NONRIGID IMAGE REGISTRATION BY THE DEFORMATION BASED GRID GENERATION

by

CHIH-YAO HSIEH

Presented to the Faculty of the Graduate School of
The University of Texas at Arlington in Partial Fulfillment
of the Requirements
for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT ARLINGTON
December 2008

To my parents and my friends

who helped me and supported me to complete my dream.

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my supervising professor Dr. Hua-Mei Chen for his constantly motivating, invaluable guidance, and priceless advice during the course of my doctoral studies.

Special thanks goes to Dr. Guojun Liao and Dr. Roderick McColl for their priceless enlightening advices in the mathematics and radiology fields. I also want to thank my committee members: Dr. Roger Walker, Dr. Jean Gao, and Dr. Guojun Liao for their interest in my research and for taking time to serve in my dissertation committee.

I am pleased to express my gratitude to the Engineering School of the University of Texas of Arlington and the Computer Science Department for their financial and hardware support.

Finally, it is a great opportunity to express my sincere appreciation to my parents and Chih-Wei Yuan for their selfless support and encouragement. Thanks for Dr. Chang's revision of this dissertation and valuable suggestions. I also thank my lab colleagues, Ting-Hung Lin, Hsi-Yue Hsiao, and Mei-Yi Chu. for their support and generous sharing of knowledge.

November 21, 2008

ABSTRACT

NONRIGID IMAGE REGISTRATION BY THE DEFORMATION BASED GRID GENERATION

CHIH-YAO HSIEH, Ph.D.

The University of Texas at Arlington, 2008

Supervising Professor: Hua-Mei Chen

A novel nonrigid image registration algorithm is developed using a well-established mathematic work known as deformation based grid generation.

The deformation based grid generation is capable to generate a grid free of mesh folding, which is achieved by devising a positive monitor function describing the anticipated grid point density in the computational domain. Based on this method, a novel nonrigid image registration algorithm is successfully developed with many interesting features. First of all, the functional to be optimized during the image registration process consists of only one term — the similarity term. Thus, no regularization functional is required in this method, not to mention the weight to balance the regularization functional and the similarity functional commonly required in many nonrigid image registration methods. Nevertheless, the regularity (no mesh folding) of the resultant deformation vector field is theoretically guaranteed. Secondly, since no regularization term is introduced in the functional to be optimized, the resultant deformation vector field is highly flexible that large deformation frequently experienced in inter-patient or image-atlas registration tasks can be accurately estimated.

We present the detailed description of our proposed nonrigid image registration method with different implementations, alone with several 2D and 3D experimental results evaluating the registration quality, performance, and noise tolerance capability.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Image Registration

Image acquisition technologies for medical imaging have greatly advanced in the past several decades. Numerous choices of modalities of medical images are available for clinicians, physicians, and neuroscientists. Some modalities like computed tomography (CT), magnetic resonance imaging (MRI), diffusion tensor imaging (DTI), ultrasound, and X-ray provide anatomical information, while others like positron emission tomography (PET), single positron emission tomography (SPECT), and functional MRI (fMRI) provide information on metabolic processes.

Clinically, images taken from the same or different patients are often compared for various purposes. This kind of task requires alignment between images due to inherent image distortion, patient movement during the image acquisition process, and temporal differences of the image acquisition [1]. Therefore, the automatic image registration is desirable to fulfill the need to align those images and avoid tedious manual image registration process.

Image registration [2, 3, 4, 5] plays an essential role in various medical imaging applications, such as fusion of images from different modalities [6, 7], super-resolution in positron emission tomography (PET) imaging [8], visualizing diffusion tensor MR images (DTI) [9, 10], atlas based segmentation [11, 12], geometric correction of functional magnetic resonance imaging (fMRI) [13], pattern recognition [14], etc. The purpose of image registration in these applications is to establish the correspondence among the pixels/voxels of image pairs.

Many existing nonrigid image registration methods are based on physical models such as elastic model proposed by Briot [15], diffusion image registration [16], viscous fluid model proposed by Christensen [17], and curverture based image registration [18]. These models construct partial differential equations (PDEs) to simulate the deformation of physical objects (elastic solids, viscous fluid) under external forces. In nonrigid image registration, these external forces are derived from the similarity measures, such as sum of squared difference (SSD) or mutual information (MI). By solving the PDEs, the spatial transformations between an image pair can be obtained. In this type of work, image registration problem is solved using the variational formulation, i.e., minimization of the cost functional comprising two terms [19]: a similarity functional and a regularization functional. That is:

$$\textit{cost functional} = \textit{similarity functional} + \beta \times \textit{regularization functional.} \tag{1.1}$$

The similarity functional is designed to measure the similarity of two images subject to a given deformation field and the regularization functional is devised to manage the quality of the deformation to ensure that the generated deformation field is physically plausible — no grid folding. For this purpose, the regularization functional is usually designed to penalize non-smoothing deformation fields.

Obviously, the registration quality is affected by the choices of the regularization functional and the regularization parameter β. For the physically based models, the optimal positive regularization parameter β is application specific, and the most appropriate β can only be determined from experiments, which means that there is no convenient, universal, and unique optimal β for all types of applications. Moreover, different tissues represented in an image may have differently preferred β values. This is due to the varying physical properties of different tissues. If a lower than appropriate application specific β value is chosen, the resulting deformation field may suffer from grid folding. Therefore, it is desirable to develop a more universal nonrigid image registration scheme suitable for the general registration problems.

1.2 Grid Generation

Grid generation is an essential step to solve differential equations numerically. There are two types of grids [20] - structured and unstructured. A structured grid is defined as a grid with regular connectivity, which means that each vertex of the grid, excluding the boundary, has the same number of vertices in the vicinity. This restriction implies that each element should be quadrilateral for a 2D grid and hexahedra for a 3D grid. An unstructured grid is specified as a grid with irregular connectivity, that is, for each vertex of the grid, the number of neighboring vertices may be different.

Among the structured grid generation methods, adaptive grid generation is popularly used in scientific computing, such as fluid flow field simulation and fluid-structure interaction, to accommodate a complicate domain [21, 22]. The adaptive grid generation was introduced as an iterative mesh regeneration approach to accommodate complex domain and obtain a more accurate solution. However, the adaptive grid generation may introduce the "mesh tangling" [23, 24], which is undesirable.

In 1992, Liao and Anderson purposed a deformation based adaptive grid generation method [25, 26, 27], which can avoid the grid folding problem by enforcing the Jacobian determinant of the generated grid to be strictly positive through the control of a monitor function. The details of this method is reviewed in Chap. 2.

1.3 Motivation

Considering the theoretically guaranteed no grid folding feature of the deformation based adaptive grid generation methods [25, 26, 27], it is natural to investigate the possibility of applying them to nonrigid image registration problems, which motivates this work. In this dissertation, we developed a new nonrigid image registration approach based on the deformation based adaptive grid generation method.

The major difference between this work and the physically based nonrigid image registration methods is that there is only one term, the similarity functional, involved in the cost functional to be optimized, while the regularity of the deformation field is guaranteed.

This work is formulated as a constrained parametric optimization problem

$$\boldsymbol{f} = \arg\{\text{optimize}_{\boldsymbol{f}}\, c(\boldsymbol{u} \mid \boldsymbol{f}) \mid \boldsymbol{f} \text{ is subject to certain constraints}\}$$

where c is the cost functional, $\boldsymbol{u}$ represents the deformation field, and $\boldsymbol{f}$ is a set of the constrained parameters. The parameters are associated with a set of control points defined as the grid points of a coarse grid. The enforced constraint guarantees no grid folding will occur in the resulting grid.

This dissertation is organized as follows. Chapter two reviews the deformation based grid generation method. Chapter three is the proposed nonrigid image registration work. Chapter four covers the implementation issues. Chapter five are experiment results and analysis. Chapter six is the conclusion and future works.

CHAPTER 2

DEFORMATION BASED GRID GENERATION

Numerical grid generation is a common tool for numerical methods that employ finite differences, finite volumes, or finite elements to solve partial differential equations (PDEs) on arbitrarily shaped domains. It is a process of discretizing the solution field/physical domain into small elements or cells and acquire the solutions at each of the grid points. The grids generated for solving PDEs are crucial for the accuracy of the numerical solution of PDEs. Detailed information about grid generation is available in [28, 29].

Among those numerous grid generation methods, we are intrigued by a structured mesh generation method known as the deformation based adaptive grid generation proposed by Liao and Anderson in 1992 [25, 26, 27], where the structured mesh is defined as all elements and nodes have the same topology (same number of neighbors). This approach is able to construct a desired structured grid by generating an one to one and onto mapping function $\boldsymbol{\phi}$ that maps the coordinates of the original grid points to the new grid points.

Adaptive grid generation was introduced as an iterative mesh regeneration approach to accommodate complex domain and obtain a more accurate solution. In general, adaptive grid generation may introduce “mesh tangling” and thus are not physically plausible [23, 24]. However, the deformation based adaptive grid generation considered in this work suffers no such concern. In [25], this is achieved by enforcing the Jacobian determinant of the deformation field to be strictly positive where the Jacobian determinant is specified on the original grid before adaptation.

In 1994, Liao et al. [30] improved the deformation method so that the Jacobian determinant can be specified on the grid after adaptation. Later, in [31] and [32], they developed a further improved deformation method with real time adaptation. The three versions of the deformation based grid generation methods are described in the following sections.

2.1 Deformation Method: Version 1

This is one of the steady versions of deformation method where the transformation Jacobian determinant is specified on the old grid $\boldsymbol{\xi}$ before adaptation [25]. Problem: Given a monitor function $f(\boldsymbol{\xi})$, normalized by

$$\int_{\Omega} f = |\Omega|, \tag{2.1}$$

where $|\Omega|$ is the volume of the solution domain Ω. Find a mapping function $\boldsymbol{\phi}_1 : \Omega \rightarrow \Omega$, $\partial\Omega \rightarrow \partial\Omega$ such that

$$J(\boldsymbol{\phi}_1(\boldsymbol{\xi})) = \det \nabla \boldsymbol{\phi}_1(\boldsymbol{\xi}) = f(\boldsymbol{\xi}). \tag{2.2}$$

The following two steps are used to find such a mapping.

Step 1: Find a intermediate vector field $\boldsymbol{\eta}(\boldsymbol{\xi})$ that satisfies:

$$\begin{cases} \nabla \cdot \boldsymbol{\eta}(\boldsymbol{\xi}) = f(\boldsymbol{\xi}) - 1 \\ \nabla \times \boldsymbol{\eta}(\boldsymbol{\xi}) = 0 \end{cases}, \quad \boldsymbol{\xi} \in \Omega \tag{2.3}$$

and

$$\boldsymbol{\eta}(\boldsymbol{\xi}) \cdot \hat{n} = 0, \quad \boldsymbol{\xi} \in \partial\Omega. \tag{2.4}$$

Step 2: Form a time dependent velocity vector field from $\boldsymbol{\eta}(\boldsymbol{\xi})$,

$$\boldsymbol{V}(\boldsymbol{\xi}, t) = \frac{\boldsymbol{\eta}(\boldsymbol{\xi})}{t + (1-t)f(\boldsymbol{\xi})}, \quad \boldsymbol{\xi} \in \Omega, \tag{2.5}$$

Step 3: Find $\boldsymbol{\phi}_t(\boldsymbol{\xi}, t)$ by solving the following ordinary differential equation (ODE)

$$\frac{d\boldsymbol{\phi}(\boldsymbol{\xi}, t)}{dt} = \boldsymbol{V}_t(\boldsymbol{\phi}(\boldsymbol{\xi}, t)), \quad t \in [0, 1], \tag{2.6}$$

where $\boldsymbol{\xi} \in \Omega$. Denote $\boldsymbol{\phi}_t(\boldsymbol{\xi}) = \boldsymbol{\phi}(\boldsymbol{\xi}, t)$, then the mapping $\boldsymbol{\phi}_1(\boldsymbol{\xi}) = \boldsymbol{\phi}(\boldsymbol{\xi}, t = 1)$. The proof of the above approach can be found in [33].

This version provides an effective practice to generate a well regularized deformation field by imposing constraints on the monitor function f.

Notice that the curl of $\boldsymbol{\eta}$ is restricted to be 0. This restriction was originally imposed so that the div-curl system can be transformed into a single Poisson's equation that can be solved easily using the finite difference method.

2.2 Deformation Method: Version 2

This is another static version of the deformation method where the transformation Jacobian determinant is specified on the new grid $\boldsymbol{\phi}(\boldsymbol{\xi})$ after adaptation [26].

Problem: Given a monitor function f normalized by

$$\int_\Omega \frac{1}{f} = |\Omega|, \tag{2.7}$$

find a mapping function $\boldsymbol{\phi}_1 : \Omega \to \Omega$, $\partial\Omega \to \partial\Omega$ such that

$$J(\boldsymbol{\phi}_1(\boldsymbol{\xi})) = f(\boldsymbol{\phi}_1(\boldsymbol{\xi})), \quad \boldsymbol{\xi} \in \Omega. \tag{2.8}$$

To find such a transformation, the following steps are proposed in [26].

Step 1: Compute $\boldsymbol{\eta}(\boldsymbol{\xi})$ such that

$$\begin{cases} \nabla \cdot \boldsymbol{\eta}(\boldsymbol{\xi}) = 1 - \dfrac{1}{f(\boldsymbol{\xi})} \\ \nabla \times \boldsymbol{\eta}(\boldsymbol{\xi}) = 0 \end{cases}, \quad \boldsymbol{\xi} \in \Omega, \tag{2.9}$$

and

$$\boldsymbol{\eta}(\boldsymbol{\xi}) \cdot \hat{n} = 0, \quad \boldsymbol{\xi} \in \partial\Omega. \tag{2.10}$$

Step 2: For each fixed node $\boldsymbol{\xi}$, $\boldsymbol{\xi} \in \Omega$, solve the ODE

$$\frac{\partial \boldsymbol{\phi}(\boldsymbol{\xi}, t)}{\partial t} = \boldsymbol{V}(\boldsymbol{\phi}(\boldsymbol{\xi}, t), t), \quad t \in [0, 1] \tag{2.11}$$

with $\boldsymbol{\phi}(\boldsymbol{\xi},\, 0) = \boldsymbol{\xi}$, where

$$\boldsymbol{V}(\boldsymbol{\phi}(\boldsymbol{\xi}, t), t) = \frac{\boldsymbol{\eta}(\boldsymbol{\phi}(\boldsymbol{\xi}, t))}{t \dfrac{1}{f(\boldsymbol{\phi}(\boldsymbol{\xi}, t))} + (1 - t)}. \tag{2.12}$$

Define $\boldsymbol{\phi}_t(\boldsymbol{\xi}) = \boldsymbol{\phi}(\boldsymbol{\xi}, t)$, then $\boldsymbol{\phi}_1(\boldsymbol{\xi}) = \boldsymbol{\phi}(\boldsymbol{\xi}, 1)$ is the solution.

The proof of the above approach is very similar to the proof of the first version. Detailed proof can be found in [33].

2.3 Deformation Method: Version 3

This is the version of deformation method with real time adaptation [27].

Problem: Given a time-varing monitor function $f(\boldsymbol{\xi}, t) > 0$, normalized with

$$\int_{\Omega(t)} \frac{1}{f(\omega, t)}\, d\omega = |\Omega(t = 0)|, \tag{2.13}$$

find a mapping $\boldsymbol{\phi}(\cdot, t) : \Omega(t=0) \rightarrow \Omega(t)$ such that:

$$J(\boldsymbol{\phi}(\boldsymbol{\xi}, t)) = f(\boldsymbol{\phi}(\boldsymbol{\xi}, t), t), \quad t > 0 \tag{2.14}$$

(assuming (2.14) is true at $t = 0$) where $J(\boldsymbol{\phi}(\boldsymbol{\xi}, t)) = \det \nabla \boldsymbol{\phi}(\boldsymbol{\xi}, t)$ is the Jacobian determinant of the transformation, $\forall \boldsymbol{\xi} \in \partial\Omega(t = 0)$, and $\boldsymbol{\phi}(\boldsymbol{\xi}, t) \in \partial\Omega(t)$. The transformation $\boldsymbol{\phi}$ can be found by the following two steps.

Step 1: Find a vector field $\boldsymbol{\eta}(\boldsymbol{\xi}, t)$ such that:

$$\begin{cases} \nabla \cdot \boldsymbol{\eta}(\boldsymbol{\xi}, t) = -\dfrac{\partial}{\partial t}\dfrac{1}{f(\boldsymbol{\xi}, t)} \\ \nabla \times \boldsymbol{\eta}(\boldsymbol{\xi}, t) = 0 \end{cases}, \quad \boldsymbol{\xi} \in \Omega \tag{2.15}$$

and

$$\boldsymbol{\eta}(\boldsymbol{\xi}, t) \cdot \hat{n} = 0, \quad \forall \boldsymbol{\xi} \in \partial\Omega(t). \tag{2.16}$$

Step 2: Solve the ODE (2.17) for the transformation $\boldsymbol{\phi}(\boldsymbol{\xi},\ t)$,

$$\frac{\partial \boldsymbol{\phi}(\boldsymbol{\xi}, t)}{\partial t} = V(\boldsymbol{\phi}(\boldsymbol{\xi}, t), t) = f(\boldsymbol{\phi}(\boldsymbol{\xi}, t), t)\boldsymbol{\eta}(\boldsymbol{\phi}(\boldsymbol{\xi}, t), t) \tag{2.17}$$

We can show that the $\boldsymbol{\phi}(\boldsymbol{\xi}, t)$ found by this way satisfies equation (2.14) [33].

2.4 Examples of the Deformation Based Grid Generation

This section demonstrates the example grids generated by the deformation based grid generation version 1, 2, and 3.

Given a set of monitor function values assigned on a 7 × 7 grid as Fig. 2.1a shows, by applying the deformation method based grid generation version 1 and 2, we can get the grids generated in Fig. 2.1b and Fig. 2.1c respectively.

