PET Registration with Automated Algorithm. *Journal of Computer Assisted Tomography*, 17(4):536–546, July/August 1993.

[76] M.A. Yassa and C.E.L. Stark. A quantitative evaluation of cross-participant registration techniques for mri studies of the medial temporal lobe. *NeuroImage*, 44:319–327, 2009.