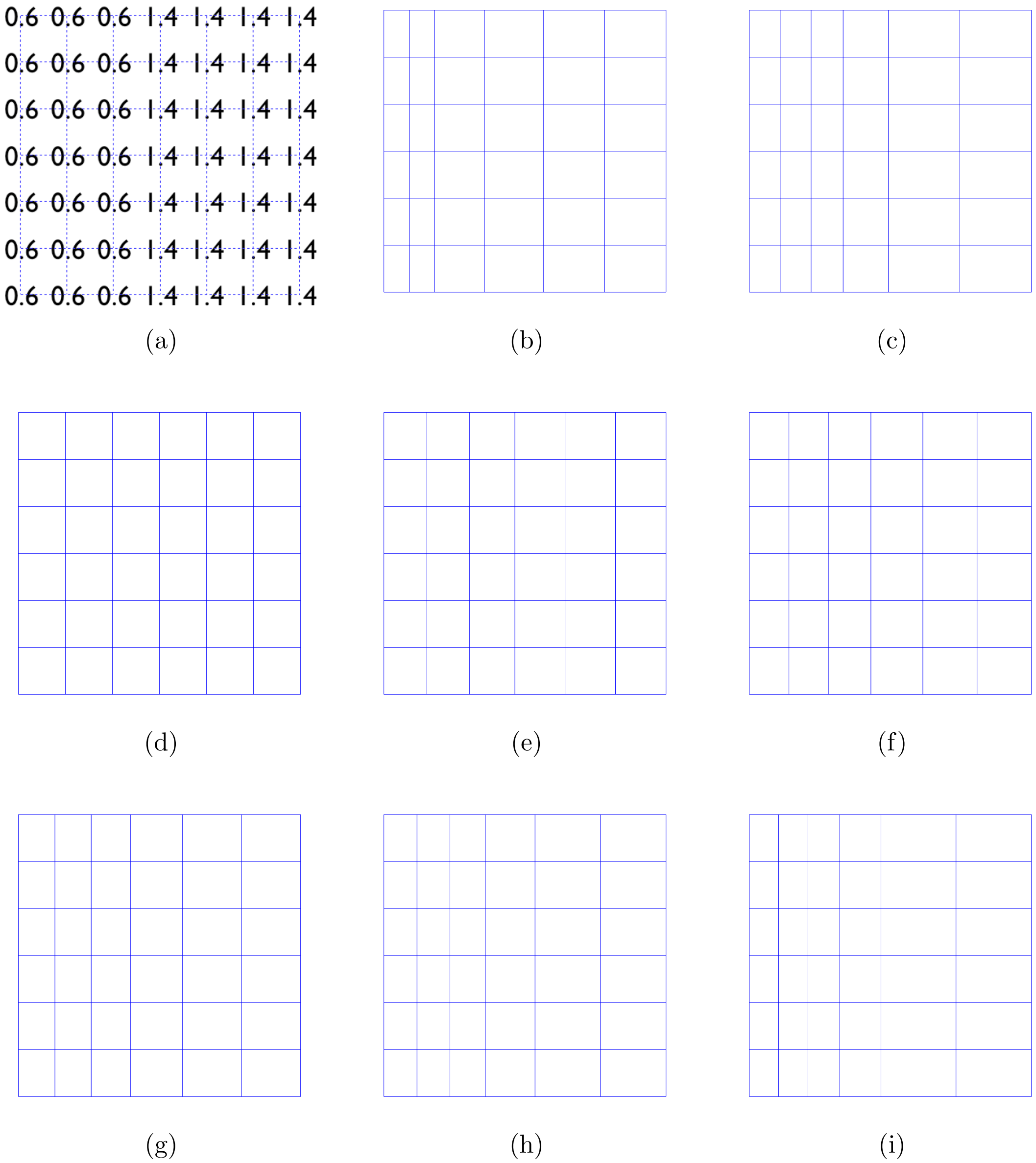


Figure 2.1. (a) The monitor function values assigned on the original grid points to generate the deformed grids shown from (b) to (i). (b) Grid generated by the deformation based grid generation version one. (c) Grid generated by the deformation based grid generation version 2. (d) to (i) shows the grid generated by the deformation based grid generation version 3 at time 0, 0.2, 0.4, 0.6, 0.8, and 1, respectively.

For version 3, given a set of monitor function values $f_1(\boldsymbol{\xi})$ assigned on a 7×7 grid as Fig. 2.1a shows, where $f_1(\boldsymbol{\xi}) = f(\boldsymbol{\xi}, t = 1)$. Let

$$f(\boldsymbol{\xi}, t) = (1 - t) + t f_1(\boldsymbol{\xi}), \tag{2.18}$$

then

$$\frac{d}{dt}\frac{1}{f(\boldsymbol{\xi},t)} = \frac{f_1(\boldsymbol{\xi})-1}{((1-t)+tf_1(\boldsymbol{\xi}))^2}. \tag{2.19}$$

We can get the series of generated grid as Fig. 2.1d to Fig. 2.1i shows, where $J(\boldsymbol{\phi}(\boldsymbol{\xi},t)) = f(\boldsymbol{\phi}(\boldsymbol{\xi},t),t)$.

CHAPTER 3

PROPOSED NONRIGID IMAGE REGISTRATION BY THE DEFORMATION METHOD

Based on the well-established deformation based grid generation described in Chapter 2, a new formulation for nonrigid image registration is proposed in this dissertation. It is formulated as a constrained parametric optimization problem involving the similarity functional only, thus eliminating the application specific β in Eq. (1.1). The idea can be expressed mathematically as the following:

$$\boldsymbol{f} = \arg\{\text{optimize}_{\boldsymbol{f}}\, c_{R,T}(\boldsymbol{u} \,|\, \boldsymbol{f}) \,|\, \boldsymbol{f} \text{ is subject to certain constraints}\}$$

where c is the cost functional to be optimized, the subscripts R and T represent the two images to be registered, $\boldsymbol{u}$ is the deformation field wholly determined by a set of parameters $\boldsymbol{f}$, which is subject to certain constraints. The parameters are associated with a set of control points defined as the grid points of a coarse grid. The enforced constraints guarantee that no grid folding will occur in the resulting grid.

These parameters are not explicitly related to the deformation field, but related to the divergence and curl of an intermediate vector field from which the deformation field can be obtained. The parameters associated with each control point are the monitor function values and the curl values of the intermediate vector field, where the monitor function is defined directly related to the divergence of the intermediate vector field. The regularization requirement of the deformation field is fulfilled by restricting the monitor function to be strictly positive since the monitor function is equivalent to the Jacobian determinant of the deformation field.

3.1 Problem Formulation

Having two d-dimensional discrete images $R(\boldsymbol{\xi})$ and $T(\boldsymbol{\xi})$, where $d = 2$ or 3, R is referred to as a reference image, T is a test image, and $\boldsymbol{\xi} \in I \subset \mathbb{Z}^d$ is the coordinate of a point in the discrete image domain I. The goal is to register the test image T to the reference image R.

To properly register T to R, we need to find a permissible deformation field $\boldsymbol{\phi} : I \to I^c$, where I^c is the continuous image domain, such that the deformed test image $T^{\boldsymbol{\phi}}(\boldsymbol{\xi}) \equiv T^c(\boldsymbol{\phi}(\boldsymbol{\xi}))$ is geometrically aligned with the reference image $R(\boldsymbol{\xi})$, where $T^c : I^c \subset \mathbb{R}$ is the continuous version of the test image.

Next, we discuss the proposed nonrigid image registration method by the deformation method in three versions separately.

3.2 Version 1

3.2.1 Permissible Set

For version 1, we define a *permissible* set as the collection of deformation fields generated by the following steps.

1. Generate an arbitrary positive scalar monitor function f and an arbitrary scalar g (for 2D case) or vector (for 3D case) curl $\boldsymbol{g}$ on a discrete image domain I, where f is normalized by

$$\int_\Omega f = |\Omega|, \tag{3.1}$$

   and $\boldsymbol{g}$ consists of three components g^1, g^2, and g^3 in 3D case. Theoretically, $\boldsymbol{g}$ in 3D case must satisfy the solvability constraint [34] to ensure the existence of the solution of Eq. (3.2), which will be discussed in Sec. 4.9. This constraint is not considered for the current context.

2. Solve the following div-curl system for an intermediate vector field $\boldsymbol{\eta}(\boldsymbol{\xi})$ using the Dirichlet boundary condition.

$$\begin{cases} \nabla \cdot \boldsymbol{\eta}(\boldsymbol{\xi}) = f(\boldsymbol{\xi}) - 1 \\ \nabla \times \boldsymbol{\eta}(\boldsymbol{\xi}) = \boldsymbol{g}(\boldsymbol{\xi}) \end{cases} \tag{3.2}$$

3. Generate a time-dependent velocity vector field $\boldsymbol{V}$ from $\boldsymbol{\eta}$ and f

$$\boldsymbol{V}_t(\boldsymbol{\xi}) = \boldsymbol{V}(\boldsymbol{\xi}, t) = \frac{\boldsymbol{\eta}(\boldsymbol{\xi})}{t + (1-t)f(\boldsymbol{\xi})}. \tag{3.3}$$

4. Find the deformation field $\boldsymbol{\phi}_1(\boldsymbol{\xi})$ by solving the following ODE

$$\frac{d\boldsymbol{\phi}_t(\boldsymbol{\xi})}{dt} = \boldsymbol{V}_t(\boldsymbol{\phi}_t(\boldsymbol{\xi})), \quad t \in [0, 1], \tag{3.4}$$

where $\boldsymbol{\phi}_1(\boldsymbol{\xi}) = \boldsymbol{\phi}(\boldsymbol{\xi}, 1)$ and $\boldsymbol{\phi}_t(\boldsymbol{\xi}) = \boldsymbol{\phi}(\boldsymbol{\xi}, t)$.

The deformation field $\boldsymbol{\phi}$ generated by this procedure is called a *permissible* transformation because it possesses the property that the Jacobian determinant of $\boldsymbol{\phi}_1(\boldsymbol{\xi})$ is equal to the specified strictly positive monitor function $f(\boldsymbol{\xi})$. This can be justified by the fact that the proof of $\boldsymbol{\phi}_1(\boldsymbol{\xi}) = f(\boldsymbol{\xi})$ in the deformation based grid generation version 1 [25, 33] is independent of the curl of the intermediate field $\boldsymbol{\eta}$. Since the modification of curl $\boldsymbol{g}$ to be nonzero is the only modification we made, as long as the monitor function f is restricted to be strictly positive, the resulting deformation field $\boldsymbol{\phi}$ suffers no mesh folding and thus is physically plausible (permissible).

3.2.2 Major Components

3.2.2.1 A Div-Curl Solver by LSFEM

To solve the intermediate vector field $\boldsymbol{\eta}(\boldsymbol{\xi})$ in Eq. (3.2), a numerical div-curl solver is required. In this work, a div-curl solver using the LSFEM [27] is adopted and described below.

Let

$$\mathbf{A}\boldsymbol{\eta} = \boldsymbol{f} \tag{3.5}$$

denote the div-curl system from Eq. (3.2) with the Dirichilet boundary condition.

For a 2D case,

$$\mathbf{A} = \begin{pmatrix} \frac{\partial}{\partial x} & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & -\frac{\partial}{\partial x} \end{pmatrix}, \boldsymbol{\eta} = \begin{pmatrix} \eta^x \\ \eta^y \end{pmatrix}, \text{ and } \boldsymbol{f} = \begin{pmatrix} f^1 \\ f^2 \end{pmatrix} \tag{3.6}$$

, where $f^1 = f - 1$ (the monitor function $-$ 1) and $f^2 = g$ (the curl value).

As for a 3D case,

$$\mathbf{A} = \begin{pmatrix} \frac{\partial}{\partial x} & \frac{\partial}{\partial y} & \frac{\partial}{\partial z} \\ 0 & -\frac{\partial}{\partial z} & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} & 0 & -\frac{\partial}{\partial x} \\ -\frac{\partial}{\partial y} & \frac{\partial}{\partial x} & 0 \end{pmatrix}, \boldsymbol{\eta} = \begin{pmatrix} \eta^x \\ \eta^y \\ \eta^z \end{pmatrix}, \text{ and } \boldsymbol{f} = \begin{pmatrix} f^1 \\ f^2 \\ f^3 \\ f^4 \end{pmatrix} \tag{3.7}$$

where $f^1 = f - 1$ (the monitor function $-$ 1) and $[f^2, f^3, f^4]^T = [g^1, g^2, g^3]^T$ (the three component curl values).

For an arbitrary test solution v, the residual is defined as $\mathbf{A}\boldsymbol{v} - \boldsymbol{f}$. The LSFEM method is based on the minimization of the residual in a least-square sense. That is, the solution of Eq. (3.2) is the one that minimizes the L_2 distance $d(\boldsymbol{v})$ between $\mathbf{A}\boldsymbol{v}$ and $\boldsymbol{f}$, where

$$d(\boldsymbol{v}) = \left\|\mathbf{A}\boldsymbol{v} - \boldsymbol{f}\right\|^2 = \langle \mathbf{A}\boldsymbol{v} - \boldsymbol{f}, \mathbf{A}\boldsymbol{v} - \boldsymbol{f} \rangle \,. \tag{3.8}$$

To minimize the distance between $\mathbf{A}\boldsymbol{v}$ and $\boldsymbol{f}$ at solution $\boldsymbol{\eta}$, the following condition must be met:

$$\begin{aligned}
\lim_{t\to 0} \frac{d}{dt} d(\boldsymbol{\eta} + t\boldsymbol{v}) &= \int_\Omega \lim_{t\to 0} \frac{d}{dt} \langle \mathbf{A}(\boldsymbol{\eta} + t\boldsymbol{v}) - \boldsymbol{f}, \mathbf{A}(\boldsymbol{\eta} + t\boldsymbol{v}) - \boldsymbol{f} \rangle \; d\omega \\
&= 2 \int_\Omega \langle \mathbf{A}\boldsymbol{v}, \mathbf{A}\boldsymbol{\eta} - \boldsymbol{f} \rangle \; d\omega \qquad (3.9) \\
&= 0.
\end{aligned}$$

Therefore,

$$\int_\Omega \mathbf{A}\boldsymbol{\eta} \cdot \mathbf{A}\boldsymbol{v} \, d\omega = \int_\Omega \boldsymbol{f} \cdot \mathbf{A}\boldsymbol{v} \, d\omega. \tag{3.10}$$

After segmenting the domain Ω into a union of finite elements, construct $\boldsymbol{\eta}$ and $\boldsymbol{f}$ in each element from their nodal values using an appropriate shape function ψ.

For example in 2D case

$$\boldsymbol{\eta}^e(\boldsymbol{x}) = \sum_{j=1}^{N_e} \psi_j(\boldsymbol{x}) \begin{pmatrix} \eta^1(I_j) \\ \eta^2(I_j) \end{pmatrix} \qquad \boldsymbol{x} \subset \Omega_e, \tag{3.11}$$

$$\boldsymbol{f}^e(\boldsymbol{x}) = \sum_{j=1}^{N_e} \psi_j(\boldsymbol{x}) \begin{pmatrix} f^1(I_j) \\ f^2(I_j) \end{pmatrix} \qquad \boldsymbol{x} \subset \Omega_e, \tag{3.12}$$

where N_e is the number of nodes in each element, ψ_j is the shape function at node I_j, $(\eta^1(I_j), \eta^2(I_j))^T$ are the solutions at node I_j, $f^1(I_j)$ is the div value at node I_j, $f^2(I_j)$

is the curl value at node I_j, and Ω_e is the element domain. By substituting $\boldsymbol{v}$ with a shape function at each node, we can obtain a linear system of algebraic equations

$$\boldsymbol{K}^{\eta}\boldsymbol{\eta}(\vec{I}) = \boldsymbol{K}^{f}\boldsymbol{f}(\vec{I}) \tag{3.13}$$

where $\vec{I}$ represents the column vector formed from grid I and $\boldsymbol{\eta}(\vec{I})$ means that the solutions at all nodes are rearranged into a column vector form. Similarly, $\boldsymbol{f}(\vec{I})$ is a column vector consisting of the given divergence and curl values at all nodes. $\boldsymbol{K}^{\eta}$ and $\boldsymbol{K}^{f}$ are assembled from element matrices $\boldsymbol{K}_e^{\eta}$ and $\boldsymbol{K}_e^{f}$:

$$\boldsymbol{K}_e^{\eta} = \int_{\Omega_e} (\mathbf{A}\psi_1, \mathbf{A}\psi_2, \ldots, \mathbf{A}\psi_{N_e})^T (\mathbf{A}\psi_1, \mathbf{A}\psi_2, \ldots, \mathbf{A}\psi_{N_e})\, d\omega \tag{3.14}$$

$$\boldsymbol{K}_e^{f} = \int_{\Omega_e} (\mathbf{A}\psi_1, \mathbf{A}\psi_2, \ldots, \mathbf{A}\psi_{N_e})^T (\psi_1, \psi_2, \ldots, \psi_{N_e})\, d\omega \tag{3.15}$$

Applying the Dirichlet boundary condition, Eq. (3.13) becomes

$$\tilde{\boldsymbol{K}}^{\eta}\tilde{\boldsymbol{\eta}}(\vec{I}) = \tilde{\boldsymbol{K}}^{f}\tilde{\boldsymbol{f}}(\vec{I}) \tag{3.16}$$

where $\tilde{\boldsymbol{\eta}}(\vec{I})$ and $\tilde{\boldsymbol{f}}(\vec{I})$ are the values at internal grid points. The LSFEM solution can be obtained through the inverse of the matrix $\tilde{\boldsymbol{K}}^{\eta}$:

$$\tilde{\boldsymbol{\eta}}(\vec{I}) = (\tilde{\boldsymbol{K}}^{\eta})^{-1}\tilde{\boldsymbol{K}}^{f}\tilde{\boldsymbol{f}}(\vec{I}) \tag{3.17}$$

However, the size of $\boldsymbol{K}^{\eta}$ soon becomes unmanageable. For a 2D image of size 64×64, the size of $\boldsymbol{K}^{\eta}$ is as large as 8192×8192. Though an iterative element by element approach is available [34], it is still not efficient. Next, we present an approximate solution using the inverse filtering which can overcome this unmanageable size problem.

3.2.2.2 The Proposed LSFEM Div-curl Solver Using Inverse Filtering

Properly arranging the orders of $\tilde{\boldsymbol{\eta}}(\vec{I})$ and $\tilde{\boldsymbol{f}}(\vec{I})$, we can transform the linear system (3.16) into the following format for 2D case,

$$\begin{pmatrix} \tilde{\boldsymbol{K}}_{11}^{\eta} & 0 \\ 0 & \tilde{\boldsymbol{K}}_{22}^{\eta} \end{pmatrix} \begin{pmatrix} \tilde{\boldsymbol{\eta}}^{x}(\vec{I}) \\ \tilde{\boldsymbol{\eta}}^{y}(\vec{I}) \end{pmatrix} = \begin{pmatrix} \tilde{\boldsymbol{K}}_{11}^{f} & \tilde{\boldsymbol{K}}_{12}^{f} \\ \tilde{\boldsymbol{K}}_{21}^{f} & \tilde{\boldsymbol{K}}_{22}^{f} \end{pmatrix} \begin{pmatrix} \tilde{\boldsymbol{f}}^{f_1}(\vec{I}) \\ \tilde{\boldsymbol{f}}^{f_2}(\vec{I}) \end{pmatrix} \tag{3.18}$$

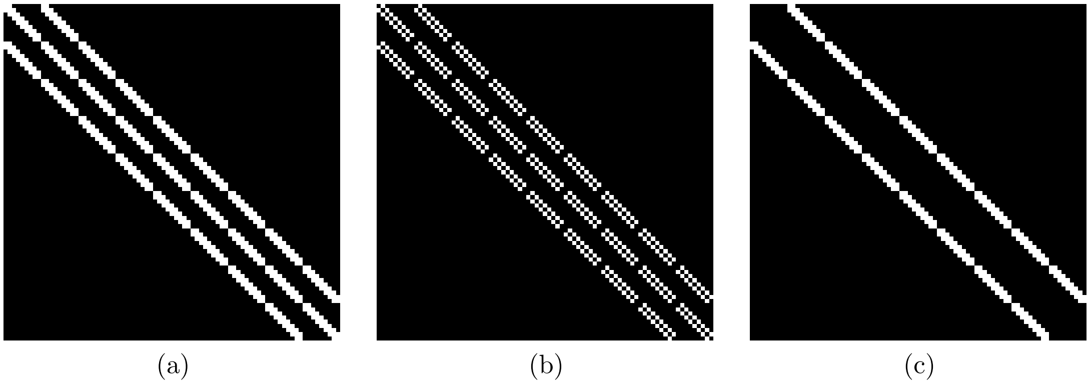

(a) (b) (c)

Figure 3.1. Non-zero pattern of the blocks of $\tilde{\boldsymbol{K}}^{\eta}$ and $\tilde{\boldsymbol{K}}^{f}$. (a) $\tilde{\boldsymbol{K}}_{11}^{\eta}$ and $\tilde{\boldsymbol{K}}_{22}^{\eta}$, (b) $\tilde{\boldsymbol{K}}_{11}^{f}$ and $\tilde{\boldsymbol{K}}_{22}^{f}$, and (c) $\tilde{\boldsymbol{K}}_{12}^{f}$ and $\tilde{\boldsymbol{K}}_{21}^{f}$.

The non-zero patterns of the blocks of the matrices $\tilde{\boldsymbol{K}}^{\eta}$ and $\tilde{\boldsymbol{K}}^{f}$ are band-diagonal as Figure 3.1 shows. This implies the following approximation using the convolution like operator $\otimes_t$ —

$$\begin{aligned} \boldsymbol{m}_{11}^{\eta} \otimes_t \tilde{\boldsymbol{\eta}}^{x}(I) &= \boldsymbol{m}_{11}^{f} \otimes_t \tilde{\boldsymbol{f}}^{f_1}(I) + \boldsymbol{m}_{12}^{f} \otimes_t \tilde{\boldsymbol{f}}^{f_2}(I) \equiv \tilde{\boldsymbol{F}}^{1}(I) \\ \boldsymbol{m}_{22}^{\eta} \otimes_t \tilde{\boldsymbol{\eta}}^{y}(I) &= \boldsymbol{m}_{21}^{f} \otimes_t \tilde{\boldsymbol{f}}^{f_1}(I) + \boldsymbol{m}_{22}^{f} \otimes_t \tilde{\boldsymbol{f}}^{f_2}(I) \equiv \tilde{\boldsymbol{F}}^{2}(I), \end{aligned} \tag{3.19}$$

where the notation $\tilde{\boldsymbol{\eta}}^{x}(I)$ represents the values of $\tilde{\boldsymbol{\eta}}^{x}$ on grid I and $\otimes_t$ is introduced here as a truncated convolution operator. For example, for an $M \times N$ data A and an

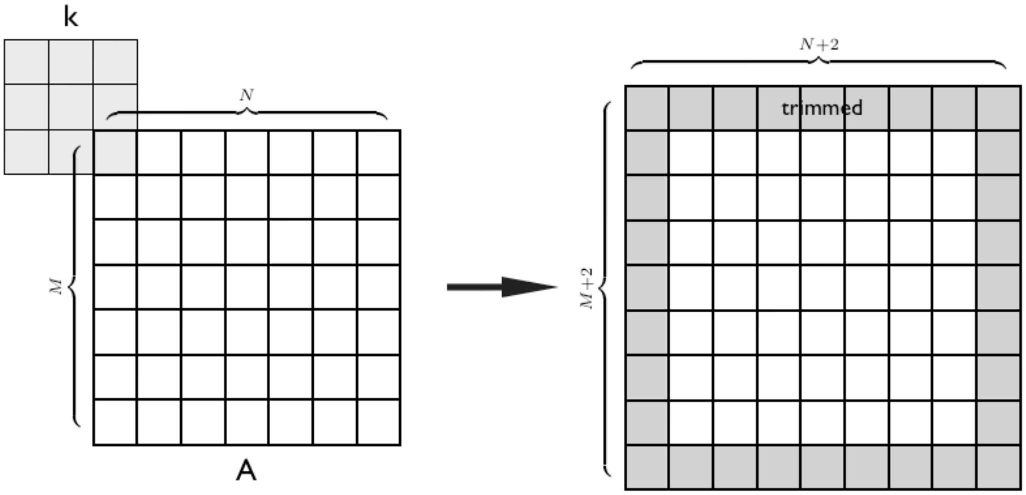


Figure 3.2. This figure shows how trimmed convolution operator works for $k \otimes_t A$.

3×3 convolution kernel k, the returned size of $k \otimes_t A$ is $M \times N$ as Fig. 3.2 shows. The filters in Eq. (3.19) are given below:

$$\boldsymbol{m}_{11}^{\eta} = \boldsymbol{m}_{22}^{\eta} = \begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix} \tag{3.20}$$

$$\boldsymbol{m}_{11}^{f} = \boldsymbol{m}_{22}^{f} = {\boldsymbol{m}_{21}^{f}}^{T} = -{\boldsymbol{m}_{12}^{f}}^{T} = \begin{pmatrix} 1 & 4 & 1 \\ 0 & 0 & 0 \\ -1 & -4 & -1 \end{pmatrix} \tag{3.21}$$

Thus, if $\otimes_t$ has the associative property, then the solution can be obtained using the inverse filtering as

$$\begin{aligned} \tilde{\boldsymbol{\eta}}^{x}(I) &= (\boldsymbol{m}_{11}^{\eta})^{-1} \otimes_t \tilde{\boldsymbol{F}}^{1}(I) \\ \tilde{\boldsymbol{\eta}}^{y}(I) &= (\boldsymbol{m}_{22}^{\eta})^{-1} \otimes_t \tilde{\boldsymbol{F}}^{2}(I). \end{aligned} \tag{3.22}$$

However, there is no associative property for $\otimes_t$, therefore, Eq. (3.22) can only be considered as an approximated solution.

In 3D case, after properly arranging the orders of $\tilde{\boldsymbol{\eta}}(\vec{I})$ and $\tilde{\boldsymbol{f}}(\vec{I})$, we can rewrite the Eq. (3.16) in the following form:

$$\begin{pmatrix} \tilde{\boldsymbol{K}}^{\eta}_{11} & 0 & 0 \\ 0 & \tilde{\boldsymbol{K}}^{\eta}_{22} & 0 \\ 0 & 0 & \tilde{\boldsymbol{K}}^{\eta}_{33} \end{pmatrix} \begin{pmatrix} \tilde{\boldsymbol{\eta}}^{x}(\vec{I}) \\ \tilde{\boldsymbol{\eta}}^{y}(\vec{I}) \\ \tilde{\boldsymbol{\eta}}^{z}(\vec{I}) \end{pmatrix} = \begin{pmatrix} \tilde{\boldsymbol{K}}^{f}_{11} & 0 & \tilde{\boldsymbol{K}}^{f}_{13} & \tilde{\boldsymbol{K}}^{f}_{14} \\ \tilde{\boldsymbol{K}}^{f}_{21} & \tilde{\boldsymbol{K}}^{f}_{22} & 0 & \tilde{\boldsymbol{K}}^{f}_{24} \\ \tilde{\boldsymbol{K}}^{f}_{31} & \tilde{\boldsymbol{K}}^{f}_{32} & \tilde{\boldsymbol{K}}^{f}_{33} & 0 \end{pmatrix} \begin{pmatrix} \tilde{\boldsymbol{f}}^{f_1}(\vec{I}) \\ \tilde{\boldsymbol{f}}^{f_2}(\vec{I}) \\ \tilde{\boldsymbol{f}}^{f_3}(\vec{I}) \\ \tilde{\boldsymbol{f}}^{f_4}(\vec{I}) \end{pmatrix} \tag{3.23}$$

This implies the following convolution form of equation (3.23):

$$\begin{aligned} \boldsymbol{m}^{\eta}_{11} \otimes_t \tilde{\boldsymbol{\eta}}^{x}(I) &= \boldsymbol{m}^{f}_{11} \otimes_t \tilde{\boldsymbol{f}}^{f^1}(I) + \boldsymbol{m}^{f}_{13} \otimes_t \tilde{\boldsymbol{f}}^{f^3}(I) + \boldsymbol{m}^{f}_{14} \otimes_t \tilde{\boldsymbol{f}}^{f^4}(I) \equiv \tilde{\boldsymbol{F}}^{1}(I) \\ \boldsymbol{m}^{\eta}_{22} \otimes_t \tilde{\boldsymbol{\eta}}^{y}(I) &= \boldsymbol{m}^{f}_{21} \otimes_t \tilde{\boldsymbol{f}}^{f^1}(I) + \boldsymbol{m}^{f}_{22} \otimes_t \tilde{\boldsymbol{f}}^{f^2}(I) + \boldsymbol{m}^{f}_{24} \otimes_t \tilde{\boldsymbol{f}}^{f^4}(I) \equiv \tilde{\boldsymbol{F}}^{2}(I) \\ \boldsymbol{m}^{\eta}_{33} \otimes_t \tilde{\boldsymbol{\eta}}^{z}(I) &= \boldsymbol{m}^{f}_{31} \otimes_t \tilde{\boldsymbol{f}}^{f^1}(I) + \boldsymbol{m}^{f}_{32} \otimes_t \tilde{\boldsymbol{f}}^{f^2}(I) + \boldsymbol{m}^{f}_{33} \otimes_t \tilde{\boldsymbol{f}}^{f^3}(I) \equiv \tilde{\boldsymbol{F}}^{3}(I) \end{aligned} \tag{3.24}$$

where $\boldsymbol{m}^{\eta}_{11} = \boldsymbol{m}^{\eta}_{22} = \boldsymbol{m}^{\eta}_{33} =$

$$\left(\begin{pmatrix} -1 & -2 & -1 \\ -2 & 0 & -2 \\ -1 & -2 & -1 \end{pmatrix} ; \begin{pmatrix} -2 & 0 & -2 \\ 0 & 32 & 0 \\ -2 & 0 & -2 \end{pmatrix} ; \begin{pmatrix} -1 & -2 & -1 \\ -2 & 0 & -2 \\ -1 & -2 & -1 \end{pmatrix} \right)_{3\times3\times3} , \tag{3.25}$$

$\boldsymbol{m}^{f}_{11} = {\boldsymbol{m}^{f}_{14}}^{T} = -{\boldsymbol{m}^{f}_{21}}^{T} = -{\boldsymbol{m}^{f}_{32}}^{T} = -\boldsymbol{m}^{f}_{33} =$

$$\left(\begin{pmatrix} 1 & 4 & 1 \\ 0 & 0 & 0 \\ -1 & -4 & -1 \end{pmatrix} ; \begin{pmatrix} 4 & 16 & 4 \\ 0 & 0 & 0 \\ -4 & -16 & -4 \end{pmatrix} ; \begin{pmatrix} 1 & 4 & 1 \\ 0 & 0 & 0 \\ -1 & -4 & -1 \end{pmatrix} \right)_{3\times3\times3} , \tag{3.26}$$

and $\boldsymbol{m}^f_{13} = -\boldsymbol{m}^f_{22} = \boldsymbol{m}^f_{31} =$

$$\left(\begin{pmatrix} 1 & 4 & 1 \\ 4 & 16 & 4 \\ 1 & 4 & 1 \end{pmatrix}; \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}; -\begin{pmatrix} 1 & 4 & 1 \\ 4 & 16 & 4 \\ 1 & 4 & 1 \end{pmatrix}\right)_{3\times3\times3} . \tag{3.27}$$

Thus, we can get the solution by using inverse filtering,

$$\begin{aligned} \tilde{\boldsymbol{\eta}}^x(I) &= (\boldsymbol{m}^\eta_{11})^{-1} \otimes_t \tilde{\boldsymbol{F}}^1(I) \\ \tilde{\boldsymbol{\eta}}^y(I) &= (\boldsymbol{m}^\eta_{22})^{-1} \otimes_t \tilde{\boldsymbol{F}}^2(I) \\ \tilde{\boldsymbol{\eta}}^z(I) &= (\boldsymbol{m}^\eta_{33})^{-1} \otimes_t \tilde{\boldsymbol{F}}^3(I) \end{aligned} \tag{3.28}$$

3.2.2.3 Solving the Inverse Filters

Rewriting the linear system Eq. (3.17) into a form involving only the convolutions as Eq. (3.22) and Eq. (3.28) poses another question, which is how to find the inverse filter $\boldsymbol{m}^{-1}$ for the given $\boldsymbol{m}$ using the truncated convolution operator $\otimes_t$.

A popular method to approximate the inverse of a finite impulse response (FIR) filter is by minimizing the least mean square (LMS) error between the response of $\tilde{\boldsymbol{m}}^{-1} \otimes_t \boldsymbol{m}$ and an impulse [35]. However, this approach involves representing the convolution in matrix form and finding the inverse of the resulting matrix, which goes back to where we started before and thus is not practically useful if the size of $\tilde{\boldsymbol{m}}^{-1}$ needs to be large, which is the case for our application.

To overcome this problem, we adopted the successive over relaxation (SOR) method [36] to approximate the inverse of a given FIR filter. To increase the rate of convergence, a good initial approximation of $\boldsymbol{m}^{-1}$ is desirable. Since this inverse filter is independent of the image data, it can be calculated offline. In this work, we use the method in [35] to find a small inverse filter and expand it by zero-padding to become the initial approximate of a larger inverse filter. This procedure is repeated until a desired size is achieved.

3.2.2.4 Optimization

An efficient optimizer is a key element to the proposed nonrigid image registration framework. A brute force optimization approach was previously reported in [37] to test the feasibility of the proposed algorithm. In the following sections, a gradient based optimization approach is described . This approach employs the “discretize then optimize” [38] concept.

2D Optimization Assuming SSD is applied as the similarity measure, then the required gradient of SSD with respect to the parameters $\boldsymbol{f} = (f^1, f^2)^T$ can be expressed as

$$\frac{\partial ssd}{\partial f^n(I_i)}, \tag{3.29}$$

where I_i represents the i_{th} grid point on mesh I, $i = 1, 2, \ldots, Nd$, Nd is the total number of grid points, and $n = 1$ or 2 representing the component of $\boldsymbol{f}$. Based on the inverse filtering formulation of the solution of the div-curl system, by applying the chain rule repeatedly, we can derive the following expression:

$$\begin{aligned}
\frac{\partial ssd}{\partial f^n(I_i)} = & \sum_{k\in N(j)} \frac{\partial ssd}{\partial \phi^x(I_k)} \frac{\partial \phi^x(I_k)}{\partial \eta^x(I_k)} \sum_{j\in N(i)} \frac{\partial \eta^x(I_k)}{\partial \tilde{F}^1(I_j)} \frac{\partial \tilde{F}^1(I_j)}{\partial f^n(I_i)} \\
& + \sum_{k\in N(j)} \frac{\partial ssd}{\partial \phi^x(I_k)} \frac{\partial \phi^x(I_k)}{\eta^x(I_k)} \sum_{j\in N(i)} \frac{\partial \eta^x(I_k)}{\partial \tilde{F}^2(I_j)} \frac{\partial \tilde{F}^2(I_j)}{\partial f^n(I_i)} \\
& + \sum_{k\in N(j)} \frac{\partial ssd}{\partial \phi^y(I_k)} \frac{\partial \phi^y(I_k)}{\partial \eta^y(I_k)} \sum_{j\in N(i)} \frac{\partial \eta^y(I_k)}{\partial \tilde{F}^1(I_j)} \frac{\partial \tilde{F}^1(I_j)}{\partial f^n(I_i)}
\end{aligned}$$

$$+\sum_{k\in N(j)}\frac{\partial ssd}{\partial\phi^y(I_k)}\frac{\partial\phi^y(I_k)}{\partial\eta^y(I_k)}\sum_{j\in N(i)}\frac{\partial\eta^y(I_k)}{\partial\tilde{F}^2(I_j)}\frac{\partial\tilde{F}^2(I_j)}{\partial f^n(I_i)} \tag{3.30}$$

where the superscripts x, y indicate the x, y component of the vector fields $\boldsymbol{\eta}$ and $\boldsymbol{\phi}$. $N(i)$ represents the set of adjacent pixel indices of the grid point I_i, which can be determined from Eq. (3.19). $N(j)$ is the neighboring pixel indices set of the grid point I_j, which can be determined from the inverse filters in Eq. (3.22). From the inverse filtering solution Eq. (3.22), we know that η^x depends on $\tilde{F}^1(I)$ and η^y depends on $\tilde{F}^2(I)$ only, therefore, only two terms left in Eq. (3.30). That is,

$$\begin{aligned}\frac{\partial ssd(I)}{\partial f^n(I_i)} &= \sum_{k\in N(j)}\frac{\partial ssd(I)}{\partial\phi^x(I_k)}\frac{\partial\phi^x(I_k)}{\partial\eta^x(I_k)}\sum_{j\in N(i)}\frac{\partial\eta^x(I_k)}{\partial\tilde{F}^1(I_j)}\frac{\partial\tilde{F}^1(I_j)}{\partial f^n(I_i)}\\ &\quad+\sum_{k\in N(j)}\frac{\partial ssd(I)}{\partial\phi^y(I_k)}\frac{\partial\phi^y(I_k)}{\partial\eta^y(I_k)}\sum_{j\in N(i)}\frac{\partial\eta^y(I_k)}{\partial\tilde{F}^2(I_j)}\frac{\partial\tilde{F}^2(I_j)}{\partial f^n(I_i)},\end{aligned} \tag{3.31}$$

where $\dfrac{\partial ssd}{\partial\phi^n}$, $n = x$ or y, can be derived as

$$\frac{\partial ssd}{\partial\phi^n} = \frac{\partial(T-R)^2}{\partial T}\frac{\partial T}{\partial\phi^n} = 2(T-R)\nabla_n T. \tag{3.32}$$

As for $\dfrac{\partial\phi^n}{\partial\eta^n}$, $n = x$ or y, referring to Eq. (3.3) and Eq. (3.4), if ϕ_1^n is approximated by the one step first order Runge-Kutta method, then

$$V_0^n = \frac{\eta^n}{f}, \tag{3.33}$$

$$\phi_1^n = \phi_0^n + V_0^n\times(1-0) = \phi_0^n + \frac{\eta^n}{f}, \tag{3.34}$$

$$\frac{\partial\phi^n}{\partial\eta^n} = \frac{\partial\left(\phi_0^n + \dfrac{\eta^n}{f}\right)}{\partial\eta^n} = \frac{1}{f}. \tag{3.35}$$

If we try to approximate ϕ_1^n by a higher order Runge-Kutta method or more than one time step approach, $\dfrac{\partial \phi^n}{\partial \eta^n}$ will be too complex to be obtained.

With Eq. (3.32), Eq. (3.35), Eq. (3.19), and Eq. (3.22), we can revise Eq. (3.31) into the convolution form

$$\begin{aligned}\frac{\partial ssd(I)}{\partial f^n(I)} = &\left(2(T(I) - R(I))(\nabla_x T(I))\frac{1}{f(I)}\right) \otimes_t (\boldsymbol{m}_{11}^{\eta})^{-1} \otimes_t \boldsymbol{m}_{1n}^{f} \\ &+ \left(2(T(I) - R(I))(\nabla_y T(I))\frac{1}{f(I)}\right) \otimes_t (\boldsymbol{m}_{22}^{\eta})^{-1} \otimes_t \boldsymbol{m}_{2n}^{f}. \end{aligned} \tag{3.36}$$

If a multi-resolution strategy is applied, Eq. (3.36) needs an additional interpolation kernel H, where H is determined by how we interpolate the monitor functions and curl values on coarse grid I_h with knot space h into the fine grid I. Therefore, with multi-resolution, Eq. (3.36) is rewritten as

$$\begin{aligned}\frac{\partial ssd(I)}{\partial f^n(I_h)} = &\left(2(T(I) - R(I))(\nabla_x T(I))\frac{1}{f(I)}\right) \otimes_t (\boldsymbol{m}_{11}^{\eta})^{-1} \otimes_t \boldsymbol{m}_{1n}^{f} \otimes_t H \\ &+ \left(2(T(I) - R(I))(\nabla_y T(I))\frac{1}{f(I)}\right) \otimes_t (\boldsymbol{m}_{22}^{\eta})^{-1} \otimes_t \boldsymbol{m}_{2n}^{f} \otimes_t H. \end{aligned} \tag{3.37}$$

3D Optimization Similarly, to derive the gradient of the similarity measure with respect to the parameters $\boldsymbol{f} = (f^1, f^2, f^3, f^4)^T$ at each of the grid points in 3D, we employ SSD to be the similarity measure and express the gradient of SSD as

$$\frac{\partial ssd}{\partial f^n(I_i)}, \tag{3.38}$$

where I_i represents the i^{th} grid point of mesh I, $i = 1, 2, \ldots, Nd$, Nd is the total number of grid points, and $n = 1 \sim 4$, representing the component of $\boldsymbol{f}$. By applying the chain rule, we can derive the following expression:

$$\begin{aligned}
\frac{\partial ssd}{\partial f^n(I_i)} = & \sum_{k \in N(j)} \frac{\partial ssd}{\partial \phi^x(I_k)} \frac{\partial \phi^x(I_k)}{\partial \eta^x(I_k)} \sum_{j \in N(i)} \frac{\partial \eta^x(I_k)}{\partial \tilde{F}^1(I_j)} \frac{\partial \tilde{F}^1(I_j)}{\partial f^n(I_i)} \\
& + \sum_{k \in N(j)} \frac{\partial ssd}{\partial \phi^x(I_k)} \frac{\partial \phi^x(I_k)}{\eta^x(I_k)} \sum_{j \in N(i)} \frac{\partial \eta^x(I_k)}{\partial \tilde{F}^2(I_j)} \frac{\partial \tilde{F}^2(I_j)}{\partial f^n(I_i)} \\
& + \sum_{k \in N(j)} \frac{\partial ssd}{\partial \phi^x(I_k)} \frac{\partial \phi^x(I_k)}{\eta^x(I_k)} \sum_{j \in N(i)} \frac{\partial \eta^x(I_k)}{\partial \tilde{F}^3(I_j)} \frac{\partial \tilde{F}^3(I_j)}{\partial f^n(I_i)} \\
& + \sum_{k \in N(j)} \frac{\partial ssd}{\partial \phi^y(I_k)} \frac{\partial \phi^y(I_k)}{\partial \eta^y(I_k)} \sum_{j \in N(i)} \frac{\partial \eta^y(I_k)}{\partial \tilde{F}^1(I_j)} \frac{\partial \tilde{F}^1(I_j)}{\partial f^n(I_i)} \\
& + \sum_{k \in N(j)} \frac{\partial ssd}{\partial \phi^y(I_k)} \frac{\partial \phi^y(I_k)}{\partial \eta^y(I_k)} \sum_{j \in N(i)} \frac{\partial \eta^y(I_k)}{\partial \tilde{F}^2(I_j)} \frac{\partial \tilde{F}^2(I_j)}{\partial f^n(I_i)} \\
& + \sum_{k \in N(j)} \frac{\partial ssd}{\partial \phi^y(I_k)} \frac{\partial \phi^y(I_k)}{\partial \eta^y(I_k)} \sum_{j \in N(i)} \frac{\partial \eta^y(I_k)}{\partial \tilde{F}^3(I_j)} \frac{\partial \tilde{F}^3(I_j)}{\partial f^n(I_i)} \\
& + \sum_{k \in N(j)} \frac{\partial ssd}{\partial \phi^z(I_k)} \frac{\partial \phi^z(I_k)}{\partial \eta^z(I_k)} \sum_{j \in N(i)} \frac{\partial \eta^z(I_k)}{\partial \tilde{F}^1(I_j)} \frac{\partial \tilde{F}^1(I_j)}{\partial f^n(I_i)} \\
& + \sum_{k \in N(j)} \frac{\partial ssd}{\partial \phi^z(I_k)} \frac{\partial \phi^z(I_k)}{\partial \eta^z(I_k)} \sum_{j \in N(i)} \frac{\partial \eta^z(I_k)}{\partial \tilde{F}^2(I_j)} \frac{\partial \tilde{F}^2(I_j)}{\partial f^n(I_i)} \\
& + \sum_{k \in N(j)} \frac{\partial ssd}{\partial \phi^z(I_k)} \frac{\partial \phi^z(I_k)}{\partial \eta^z(I_k)} \sum_{j \in N(i)} \frac{\partial \eta^z(I_k)}{\partial \tilde{F}^3(I_j)} \frac{\partial \tilde{F}^3(I_j)}{\partial f^n(I_i)}
\end{aligned} \tag{3.39}$$

where the superscripts x, y, and z indicate the three components of the vector fields $\boldsymbol{\eta}$ and $\boldsymbol{\phi}$. $N(i)$ represents the set of adjacent pixel indices of the grid point I_i, which can be determined from Eq. (3.19). $N(j)$ is the neighboring pixel indices set of the grid point $I(j)$, which can be determined from the inverse filters in Eq. (3.28). From the inverse filtering solution Eq. (3.28), $\eta^x(I)$ depends on $\tilde{F}^1(I)$, $\eta^y(I)$ depends on

$\tilde{F}^2(I)$, and $\eta^z(I)$ depends on $\tilde{F}^3(I)$ only. Therefore, Eq. (3.39) can be simplified into three terms:

$$\begin{aligned}\frac{\partial ssd(I)}{\partial f^n(I_i)} = &\sum_{k\in N(j)} \frac{\partial ssd(I)}{\partial \phi^x(I_k)}\frac{\partial \phi^x(I_k)}{\partial \eta^x(I_k)} \sum_{j\in N(i)} \frac{\partial \eta^x(I_k)}{\partial \tilde{F}^1(I_j)}\frac{\partial \tilde{F}^1(I_j)}{\partial f^n(I_i)} \\ &+ \sum_{k\in N(j)} \frac{\partial ssd(I)}{\partial \phi^y(I_k)}\frac{\partial \phi^y(I_k)}{\partial \eta^y(I_k)} \sum_{j\in N(i)} \frac{\partial \eta^y(I_k)}{\partial \tilde{F}^2(I_j)}\frac{\partial \tilde{F}^2(I_j)}{\partial f^n(I_i)} \\ &+ \sum_{k\in N(j)} \frac{\partial ssd(I)}{\partial \phi^z(I_k)}\frac{\partial \phi^z(I_k)}{\partial \eta^z(I_k)} \sum_{j\in N(i)} \frac{\partial \eta^z(I_k)}{\partial \tilde{F}^3(I_j)}\frac{\partial \tilde{F}^3(I_j)}{\partial f^n(I_i)}. \end{aligned} \tag{3.40}$$

Similar to the 2D case, from Eq. (3.32), Eq. (3.35), Eq. (3.19) and Eq. (3.28), we can rewrite Eq. (3.40) into the convolution form

$$\begin{aligned}\frac{\partial ssd(I)}{\partial f^n(I)} = &\left(2(T(I)-R(I))(\nabla_x T(I))\frac{1}{f(I)}\right) \otimes_t (\boldsymbol{m}^{\eta}_{11})^{-1} \otimes_t \boldsymbol{m}^{f}_{1n} \\ &+ \left(2(T(I)-R(I))(\nabla_y T(I))\frac{1}{f(I)}\right) \otimes_t (\boldsymbol{m}^{\eta}_{22})^{-1} \otimes_t \boldsymbol{m}^{f}_{2n} \\ &+ \left(2(T(I)-R(I))(\nabla_z T(I))\frac{1}{f(I)}\right) \otimes_t (\boldsymbol{m}^{\eta}_{33})^{-1} \otimes_t \boldsymbol{m}^{f}_{3n}. \end{aligned} \tag{3.41}$$

If multi-resolution is applied, Eq. (3.41) needs an additional interpolation kernel H, where H is determined by how we interpolate the monitor functions and curl values on coarse grid I_h with knot space h into the fine grid I. Therefore, with multi-resolution, Eq. (3.41) is rewritten as

$$\begin{aligned}\frac{\partial ssd(I)}{\partial f^n(I_h)} = &\left(2(T(I)-R(I))(\nabla_x T(I))\frac{1}{f(I)}\right) \otimes_t (\boldsymbol{m}^{\eta}_{11})^{-1} \otimes_t \boldsymbol{m}^{f}_{1n} \otimes_t H \\ &+ \left(2(T(I)-R(I))(\nabla_y T(I))\frac{1}{f(I)}\right) \otimes_t (\boldsymbol{m}^{\eta}_{22})^{-1} \otimes_t \boldsymbol{m}^{f}_{2n} \otimes_t H \\ &+ \left(2(T(I)-R(I))(\nabla_z T(I))\frac{1}{f(I)}\right) \otimes_t (\boldsymbol{m}^{\eta}_{33})^{-1} \otimes_t \boldsymbol{m}^{f}_{3n} \otimes_t H. \end{aligned} \tag{3.42}$$

3.2.3 Gradient Descent Optimization Strategy

Once the $\dfrac{\partial ssd}{\partial f^n}$ is available, a 2D gradient descent optimization strategy to minimize the SSD with respect to $\boldsymbol{f}$ can be devised as Fig. 3.3 shows. The gradient descent optimization strategy for the 3D case is analogous to the 2D case.

Notice that for the optimization strategy in the 3D case, according to Eq. 3.24, the intermediate vector field $\boldsymbol{\eta}$ can be optimized by f^1 and any two of the three curl components f^2, f^3, and f^4. Which means in 3D case, we should be able to optimize three variables instead of four. An experiment in Sec. 4.8 will evaluate this assertion.

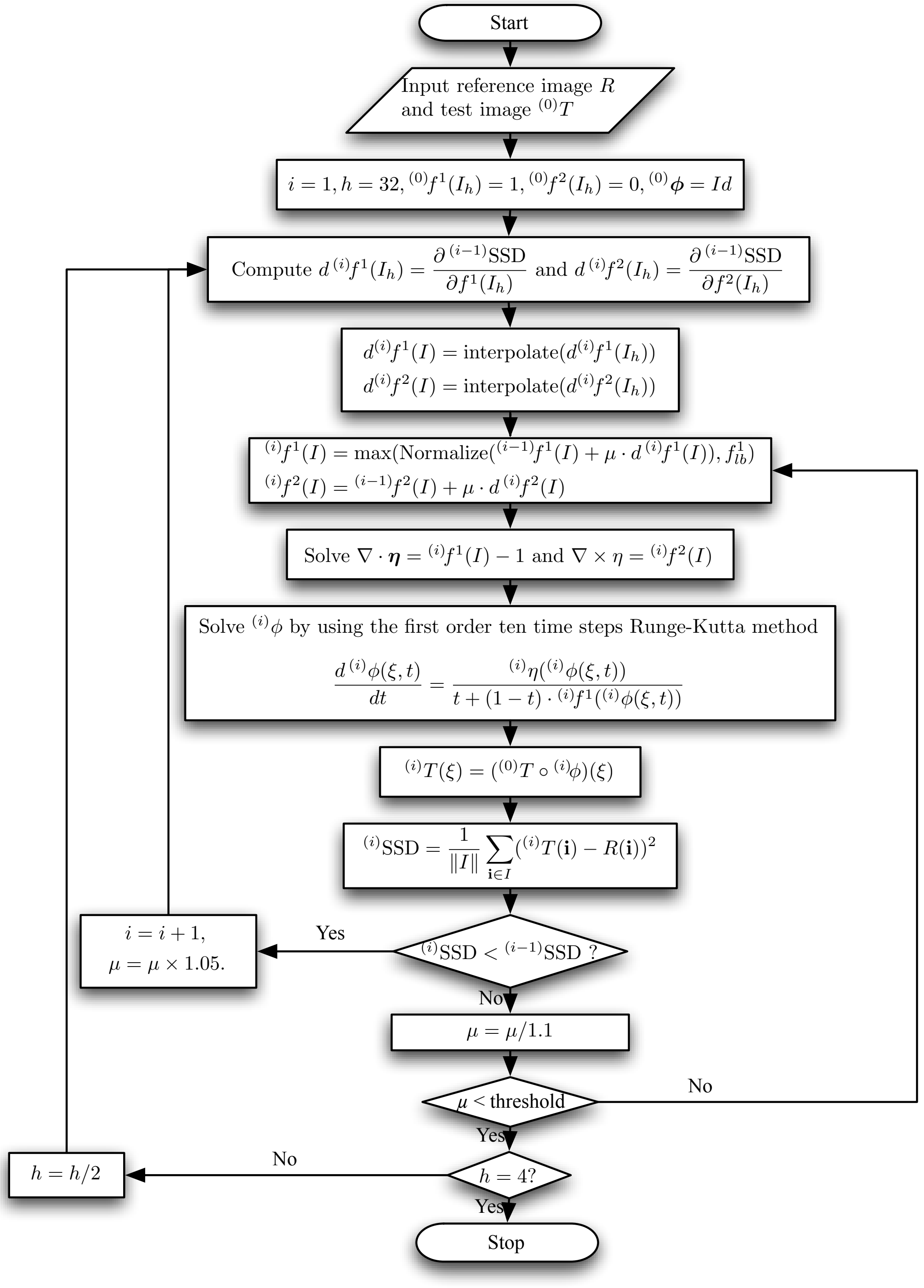

Start
Input reference image R and test image ${}^{(0)}T$
$i = 1, h = 32, {}^{(0)}f^1(I_h) = 1, {}^{(0)}f^2(I_h) = 0, {}^{(0)}\boldsymbol{\phi} = Id$
Compute $d\,{}^{(i)}f^1(I_h) = \dfrac{\partial\,{}^{(i-1)}\text{SSD}}{\partial f^1(I_h)}$ and $d\,{}^{(i)}f^2(I_h) = \dfrac{\partial\,{}^{(i-1)}\text{SSD}}{\partial f^2(I_h)}$
$d^{(i)}f^1(I) = \text{interpolate}(d^{(i)}f^1(I_h))$
$d^{(i)}f^2(I) = \text{interpolate}(d^{(i)}f^2(I_h))$
${}^{(i)}f^1(I) = \max(\text{Normalize}({}^{(i-1)}f^1(I) + \mu \cdot d\,{}^{(i)}f^1(I)), f^1_{lb})$
${}^{(i)}f^2(I) = {}^{(i-1)}f^2(I) + \mu \cdot d\,{}^{(i)}f^2(I)$
Solve $\nabla \cdot \boldsymbol{\eta} = {}^{(i)}f^1(I) - 1$ and $\nabla \times \eta = {}^{(i)}f^2(I)$
Solve ${}^{(i)}\phi$ by using the first order ten time steps Runge-Kutta method
$\dfrac{d\,{}^{(i)}\phi(\xi,t)}{dt} = \dfrac{{}^{(i)}\eta({}^{(i)}\phi(\xi,t))}{t + (1-t)\cdot {}^{(i)}f^1({}^{(i)}\phi(\xi,t))}$
${}^{(i)}T(\xi) = ({}^{(0)}T \circ {}^{(i)}\phi)(\xi)$
${}^{(i)}\text{SSD} = \dfrac{1}{\|I\|}\sum_{\mathbf{i}\in I}({}^{(i)}T(\mathbf{i}) - R(\mathbf{i}))^2$
${}^{(i)}\text{SSD} < {}^{(i-1)}\text{SSD}$?
Yes
$i = i + 1,$
$\mu = \mu \times 1.05.$
No
$\mu = \mu/1.1$
μ < threshold
No
Yes
$h = 4?$
No
$h = h/2$
Yes
Stop

Figure 3.3. The gradient descent optimization strategy of the nonrigid image registration by the deformation method version one where f^1_{lb} represents the lower bound of monitor function f^1.

3.3 Version 2

3.3.1 Permissible Set

For version 2, we define a *permissible* set as the collection of deformation fields generated by the following steps.

1. Generate an arbitrary positive scalar monitor function f and an arbitrary scalar (for 2D case) or vector (for 3D case) curl $\boldsymbol{g}$ on a discrete image domain I, where f is normalized by
$$\int_\Omega \frac{1}{f} = |\Omega|. \tag{3.43}$$
In 3D case, $\boldsymbol{g}$ consists of three components g^1, g^2, and g^3.

2. Solve the following div-curl system for an intermediate vector field $\boldsymbol{\eta}(\boldsymbol{\xi})$ using the Dirichlet boundary condition.
$$\begin{cases} \nabla \cdot \boldsymbol{\eta}(\boldsymbol{\xi}) = 1 - \dfrac{1}{f(\boldsymbol{\xi})} \\ \nabla \times \boldsymbol{\eta}(\boldsymbol{\xi}) = \boldsymbol{g}(\boldsymbol{\xi}) \end{cases} \tag{3.44}$$

3. Generate a time-dependent velocity vector field $\boldsymbol{V}$ from $\boldsymbol{\eta}$ and f
$$\boldsymbol{V}_t(\boldsymbol{\phi}_t(\boldsymbol{\xi})) = \boldsymbol{V}(\boldsymbol{\phi}(\boldsymbol{\xi}, t), t) = \frac{\boldsymbol{\eta}(\boldsymbol{\phi}_t(\boldsymbol{\xi}))}{t\dfrac{1}{f(\boldsymbol{\phi}_t(\boldsymbol{\xi}))} + (1-t)}. \tag{3.45}$$

4. Solve $\boldsymbol{\phi}_1(\boldsymbol{\xi})$ by solving the following ODE
$$\frac{d\boldsymbol{\phi}_t(\boldsymbol{\xi})}{dt} = \boldsymbol{V}_t(\boldsymbol{\phi}_t(\boldsymbol{\xi})), \quad t \in [0,1]. \tag{3.46}$$

The deformation field $\boldsymbol{\phi}$ generated by this procedure is called a *permissible* deformation because it possesses the property that the Jacobian determinant of $\boldsymbol{\phi}_1(\boldsymbol{\xi})$ is equal to the specified strictly positive monitor function $f(\boldsymbol{\phi}_1(\boldsymbol{\xi}))$. This can be justified by the fact that the proof of $\boldsymbol{\phi}_1(\boldsymbol{\xi}) = f(\boldsymbol{\phi}_1(\boldsymbol{\xi}))$ in the deformation based grid generation version 2 [26, 33] is independent of the curl $\boldsymbol{g}$ of the intermediate field $\boldsymbol{\eta}$. Since the modification of curl $\boldsymbol{g}$ to be nonzero is the only modification we made, as long as the monitor function f is restricted to be strictly positive, the resulting deformation field $\boldsymbol{\phi}$ suffers no mesh folding and thus is physically plausible (permissible).

3.3.2 A Div-Curl Solver for Version 2

To solve Eq. (3.44), a div-curl solver is essential. The derivation of a div-curl solver for the nonrigid image registration based on the deformation method version 2 is similar to the derivation of the div-curl solver described in Sec. 3.2.2.1 and Sec. 3.2.2.2. We just need to substitute $f^1 = f - 1$ with $f^1 = 1 - 1/f$, where the monitor function f satisfies

$$\int_\Omega 1 - \frac{1}{f} = |\Omega|. \tag{3.47}$$

Then, for the 2D case, we can obtain $\boldsymbol{\eta}$ by

$$\begin{aligned} \tilde{\boldsymbol{\eta}}^x(I) &= (\boldsymbol{m}_{11}^{\eta})^{-1} \otimes_t \tilde{\boldsymbol{F}}^1(I) \\ \tilde{\boldsymbol{\eta}}^y(I) &= (\boldsymbol{m}_{22}^{\eta})^{-1} \otimes_t \tilde{\boldsymbol{F}}^2(I). \end{aligned} \tag{3.48}$$

For the 3D case, we can obtain

$$\begin{aligned} \tilde{\boldsymbol{\eta}}^x(I) &= (\boldsymbol{m}_{11}^{\eta})^{-1} \otimes_t \tilde{\boldsymbol{F}}^1(I) \\ \tilde{\boldsymbol{\eta}}^y(I) &= (\boldsymbol{m}_{22}^{\eta})^{-1} \otimes_t \tilde{\boldsymbol{F}}^2(I) \\ \tilde{\boldsymbol{\eta}}^z(I) &= (\boldsymbol{m}_{33}^{\eta})^{-1} \otimes_t \tilde{\boldsymbol{F}}^3(I). \end{aligned} \tag{3.49}$$

3.3.3 Optimization

Similar to the Sec. 3.2.2.4, we can obtain

$$\begin{aligned}\frac{\partial ssd(I)}{\partial f^n(I_i)} = & \sum_{k\in N(j)} \frac{\partial ssd(I)}{\partial \phi^x(I_k)} \frac{\partial \phi^x(I_k)}{\partial \eta^x(I_k)} \sum_{j\in N(i)} \frac{\partial \eta^x(I_k)}{\partial \tilde{F}^1(I_j)} \frac{\partial \tilde{F}^1(I_j)}{\partial f^n(I_i)} \\ & + \sum_{k\in N(j)} \frac{\partial ssd(I)}{\partial \phi^y(I_k)} \frac{\partial \phi^y(I_k)}{\partial \eta^y(I_k)} \sum_{j\in N(i)} \frac{\partial \eta^y(I_k)}{\partial \tilde{F}^2(I_j)} \frac{\partial \tilde{F}^2(I_j)}{\partial f^n(I_i)}. \end{aligned} \tag{3.50}$$

For $\frac{\partial \phi^n}{\partial \eta^n}$, $n = x$ or y, referring to Eq. (3.45) and Eq. (3.46), if ϕ_1^n is approximated by the one step first order Runge-Kutta method, where $\phi_1^n = \phi^n(t = 1)$, then

$$V_0^n = \eta^n, \tag{3.51}$$

$$\phi_1^n = \phi_0^n + V_0^n \times (1 - 0) = \phi_0^n + \eta^n, \tag{3.52}$$

$$\frac{\partial \phi^n}{\partial \eta^n} = \frac{\partial\left(\phi_0^n + \eta^n\right)}{\partial \eta^n} = 1. \tag{3.53}$$

From Eq. (3.32), Eq. (3.53), Eq. (3.19), and Eq. (3.48), we can revise Eq. (3.50) into the convolution form

$$\begin{aligned}\frac{\partial ssd(I)}{\partial f^n(I)} = & \left(2(T(I) - R(I))(\nabla_x T(I))\right) \otimes_t (\boldsymbol{m}_{11}^{\eta})^{-1} \otimes_t \boldsymbol{m}_{1n}^{f} \\ & + \left(2(T(I) - R(I))(\nabla_y T(I))\right) \otimes_t (\boldsymbol{m}_{22}^{\eta})^{-1} \otimes_t \boldsymbol{m}_{2n}^{f}. \end{aligned} \tag{3.54}$$

If a multi-resolution strategy is applied, Eq. (3.36) needs an additional interpolation kernel H, where H is determined by how we interpolate the monitor functions and curl values on coarse grid I_h with knot space h into the fine grid I. Therefore, with multi-resolution, Eq. (3.36) is rewritten as

$$\frac{\partial ssd(I)}{\partial f^n(I_h)} = (2(T(I) - R(I))\nabla_x T(I)) \otimes_t (\boldsymbol{m}_{11}^{\eta})^{-1} \otimes_t \boldsymbol{m}_{1n}^{f} \otimes_t H$$

$$+ (2(T(I) - R(I))\nabla_y T(I)) \otimes_t (\boldsymbol{m}^{\eta}_{22})^{-1} \otimes_t \boldsymbol{m}^{f}_{2n} \otimes_t H. \qquad (3.55)$$

3.3.4 Gradient Descent Optimization Strategy

Once the $\dfrac{\partial ssd}{\partial f^n}$ is available, a 2D gradient descent optimization strategy to minimize the SSD with respect to $\boldsymbol{f}$ can be devised as Fig. 3.4 shows. The gradient descent optimization strategy for the 3D case is analogous to the 2D case.

Notice that for the optimization strategy in the 3D case, according to Eq. 3.24, the intermediate vector field $\boldsymbol{\eta}$ can be optimized by any three of the f_1, f_2, f_3, and f_4. Which means in 3D case, we just need to optimize three kinds of variables instead of four. An experiment in Sec. 4.8 is conducted to evaluate this assertion.

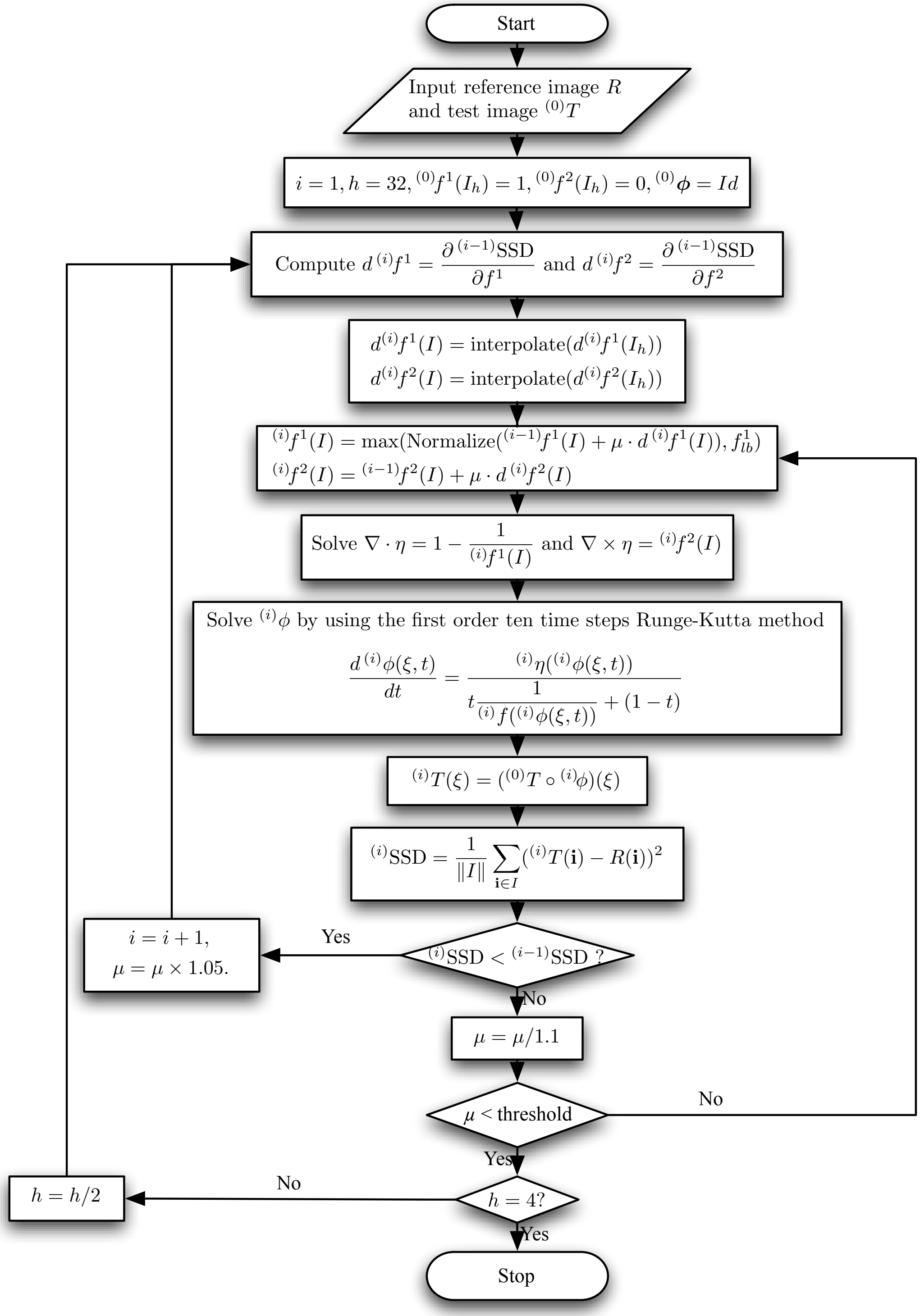

Start
Input reference image R
and test image ${}^{(0)}T$
$i = 1, h = 32, {}^{(0)}f^1(I_h) = 1, {}^{(0)}f^2(I_h) = 0, {}^{(0)}\boldsymbol{\phi} = Id$
Compute $d\,{}^{(i)}f^1 = \frac{\partial\,{}^{(i-1)}\mathrm{SSD}}{\partial f^1}$ and $d\,{}^{(i)}f^2 = \frac{\partial\,{}^{(i-1)}\mathrm{SSD}}{\partial f^2}$
$d^{(i)}f^1(I) = \mathrm{interpolate}(d^{(i)}f^1(I_h))$
$d^{(i)}f^2(I) = \mathrm{interpolate}(d^{(i)}f^2(I_h))$
${}^{(i)}f^1(I) = \max(\mathrm{Normalize}({}^{(i-1)}f^1(I) + \mu \cdot d\,{}^{(i)}f^1(I)), f^1_{lb})$
${}^{(i)}f^2(I) = {}^{(i-1)}f^2(I) + \mu \cdot d\,{}^{(i)}f^2(I)$
Solve $\nabla \cdot \eta = 1 - \frac{1}{{}^{(i)}f^1(I)}$ and $\nabla \times \eta = {}^{(i)}f^2(I)$
Solve ${}^{(i)}\phi$ by using the first order ten time steps Runge-Kutta method
$\frac{d\,{}^{(i)}\phi(\xi,t)}{dt} = \frac{{}^{(i)}\eta({}^{(i)}\phi(\xi,t))}{t\frac{1}{{}^{(i)}f({}^{(i)}\phi(\xi,t))} + (1-t)}$
${}^{(i)}T(\xi) = ({}^{(0)}T \circ {}^{(i)}\phi)(\xi)$
${}^{(i)}\mathrm{SSD} = \frac{1}{\|I\|}\sum_{\mathbf{i}\in I}({}^{(i)}T(\mathbf{i}) - R(\mathbf{i}))^2$
Yes
$i = i + 1,$
$\mu = \mu \times 1.05.$
${}^{(i)}\mathrm{SSD} < {}^{(i-1)}\mathrm{SSD}$?
No
$\mu = \mu/1.1$
μ < threshold
No
Yes
No
$h = h/2$
$h = 4?$
Yes
Stop

Figure 3.4. The gradient descent optimization strategy of the nonrigid image registration by the deformation method version two where f^1_{lb} represents the lower bound of monitor function f^1.

3.4 Version 3

3.4.1 Permissible Set

For version 3, a *permissible* set is defined as a collection of deformation fields generated by the following steps.

1. Generate an arbitrary time-varing positive scalar monitor function f_t and an arbitrary scalar (for 2D case) or vector (for 3D case) curl $\boldsymbol{g}$ on a discrete image domain I, where f_t is normalized by

$$\int_{\Omega(t)} \frac{1}{f_t} = |\Omega(t=0)|. \tag{3.56}$$

   In 3D case, $\boldsymbol{g}$ consists of three components g^1, g^2, and g^3.

2. Solve the following div-curl system for an intermediate vector field $\boldsymbol{\eta}(\boldsymbol{\xi})$ using the Dirichlet boundary condition.

$$\begin{cases} \nabla \cdot \boldsymbol{\eta}(\boldsymbol{\xi}, t) = -\dfrac{\partial}{\partial t}\dfrac{1}{f(\boldsymbol{\xi}, t)} \\ \nabla \times \boldsymbol{\eta}(\boldsymbol{\xi}, t) = \boldsymbol{g}(\boldsymbol{\xi}, t) \end{cases} \tag{3.57}$$

3. Generate a time-dependent velocity vector field $\boldsymbol{V}_t$ from $\boldsymbol{\eta}_t$ and f_t

$$\boldsymbol{V}_t(\boldsymbol{\phi}_t(\boldsymbol{\xi})) = \boldsymbol{V}(\boldsymbol{\phi}(\boldsymbol{\xi}, t), t) = \boldsymbol{\eta}_t(\boldsymbol{\phi}_t(\boldsymbol{\xi}))f_t(\boldsymbol{\phi}_t(\boldsymbol{\xi})). \tag{3.58}$$

4. Solve the deformation $\boldsymbol{\phi}_t(\boldsymbol{\xi})$ by solving the ODE

$$\frac{d\boldsymbol{\phi}_t(\boldsymbol{\xi})}{dt} = \boldsymbol{V}_t(\boldsymbol{\phi}_t(\boldsymbol{\xi})). \tag{3.59}$$

The deformation field $\boldsymbol{\phi}_t$ generated by this procedure is called a *permissible* deformation because it possesses the property that the Jacobian determinant of $\boldsymbol{\phi}_t(\boldsymbol{\xi})$

is equal to the specified strictly positive monitor function $f_t(\boldsymbol{\phi}_t(\boldsymbol{\xi}))$. This can be justified by the fact that the proof of $\boldsymbol{\phi}_t(\boldsymbol{\xi}) = f_t(\boldsymbol{\phi}_t(\boldsymbol{\xi}))$ in the deformation based grid generation version 3 [27, 33] is independent of the curl $\boldsymbol{g}$ of the intermediate field $\boldsymbol{\eta}$. Since the modification of curl $\boldsymbol{g}$ to be nonzero is the only modification we made, as long as the monitor function f is restricted to be strictly positive, the resulting deformation field $\boldsymbol{\phi}_t$ suffers no mesh folding and thus is physically plausible (permissible).

3.4.2 A Div-Curl Solver for Version 3

To solve Eq. (3.57), a div-curl solver is essential. The derivation of a div-curl solver for the nonrigid image registration based on the deformation method version 3 is similar to the derivation of the div-curl solver described in Sec. 3.2.2.1 and Sec. 3.2.2.2. We just need to substitute $f^1 = f - 1$ with

$$f^1 = -\frac{\partial}{\partial t}\frac{1}{f_t}, \tag{3.60}$$

where the monitor function at time t satisfies

$$\int_\Omega \frac{1}{f_t} = |\Omega(t = 0)|. \tag{3.61}$$

Then, for the 2D case, we can obtain $\boldsymbol{\eta}$ by

$$\begin{aligned} \tilde{\boldsymbol{\eta}}^x(I) &= (\boldsymbol{m}^{\eta}_{11})^{-1} \otimes_t \tilde{\boldsymbol{F}}^1(I) \\ \tilde{\boldsymbol{\eta}}^y(I) &= (\boldsymbol{m}^{\eta}_{22})^{-1} \otimes_t \tilde{\boldsymbol{F}}^2(I). \end{aligned} \tag{3.62}$$

For the 3D case, we can obtain

$$\begin{aligned}
\tilde{\boldsymbol{\eta}}^x(I) &= (\boldsymbol{m}_{11}^{\eta})^{-1} \otimes_t \tilde{\boldsymbol{F}}^1(I) \\
\tilde{\boldsymbol{\eta}}^y(I) &= (\boldsymbol{m}_{22}^{\eta})^{-1} \otimes_t \tilde{\boldsymbol{F}}^2(I) \\
\tilde{\boldsymbol{\eta}}^z(I) &= (\boldsymbol{m}_{33}^{\eta})^{-1} \otimes_t \tilde{\boldsymbol{F}}^3(I).
\end{aligned} \tag{3.63}$$

3.4.3 Optimization

Similar to the Sec. 3.2.2.4, we can obtain

$$\begin{aligned}
\frac{\partial ssd(I)}{\partial f^n(I_i)} &= \sum_{k\in N(j)} \frac{\partial ssd(I)}{\partial \phi^x(I_k)} \frac{\partial \phi^x(I_k)}{\partial \eta^x(I_k)} \sum_{j\in N(i)} \frac{\partial \eta^x(I_k)}{\partial \tilde{F}^1(I_j)} \frac{\partial \tilde{F}^1(I_j)}{\partial f^n(I_i)} \\
&\quad + \sum_{k\in N(j)} \frac{\partial ssd(I)}{\partial \phi^y(I_k)} \frac{\partial \phi^y(I_k)}{\partial \eta^y(I_k)} \sum_{j\in N(i)} \frac{\partial \eta^y(I_k)}{\partial \tilde{F}^2(I_j)} \frac{\partial \tilde{F}^2(I_j)}{\partial f^n(I_i)}.
\end{aligned} \tag{3.64}$$

From the optimization strategy illustrated in Fig. 3.5, the connection between $\boldsymbol{\phi}$ and $\boldsymbol{\eta}$ is through

$$\frac{d\,{}^{(i)}\boldsymbol{\phi}_t(I_k)}{dt} = {}^{(i)}\boldsymbol{\eta}({}^{(i)}\boldsymbol{\phi}_t(I_k)) \cdot {}^{(i)}f^1_{t_i}({}^{(i)}\boldsymbol{\phi}_t(I_k)), \quad t \in [t_{i-1}, t_i]. \tag{3.65}$$

If we attempt the simplest approximation of $\boldsymbol{\phi}_{t_i}$ by using the single step first order Runge-Kutta method at time t_i, we can obtain Eq. (3.66).

$$\boldsymbol{\phi}_{t_i}(I_k) = \boldsymbol{\phi}_{t_{i-1}}(I_k) + \boldsymbol{\eta}(\boldsymbol{\phi}_{t_{i-1}}(I_k)) f(\boldsymbol{\phi}_{t_{i-1}}(I_k)) \tag{3.66}$$

For version 3, it is difficult to obtain $\frac{\partial \phi^n(I_k)}{\partial \eta^n(I_k)}$ where $n = x$ or y, since what we can obtain from Eq. (3.66) is $\frac{\partial \phi_{t_i}(I_k)}{\partial \eta(\phi_{t_{i-1}}(I_k))}$. Therefore, we can not obtain $\frac{\partial ssd(I)}{\partial f^n(I)}$ as we did for the registration algorithm version 1 and version 2.

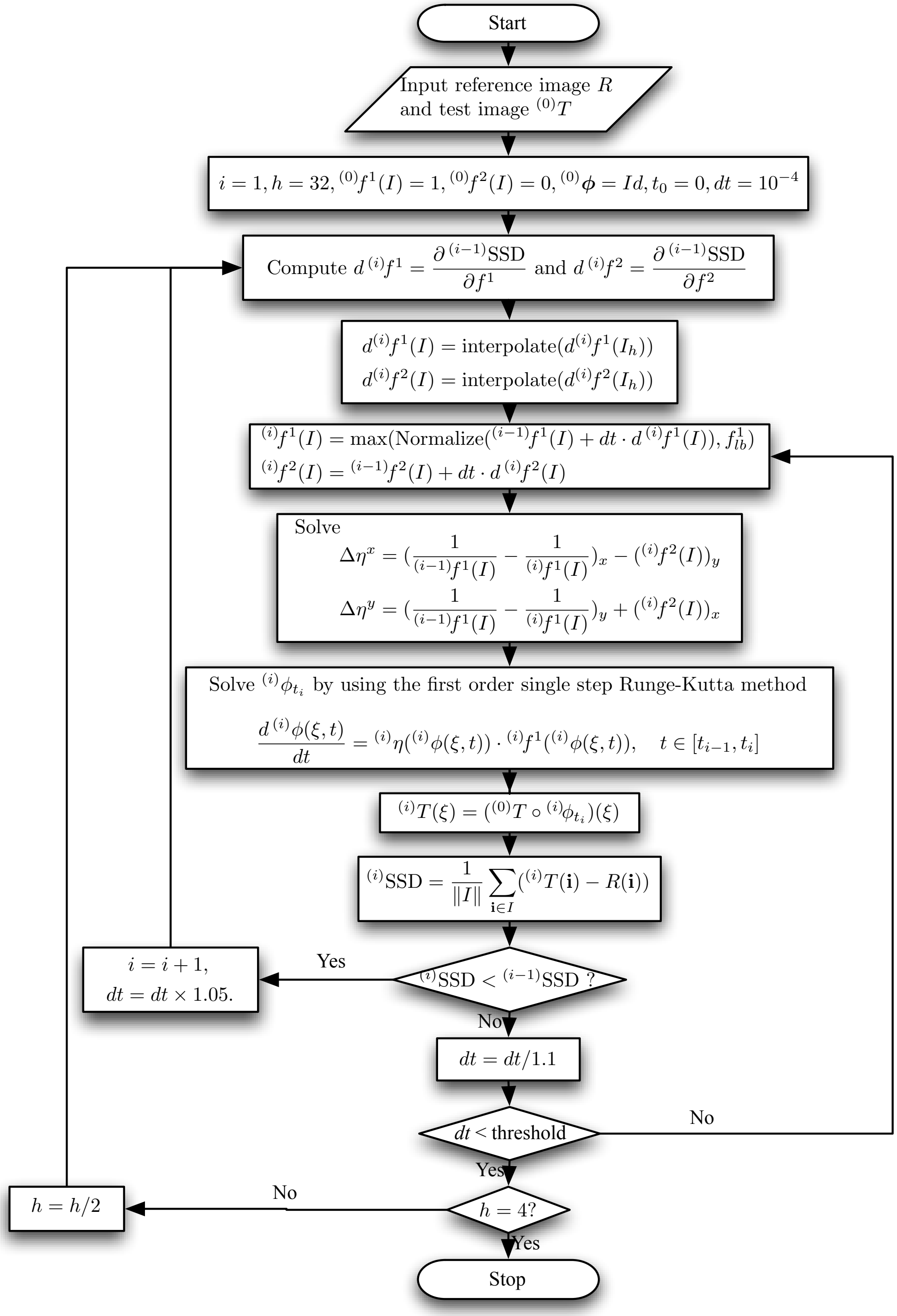

Start
Input reference image R and test image ${}^{(0)}T$
$i = 1, h = 32, {}^{(0)}f^1(I) = 1, {}^{(0)}f^2(I) = 0, {}^{(0)}\phi = Id, t_0 = 0, dt = 10^{-4}$
Compute $d\,{}^{(i)}f^1 = \frac{\partial\,{}^{(i-1)}\mathrm{SSD}}{\partial f^1}$ and $d\,{}^{(i)}f^2 = \frac{\partial\,{}^{(i-1)}\mathrm{SSD}}{\partial f^2}$
$d^{(i)}f^1(I) = \mathrm{interpolate}(d^{(i)}f^1(I_h))$
$d^{(i)}f^2(I) = \mathrm{interpolate}(d^{(i)}f^2(I_h))$
${}^{(i)}f^1(I) = \max(\mathrm{Normalize}({}^{(i-1)}f^1(I) + dt \cdot d\,{}^{(i)}f^1(I)), f^1_{lb})$
${}^{(i)}f^2(I) = {}^{(i-1)}f^2(I) + dt \cdot d\,{}^{(i)}f^2(I)$
Solve
$\Delta\eta^x = (\frac{1}{{}^{(i-1)}f^1(I)} - \frac{1}{{}^{(i)}f^1(I)})_x - ({}^{(i)}f^2(I))_y$
$\Delta\eta^y = (\frac{1}{{}^{(i-1)}f^1(I)} - \frac{1}{{}^{(i)}f^1(I)})_y + ({}^{(i)}f^2(I))_x$
Solve ${}^{(i)}\phi_{t_i}$ by using the first order single step Runge-Kutta method
$\frac{d\,{}^{(i)}\phi(\xi,t)}{dt} = {}^{(i)}\eta({}^{(i)}\phi(\xi,t)) \cdot {}^{(i)}f^1({}^{(i)}\phi(\xi,t)), \quad t \in [t_{i-1}, t_i]$
${}^{(i)}T(\xi) = ({}^{(0)}T \circ {}^{(i)}\phi_{t_i})(\xi)$
${}^{(i)}\mathrm{SSD} = \frac{1}{\|I\|}\sum_{\mathbf{i}\in I}({}^{(i)}T(\mathbf{i}) - R(\mathbf{i}))$
${}^{(i)}\mathrm{SSD} < {}^{(i-1)}\mathrm{SSD}$?
Yes
$i = i + 1,$
$dt = dt \times 1.05.$
No
$dt = dt/1.1$
dt < threshold
No
Yes
$h = 4?$
No
$h = h/2$
Yes
Stop

Figure 3.5. The gradient descent optimization strategy of the nonrigid image registration by the deformation method version three where f^1_{lb} represents the lower bound of monitor function f^1.

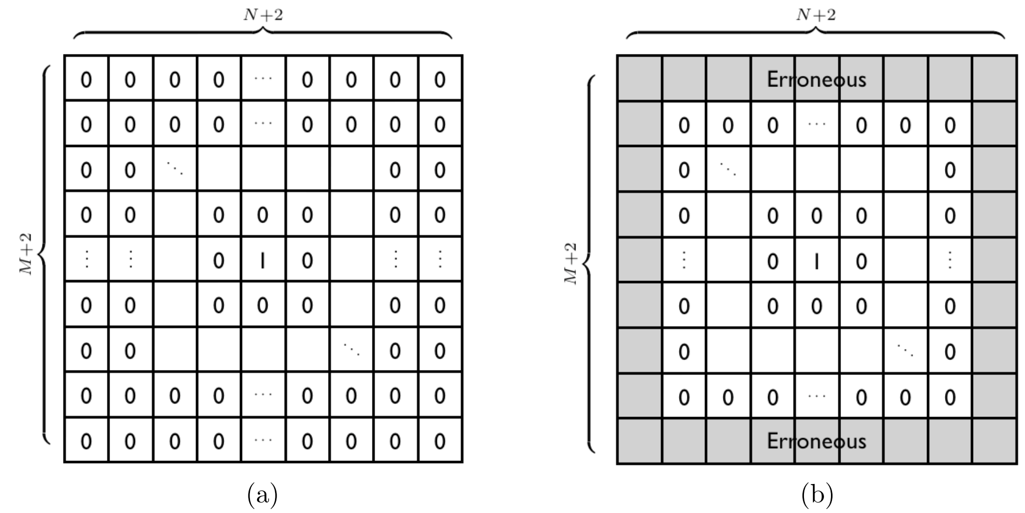


Figure 3.6. (a) Ideal $\boldsymbol{m}_{11}^{\eta} \otimes (\boldsymbol{m}_{11}^{\eta})_{ideal}^{-1}$ result. (b) Actual $\boldsymbol{m}_{11}^{\eta} \otimes (\boldsymbol{m}_{11}^{\eta})^{-1}$ result where the 0s represent the negligible numbers which are $< 10^{-8}$.

3.5 Precision and Performance Issue of Solving Div-Curl System by Inverse Filtering

Considering approximately solving an M by N matrix $\tilde{\eta}^{x}(I)$ in Eq. (3.19) by inverse filtering, we can obtain

$$(\boldsymbol{m}_{11}^{\eta})^{-1} \otimes_t (\boldsymbol{m}_{11}^{\eta} \otimes_t \tilde{\boldsymbol{\eta}}^{x}(I)) = (\boldsymbol{m}_{11}^{\eta})^{-1} \otimes_t \tilde{\boldsymbol{F}}^{1}(I). \tag{3.67}$$

Theoretically, if we can find an ideal inverse filter $(\boldsymbol{m}_{11}^{\eta})_{ideal}^{-1}$ of $\boldsymbol{m}_{11}^{\eta}$, then $(\boldsymbol{m}_{11}^{\eta})_{ideal}^{-1} \otimes \boldsymbol{m}_{11}^{\eta}$ should generate an impulse matrix as Fig. 3.6a shows, where the size of $(\boldsymbol{m}_{11}^{\eta})_{ideal}^{-1}$ is $M \times N$.

In section 3.2.2.3, we adopt the SOR to solve the inverse filter $(\boldsymbol{m}_{11}^{\eta})^{-1}$, which is possible only if we apply the truncated convolution operator $\otimes_t$. Which means that the result of $(\boldsymbol{m}_{11}^{\eta})^{-1} \otimes_t \boldsymbol{m}_{11}^{\eta}$ is an $M \times N$ impulse rather than $(M+2) \times (N+2)$. With an $M \times N$ inverse filter $(\boldsymbol{m}_{11}^{\eta})^{-1}$ obtained by using the SOR, $(\boldsymbol{m}_{11}^{\eta})^{-1} \otimes \boldsymbol{m}_{11}^{\eta}$ will generate a matrix enclosed by erroneous values as Fig. 3.6b shows. Therefore, to obtain a decent $\tilde{\boldsymbol{\eta}}^x(I)$ in Eq. (3.67), $(\boldsymbol{m}_{11}^{\eta})^{-1}$ has to be larger than $(2M+1)\times(2N+1)$ so that the erroneous area will not contaminate the solution $\tilde{\boldsymbol{\eta}}^x(I)$.

To implement $(\boldsymbol{m}_{11}^{\eta})^{-1} \otimes \tilde{F}^1(I)$ using FFT (fast fourier transform), only the central $M \times N$ sub-matrix is taken as our result to replace the convolution to improve the performance [39]. The computational complexity of $(\boldsymbol{m}_{11}^{\eta})^{-1} \otimes \boldsymbol{m}_{11}^{\eta}$ by FFT is $(2M+1)(2N+1)\log((2M+1)(2N+1))$.

3.5.1 Further Performance Enhancement

From Sec. 3.5, the complexity of solving the div-curl system through Eq. (3.22) and Eq. (3.28) using inverse filtering is high. Fortunately, it is possible to rewrite a div-curl system (2D or 3D) as a set of Poisson's equations and then be able to utilize the existing efficient Poisson solvers to find the solution and obtain the gradient information. To further improve the performance of the div-curl solver, we adopt the FFT Poisson solver [40] to solve the div-curl system since the complexity of the FFT Poisson solver for a $M \times N$ system is $MN\log(MN)$. The details of this approach is provided below.

3.5.1.1 Consider a Div-Curl System as a Set of Poisson's Equations

In 2D case, to transform the div-curl system into the Poisson's equations, we can apply the gradient operator on the div-curl system Eq. (3.68) and obtain Eq. (3.69).

$$\begin{aligned} \frac{\partial \eta^x}{\partial x} + \frac{\partial \eta^y}{\partial y} &= f^1 \\ \frac{\partial \eta^y}{\partial x} - \frac{\partial \eta^x}{\partial y} &= f^2 \end{aligned} \tag{3.68}$$

$$\begin{aligned} \nabla \left(\frac{\partial \eta^x}{\partial x} + \frac{\partial \eta^y}{\partial y} \right) &= \nabla f^1 \\ \nabla \left(\frac{\partial \eta^y}{\partial x} - \frac{\partial \eta^x}{\partial y} \right) &= \nabla f^2 \end{aligned} \tag{3.69}$$

Expanding the Eq. (3.69), we can get

$$\frac{\partial^2 \eta^x}{\partial x^2} + \frac{\partial^2 \eta^y}{\partial x \partial y} = \frac{\partial f^1}{\partial x} \tag{3.70a}$$

$$\frac{\partial^2 \eta^x}{\partial x \partial y} + \frac{\partial^2 \eta^y}{\partial y^2} = \frac{\partial f^1}{\partial y} \tag{3.70b}$$

$$\frac{\partial^2 \eta^y}{\partial x^2} - \frac{\partial^2 \eta^x}{\partial x \partial y} = \frac{\partial f^2}{\partial x} \tag{3.70c}$$

$$\frac{\partial^2 \eta^y}{\partial x \partial y} - \frac{\partial^2 \eta^x}{\partial y^2} = \frac{\partial f^2}{\partial y}. \tag{3.70d}$$

Combining (3.70a) with (3.70d) and (3.70b) with (3.70c), the div-curl system now becomes the following Poisson's equations.

$$\begin{aligned} \Delta \boldsymbol{\eta}^x &= \frac{\partial^2 \eta^x}{\partial x^2} + \frac{\partial^2 \eta^x}{\partial y^2} = \frac{\partial f^1}{\partial x} - \frac{\partial f^2}{\partial y} = F^1 \\ \Delta \boldsymbol{\eta}^y &= \frac{\partial^2 \eta^y}{\partial x^2} + \frac{\partial^2 \eta^y}{\partial y^2} = \frac{\partial f^1}{\partial y} + \frac{\partial f^2}{\partial x} = F^2 \end{aligned} \tag{3.71}$$

Therefore, we can adopt any standard Poisson solver to solve the Eq. (3.68). In particular, We adopt the FFT based Poisson solver [41] in this work.

In 3D case, similar to the 2D case, after properly simplifying and reorganizing the Eq. (3.72) to Eq. (3.75), we can get a set of Poisson's equations as Eq. (3.76) to Eq. (3.78) shows.

$$\nabla \left(\frac{\partial \eta^x}{\partial x} + \frac{\partial \eta^y}{\partial y} + \frac{\partial \eta^z}{\partial z} \right) = \nabla f^1 \tag{3.72}$$

$$\nabla \left(\frac{\partial \eta^z}{\partial y} - \frac{\partial \eta^y}{\partial z} \right) = \nabla f^2 \tag{3.73}$$

$$\nabla\left(\frac{\partial\eta^x}{\partial z}-\frac{\partial\eta^z}{\partial x}\right)=\nabla f^3 \tag{3.74}$$

$$\nabla\left(\frac{\partial\eta^y}{\partial x}-\frac{\partial\eta^x}{\partial y}\right)=\nabla f^4 \tag{3.75}$$

$$\Delta\eta^x=\frac{\partial^2\eta^x}{\partial x^2}+\frac{\partial^2\eta^x}{\partial y^2}+\frac{\partial^2\eta^x}{\partial z^2}=\frac{\partial f^1}{\partial x}-\frac{\partial f^4}{\partial y}+\frac{\partial f^3}{\partial z}=F^1 \tag{3.76}$$

$$\Delta\eta^y=\frac{\partial^2\eta^y}{\partial x^2}+\frac{\partial^2\eta^y}{\partial y^2}+\frac{\partial^2\eta^y}{\partial z^2}=\frac{\partial f^1}{\partial y}+\frac{\partial f^4}{\partial x}-\frac{\partial f^2}{\partial z}=F^2 \tag{3.77}$$

$$\Delta\eta^z=\frac{\partial^2\eta^z}{\partial x^2}+\frac{\partial^2\eta^z}{\partial y^2}+\frac{\partial^2\eta^z}{\partial z^2}=\frac{\partial f^1}{\partial z}-\frac{\partial f^3}{\partial x}+\frac{\partial f^2}{\partial y}=F^3 \tag{3.78}$$

3.5.1.2 Revised Optimizer

Registration Algorithm Version One From Eq. (3.71), Eq. (3.22) can be revised as

$$\begin{aligned}\tilde{\boldsymbol{\eta}}^x(I)&=\Delta^{-1}(\tilde{F}^1(I))\\ \tilde{\boldsymbol{\eta}}^y(I)&=\Delta^{-1}(\tilde{F}^2(I)).\end{aligned} \tag{3.79}$$

where Δ^{-1} is a Poisson's solver.

From Eq. (3.32), Eq. (3.35), Eq. (3.19), and Eq. (3.79), we can obtain a revised $\frac{ssd}{f^n}$ as Eq. (3.80) shows, where n is 1 or 2.

$$\begin{aligned}\frac{\partial ssd(I)}{\partial f^n(I)}=\Delta^{-1}&\left(2(T(I)-R(I))(\nabla_x T(I))\frac{1}{f(I)}\right)\otimes\boldsymbol{m}^f_{1n}\\ &+\Delta^{-1}\left(2(T(I)-R(I))(\nabla_y T(I))\frac{1}{f(I)}\right)\otimes\boldsymbol{m}^f_{2n}.\end{aligned} \tag{3.80}$$

Similarly, the revised 3D version of $\dfrac{ssd}{f^n}$ can be written as

$$\begin{aligned}\frac{\partial ssd(I)}{\partial f^n(I)} &= \Delta^{-1}\left(2(T(I)-R(I))(\nabla_x T(I))\frac{1}{f(I)}\right)\otimes \boldsymbol{m}^f_{1n}\\ &+ \Delta^{-1}\left(2(T(I)-R(I))(\nabla_y T(I))\frac{1}{f(I)}\right)\otimes \boldsymbol{m}^f_{2n}\\ &+ \Delta^{-1}\left(2(T(I)-R(I))(\nabla_z T(I))\frac{1}{f(I)}\right)\otimes \boldsymbol{m}^f_{3n}.\end{aligned} \tag{3.81}$$

By using this approach, the accuracy of the gradient information is limited by the numerical precision only.

Registration Algorithm Version Two Similar to registration algorithm version one, the revised 2D optimizer for registration algorithm version two can be obtained as

$$\begin{aligned}\frac{\partial ssd(I)}{\partial f^n(I)} &= \Delta^{-1}\left(2(T(I)-R(I))(\nabla_x T(I))\right)\otimes \boldsymbol{m}^f_{1n}\\ &+ \Delta^{-1}\left(2(T(I)-R(I))(\nabla_y T(I))\right)\otimes \boldsymbol{m}^f_{2n}.\end{aligned} \tag{3.82}$$

Similarly, the revised 3D version of $\dfrac{ssd}{f^n}$ can be written as

$$\begin{aligned}\frac{\partial ssd(I)}{\partial f^n(I)} &= \Delta^{-1}\left(2(T(I)-R(I))(\nabla_x T(I))\right)\otimes \boldsymbol{m}^f_{1n}\\ &+ \Delta^{-1}\left(2(T(I)-R(I))(\nabla_y T(I))\right)\otimes \boldsymbol{m}^f_{2n}\\ &+ \Delta^{-1}\left(2(T(I)-R(I))(\nabla_z T(I))\right)\otimes \boldsymbol{m}^f_{3n}.\end{aligned} \tag{3.83}$$

CHAPTER 4

EXPERIMENTAL RESULTS

This chapter presents several experiments using several auxiliary metrics to assess the quality, robustness, and noise tolerance of our proposed nonrigid image registration algorithms.

The 2D experiments are organized as follows. Sec. 4.2 evaluates the proposed nonrigid image registration version one with two different implementations — by using LSFEM inverse filtering and by using Poisson solver. Sec. 4.3 demonstrate the capability of the image registration version one to control the lower bound of the Jacobian determinant $\det(J)$ of the deformation vector field. Sec. 4.4 compares the image registration version one and version two in terms of registration accuracy, robustness, and efficiency. Sec. 4.5 compares the registration quality of proposed nonrigid image registration version one using two different optimization strategies. Sec. 4.6 compares the viscous fluid image registration method [17] with the image registration version one. Sec. 4.7 evaluates the noise tolerance capability of the proposed registration version one and the viscous fluid image registration method. 3D experimental results are conducted in Sec. 4.8 and Sec. 4.9.

All the experiments are conducted on the Ubuntu 8.04 Linux Platform with Pentium 4 2.60 GHz and 2 GB ram. The source code is implemented in MATLAB, some critical computational parts are implemented using the mex in C or C++.

4.1 Auxiliary Metrics

4.1.1 Image Registration Quality Assessment Metrics

Sum of Squared Difference (SSD) SSD is defined as

$$SSD = \frac{1}{\|T\|} \sum_{\boldsymbol{\xi} \in \Omega} (R(\boldsymbol{\xi}) - T(\phi(\boldsymbol{\xi})))^2, \tag{4.1}$$

which quantifies the difference between the test image T and registered reference image R, where $\|T\|$ is the total number of pixels in T and ϕ is a transformation function.

Warping Index Mean warping index [42],

$$\bar{\omega} = \frac{1}{\|T\|} \sum_{\boldsymbol{\xi} \in \Omega} \|\phi(\boldsymbol{\xi}) - \phi^*(\boldsymbol{\xi})\|, \tag{4.2}$$

and maximum warping index,

$$\omega_{\max} = \max_{\boldsymbol{\xi} \in \Omega} \left(\|\phi(\boldsymbol{\xi}) - \phi^*(\boldsymbol{\xi})\| \right), \tag{4.3}$$

are used as the registration quality metrics, where ϕ is the deformation field obtained after the image registration, ϕ^* is the ground truth of the deformation field, $\boldsymbol{\xi}$ is the coordinate of a grid point, T is the test image, and $\|\phi(\boldsymbol{\xi}) - \phi^*(\boldsymbol{\xi})\|$ is the Euclidean's distance between $\phi(\boldsymbol{\xi})$ and $\phi^*(\boldsymbol{\xi})$.

Mean warping index is an appropriate metric to assess the overall quality of the registration result if the ground truth deformation field is available while the maximum warping index can indicate the largest difference between ϕ^* and ϕ and detect the convergence to a local minima.

Masked Warping Index To properly reflect the quality of the image registration results, we need to exclude the homogeneous background area in the reference image R by imposing a binary mask in the process of computing the warping index.

Therefore, the masked mean warping index $\bar{\omega}^*$ is introduced here as

$$\bar{\omega}^* = \frac{1}{\|\Omega^*\|} \sum_{\boldsymbol{\xi} \in \Omega^*} \|\phi(\boldsymbol{\xi}) - \phi^*(\boldsymbol{\xi})\|, \tag{4.4}$$

and the masked maximum warping index ω^*_{max} is defined as

$$\omega^*_{\max} = \max_{\boldsymbol{\xi} \in \Omega^*} \left(\|\phi(\boldsymbol{\xi}) - \phi^*(\boldsymbol{\xi})\|\right). \tag{4.5}$$

where Ω^* is the domain excluding the homogeneous background.

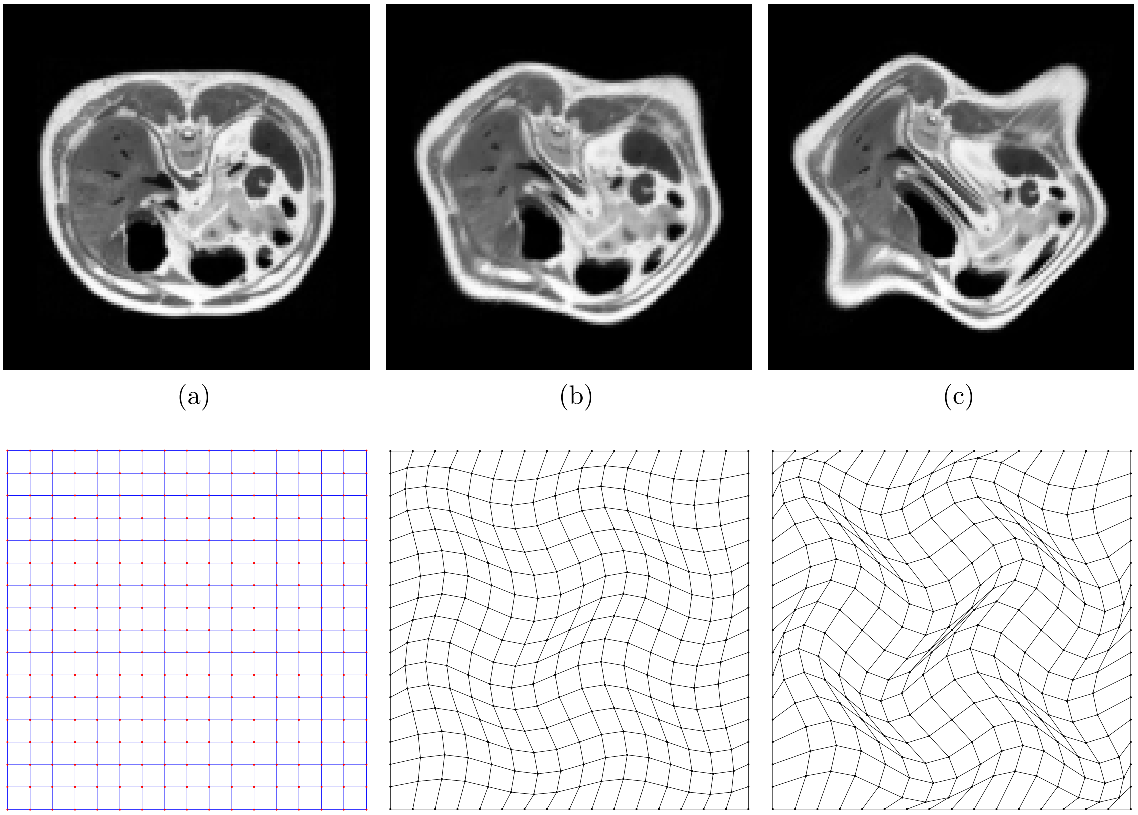
4.2 Experiment One

4.2.1 Purpose

The purpose of this experiment is to evaluate two different implementations of the proposed registration algorithm version one. First one uses the LSFEM inverse filtering and the second one uses the Poisson solver to obtain the $dSSD/df^m$, $m = 1, 2$, and solve for the intermediate vector field η mentioned in Sec. 3.2. We compared the two implementations in terms of registration accuracy, robustness, and efficiency.

4.2.2 Experimental Design

We chose the red band of a slice of the anatomical images taken from the Visible Human Project [43] (male, 70 mm data, slice 1561), then it was cropped and down-sampled bilinearly to an 129×129 test image T as Fig. 4.1a displayed. We applied the thin-plate splines principal warps method [44] with evenly distributed 17



(a) (b) (c)

(d) (e) (f)

Figure 4.1. (a) A slice of Visible Human male data used as a test image T. (b) A reference image R generated synthetically with the deformation parameter $a = 50$. (c) Synthetically generated R with $a = 100$. (d) Initial grid. (e) Deformed grid generated with $a = 50$. (f) Deformed grid generated with $a = 100$.

by 17 control points to generate a set of deformed grids as Fig. 4.1e and Fig. 4.1f show. Eq. (4.6) was applied on $\boldsymbol{x}$ to alter the position of the control points where $\boldsymbol{x} = \{x_1, x_2\}$, $x_m = \{1, 2, \ldots, 129\}$, $m = 1$ or 2, represents the original control point position and a is referred to as the deformation parameter hereafter.

$$f(x_m, a) = x_m - \frac{129 \sin(\frac{4\pi(x_m - 1)}{128}) g(x_{m^*})^{0.3}}{a}, \tag{4.6}$$

$$g(x_{m^*}) = \begin{cases} \dfrac{8(x_{m^*}-1)}{64}, & x_{m^*} \leq 65 \\ 8 - \dfrac{x_{m^*}-65}{64}, & x_{m^*} \geq 65 \end{cases}, \tag{4.7}$$

where $m = 1$ or 2, and $m \cdot m^* = 2$.

With the generated set of deformed grids, we can deform the test image and generate a series of reference images Rs. When $a = 0$, R and T are identical; when $a > 100$, R deforms too much and is tangled. Therefore, the valid value of a ranges from 0 to 100. We chose to register T to Rs with the deformation parameter a ranging from 50 to 100, which corresponds to the maximum deformation distance between R and T from 6.6391 to 13.2762 pixels and the root mean square deformation distance between R and T from 4.1949 to 8.3907 pixels.

After obtaining the Rs from T, we registered T to Rs by two different implementations of the proposed nonrigid image registration version one, one by using the LSFEM inverse filtering and the other one by the Poisson solver. To evaluate the image registration accuracy, we observed the SSD, masked mean warping index $\bar{\omega}^*$, masked maximum warping index ω^*_{max}. The selected masks used in the process of computation of $\bar{\omega}^*$ and ω^*_{max} are displayed in Fig. 4.2.

Multi-resolution Performing image registration on the coarse grid can significantly reduce the computational time, but by controlling the monitor functions and curl values on the coarse grid, the deformation method cannot express the detailed deformation field. On the contrary, using a very fine grid can register images in more detail, but it may cause the image registration procedure prone to noise interference. Therefore, in the experiments, the finest knot spacing is set up as $h = 4$. Based on our experience, we can obtain the best result if the finest knot spacing $h = 4$.

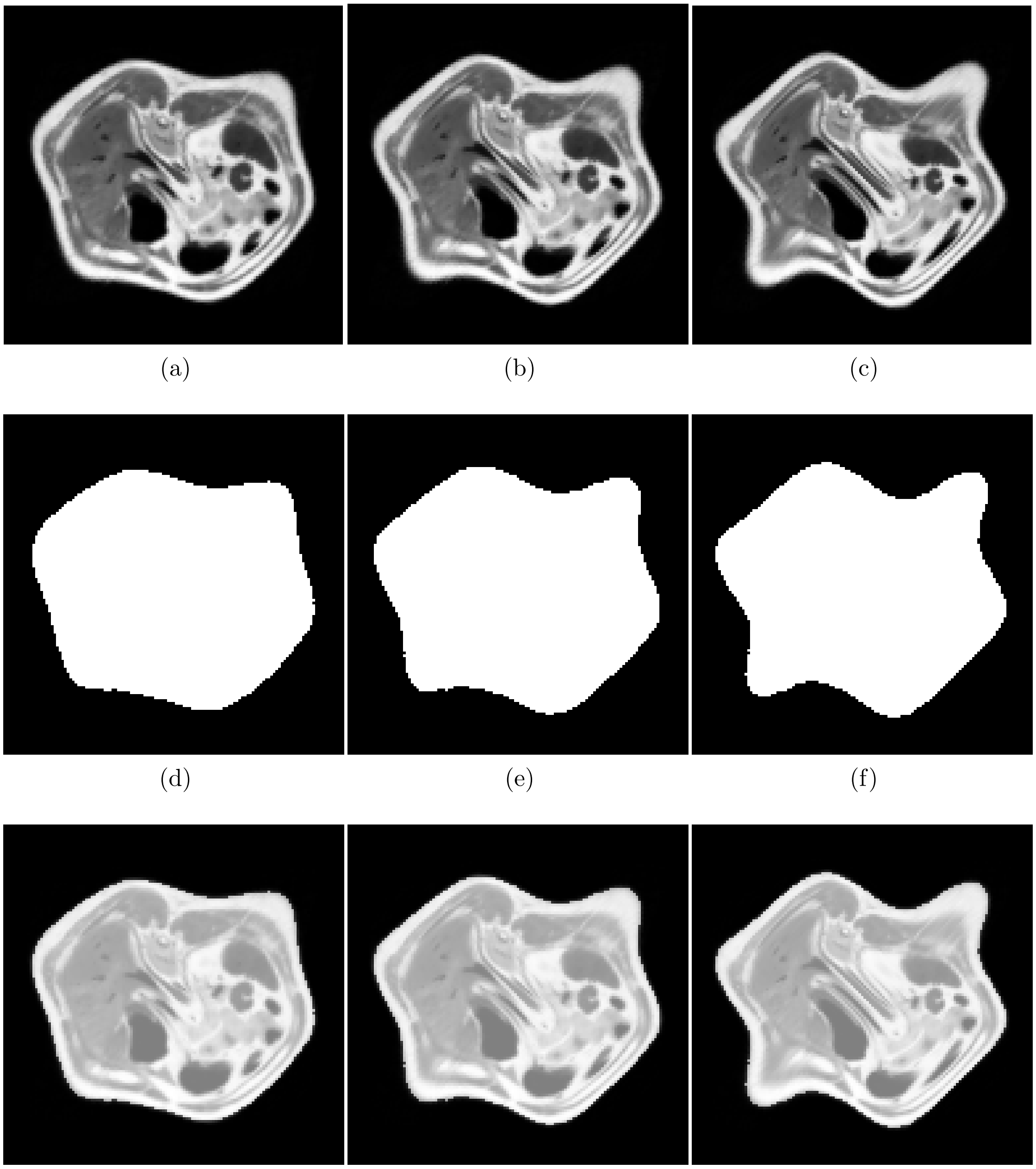


Figure 4.2. (a) to (c) are the reference images Rs synthetically generated with the deformation parameter $a = 50$, 75, and 100 respectively. (d) to (f) are masks used for masked mean warping index $\bar{\omega}^*$ and maximum masked mean warping index ω^*_{max}. (g) to (i) shows the overlapping images of (a) to (c) and (d) to (f).

There are many ways to implement the multi-resolution scheme. For the purposed algorithm, we optimize the monitor function f_h and curl $\boldsymbol{g}_h$ of the intermediate vector field $\boldsymbol{\eta}$ on the coarse grid with knot spacing h, then we interpolate f_h and $\boldsymbol{g}_h$ linearly to obtain the f_1 and $\boldsymbol{g}_1$ on the finest grid. By doing this, when h changes, there will be no abrupt fluctuation in SSD, $\bar{\omega}^*$, and $\omega^*_{\max}$.

Termination Criterion The initial step size μ used in Fig. 3.3 to adjust f^1 and f^2 is 10^{-5}. If the step size is less than 10^{-7}, we terminate the registration process.

4.2.3 Experimental Results

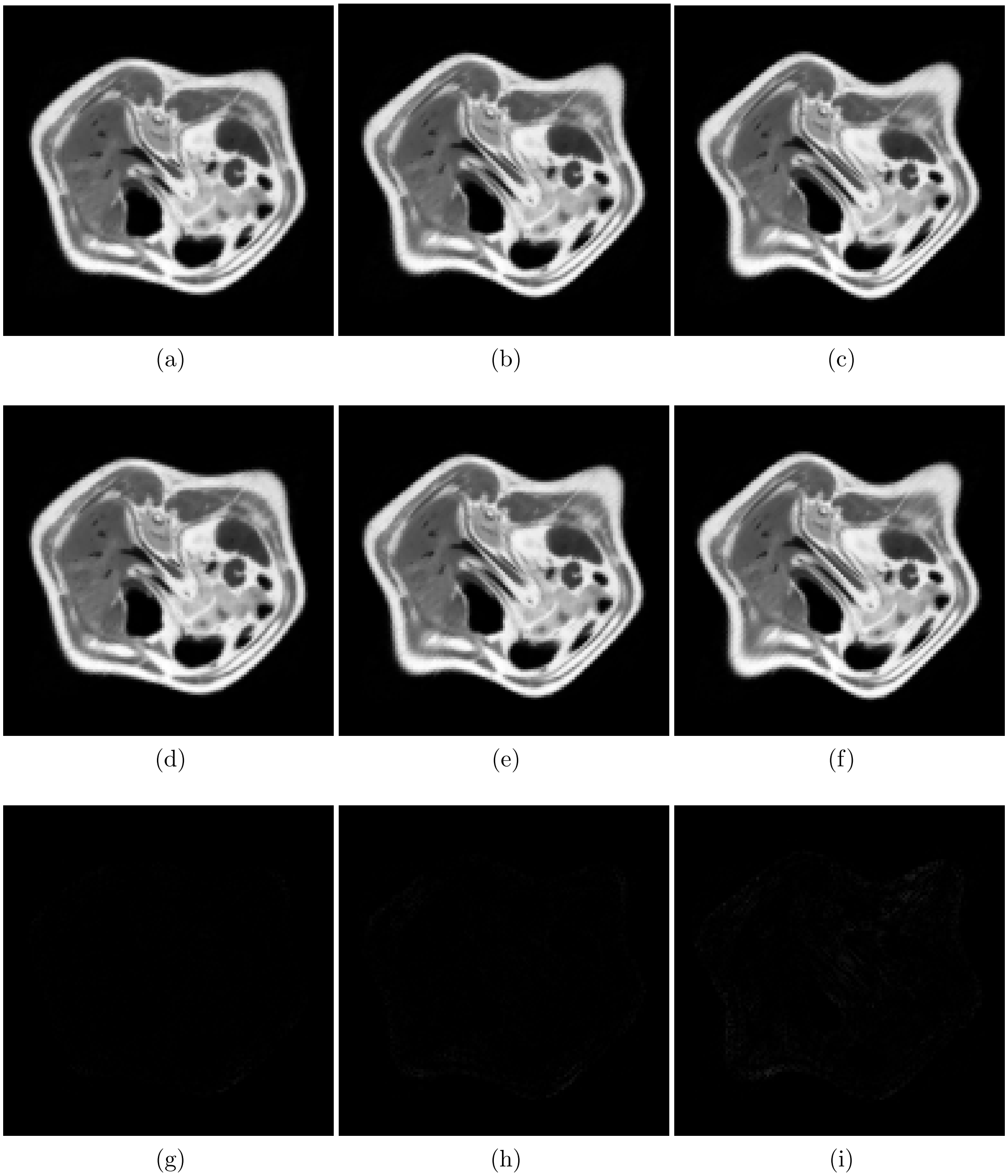
Some selected registered images and residue images (the absolute difference between R and registered T) generated by the two implementations of the nonrigid image registration version one are displayed in Fig. 4.3 and Fig. 4.4.

Robustness and Accuracy Assessment Fig. 4.5 and Fig. 4.6 illustrate the SSD versus deformation factor a, $\bar{\omega}^*$ versus a, and $\omega^*_{\max}$ versus a.

From Fig. 4.5, Fig. 4.6a, and Fig. 4.6b, we can clearly see that the registration version one with the Poisson solver outperforms the one using LSFEM inverse filtering with the 257×257 inverse filter in terms of SSD, $\bar{\omega}^*$, and ω^*_{max}.

In Fig. 4.6, the warping index values of the registration version one using Poisson solver are more stable. With Poisson solver, version one keeps the masked mean warping index $\bar{\omega}^*$ in the sub-pixel range, even when the initial transformation difference between R and T is as large as 12.6125 pixels. From Fig. 4.6b, the masked maximum warping index ω^*_{max} is under or around one pixel till $a = 87$.

Computational Performance Assessment For the computational performance comparison, from Fig. 4.7, we can observe that the total registration time of the

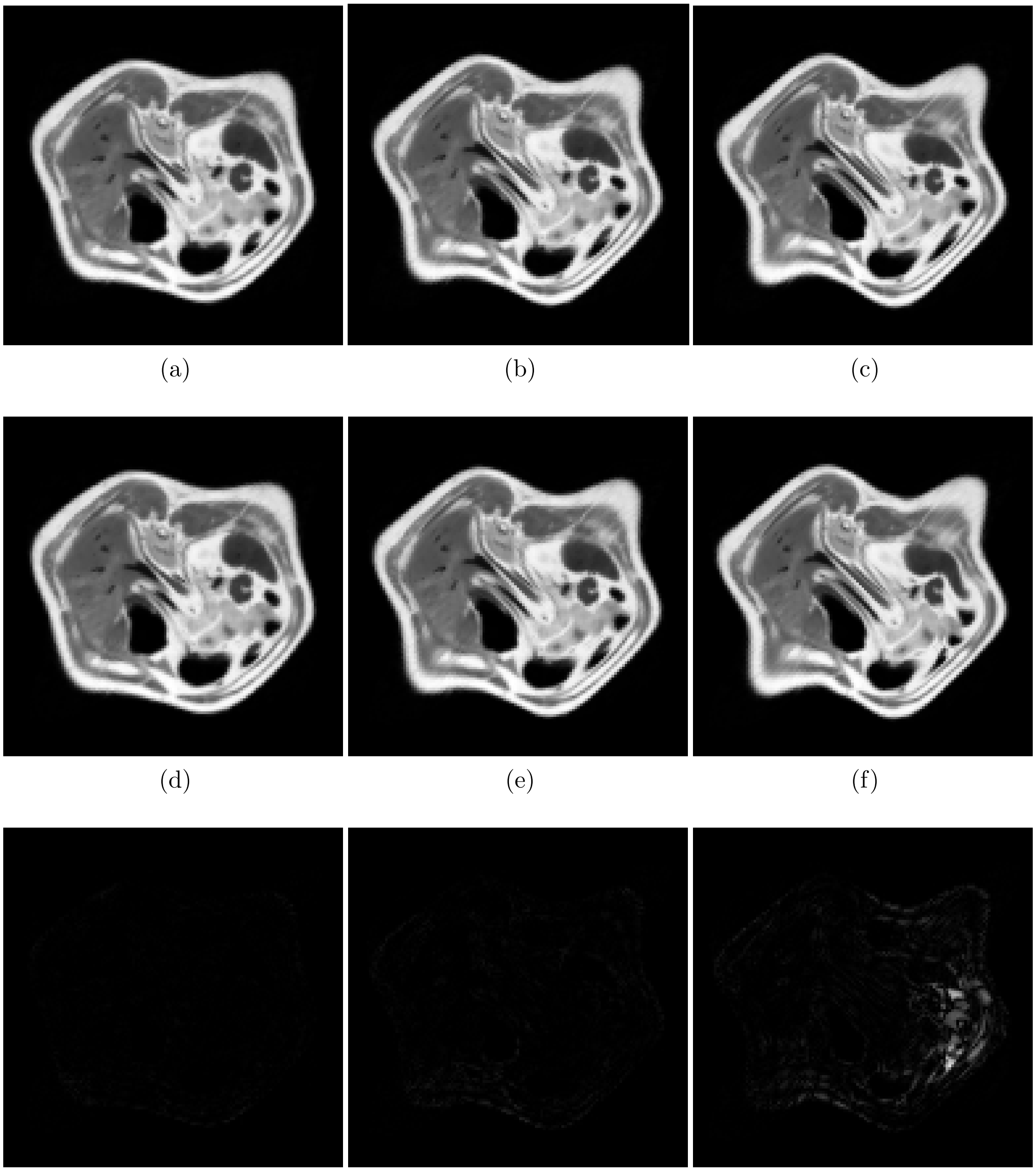


(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 4.3. (a) to (c) are the reference images Rs synthetically generated with the deformation parameter $a = 50$, 75, and 90 respectively. (d) to (f) are the test images Ts registered to Rs of $a = 50$, 75, and 90 using nonrigid image registration algorithm version one with the Poisson solver. (g) to (i) shows the residual difference between (a) to (c) and (d) to (f).



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 4.4. (a) to (c) are the reference images Rs synthetically generated with the deformation parameter $a = 50$, 75, and 90 respectively. (d) to (f) are the test images Ts registered to Rs of $a = 50$, 75, and 90 using nonrigid image registration method version one with the LSFEM inverse filtering. (g) to (i) shows the residual difference between (a) to (c) and (d) to (f).

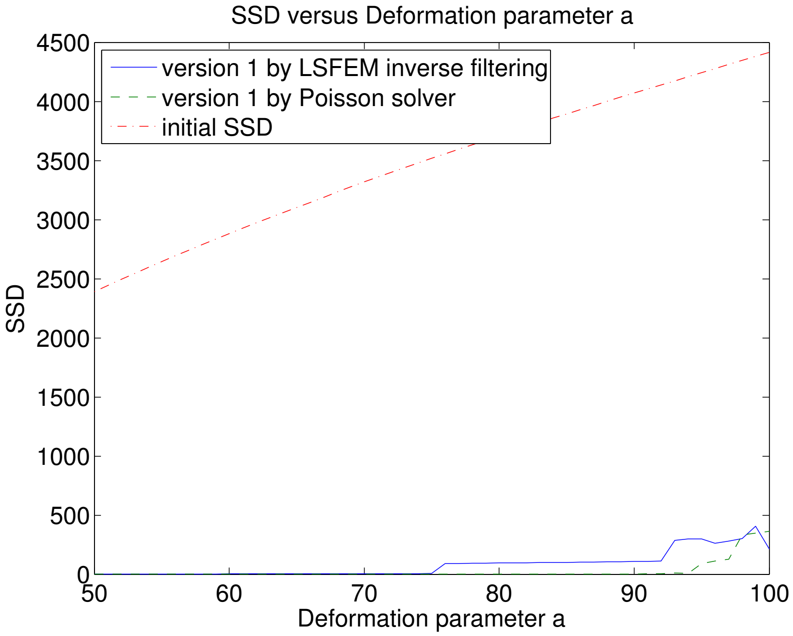
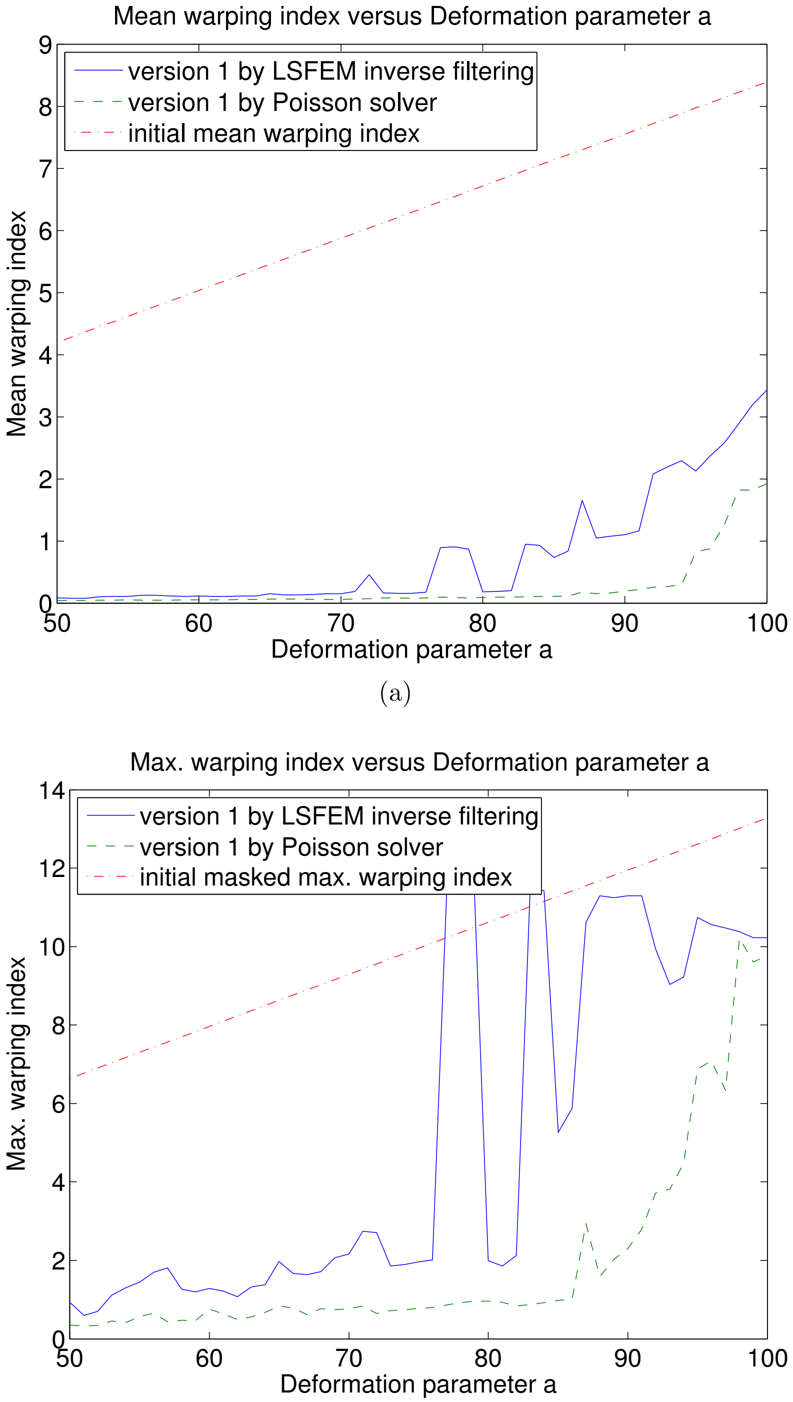


Figure 4.5. Nonrigid image registration version one SSD versus deformation parameter a experiment result. The deformation parameter a ranges from 50 to 100, which corresponds to the maximum deformation distance between R and T from 6.6391 to 13.2762 pixels.

registration version one using the Poisson solver is generally shorter than the one using the LSFEM inverse filtering method. Fig. 4.7a indicates that the registration version one with LSFEM inverse filtering is slower than the one with Poisson solver in terms of the convergence rate. The average computation time per iteration of the registration version one with Poisson solver and LSFEM inverse filtering are 0.1341 and 0.2188 second respectively.

4.2.4 Conclusion

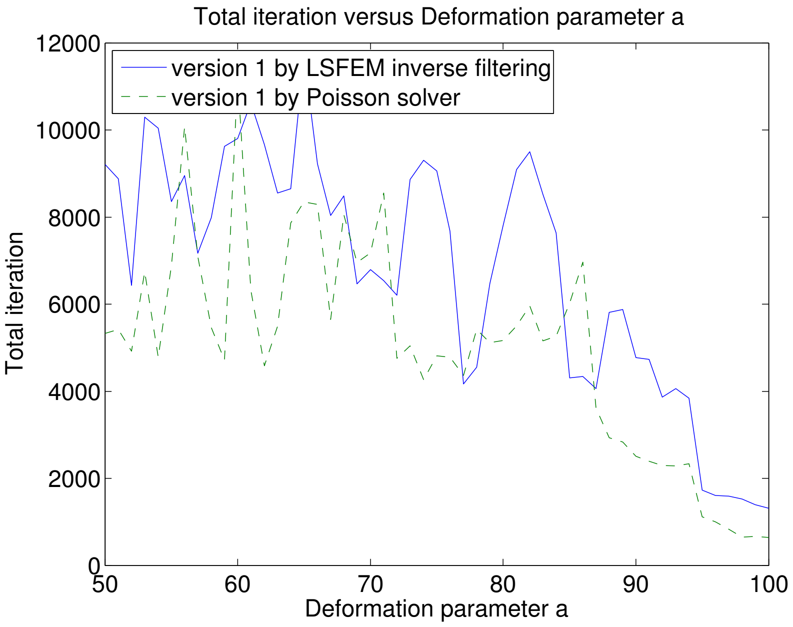
The proposed nonrigid image registration performs better by using the Poisson solver in every aspect (accuracy, robustness, and performance), which is because of the reason mentioned in Sec. 3.5 that the inverse filtering is an approximation approach and its complexity is higher.



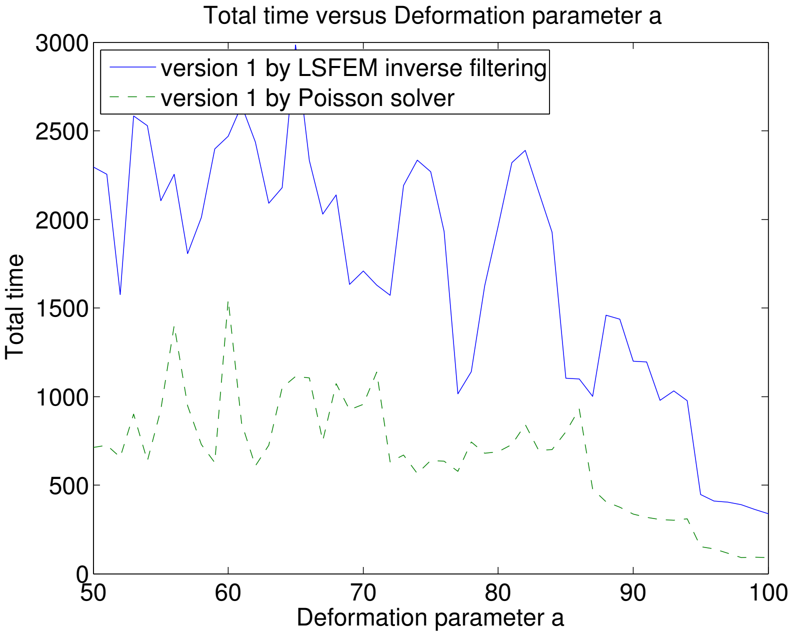
(a)

(b)

Figure 4.6. Nonrigid image registration version one experiment results. The deformation parameter a ranges from 50 to 100. (a) Masked mean warping index $\bar{\omega}^*$ versus a. (b) Masked maximum warping index $\omega^*_{\max}$ versus a.



(a)



(b)

Figure 4.7. Nonrigid image registration version one computational performance experiment results. (a) Total iteration used to register R and T versus the deformation factor a. (b) Total time of registration versus a.

4.3 Experiment Two

4.3.1 Purpose

The purpose of this experiment is to demonstrate the registration version one's ability to control the lower bound of the Jacobian determinant $\det(J)$ of the deformation vector field through setting the lower bound on the monitor function. This is the major feature of the proposed method, which guarantees no tissue folding will occur by setting a positive lower bound on the monitor function.

4.3.2 Experimental Design

We chose the reference image R ($a = 70$) and test image T generated in Sec. 4.2.2 and registered T to R by the proposed nonrigid image registration version one multiple times with different lower bound constraints enforced on $f(I)$ (0.1, 0.2, . . . , 1) to observe if we can control the minimum $\det(J)$ through imposing the lower bound on the monitor function $f(I)$.

4.3.3 Experimental Results

We imposed the lower bound constraints on the monitor function $f(I)$ (0.1, 0.2, . . . , 1) before $f(I)$ normalization. We monitored the final (when the image registration process is done) minimum monitor function f before normalization, minimum f after normalization, and the minimum $\det(J)$ when the registration process is completed.

In Fig. 4.8, minimum f after normalization is closely related to the minimum $\det(J)$ while the minimum f before normalization deviates from the minimum $\det(J)$ after the 0.5 f lower bound constraint. This is caused by the reason that $f(I)$ will change after the normalization, therefore, if we impose the lower bound constraint on $f(I)$ before normalization, the imposed value will deviate from the minimum $\det(J)$.

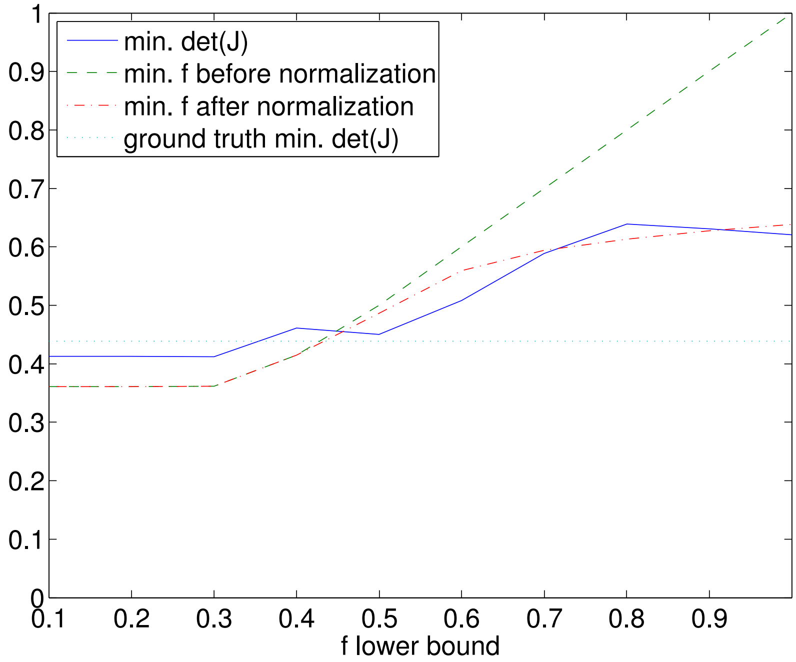


Figure 4.8. This figure displays the relationship between the minimum $\det(J)$, the minimum f before normalization ($\int_\Omega f = \|\Omega\|$), and the minimum f after normalization.

The reason of why minimum $\det(J)$ slightly deviates from the minimum f after normalization is because of the numerical error.

4.4 Experiment Three

4.4.1 Purpose

To evaluate the proposed nonrigid image registration version one and version two using the Poisson solver in terms of registration accuracy, robustness, and computational performance.

4.4.2 Experimental Design

The experiment conducted in this section adopts the same data set and experiment settings described in Sec. 4.2.2.

We chose the Rs and T generated in Sec. 4.2.2 and registered T to Rs by the proposed nonrigid image registration version one and version two. To assess the quality of registration, we observed SSD, masked mean warping index $\bar{\omega}^*$, masked maximum warping index $\omega^*_{\max}$, and the selected residue images (the difference between R and registered T) after registration, where the selected mask used in $\bar{\omega}^*$ and $\omega^*_{\max}$ is displayed in Fig. 4.2.

4.4.3 Experimental Results

Robustness and Accuracy Assessment From Fig. 4.9, Fig. 4.10a, and Fig. 4.10b, we can observe that the registration version one outperforms the version two in terms of SSD, $\bar{\omega}^*$, and ω^*_{max}. Version one keeps the masked mean warping index $\bar{\omega}^*$ in the sub-pixel range, even when the initial transformation difference between R and T is as large as 12.6125 pixels. From Fig. 4.10b, the masked maximum warping index ω^*_{max} is under or around one pixel till $a = 87$.

Computational Performance Assessment For the computational performance comparison, from Fig. 4.11, nonrigid image registration version two seems to be able

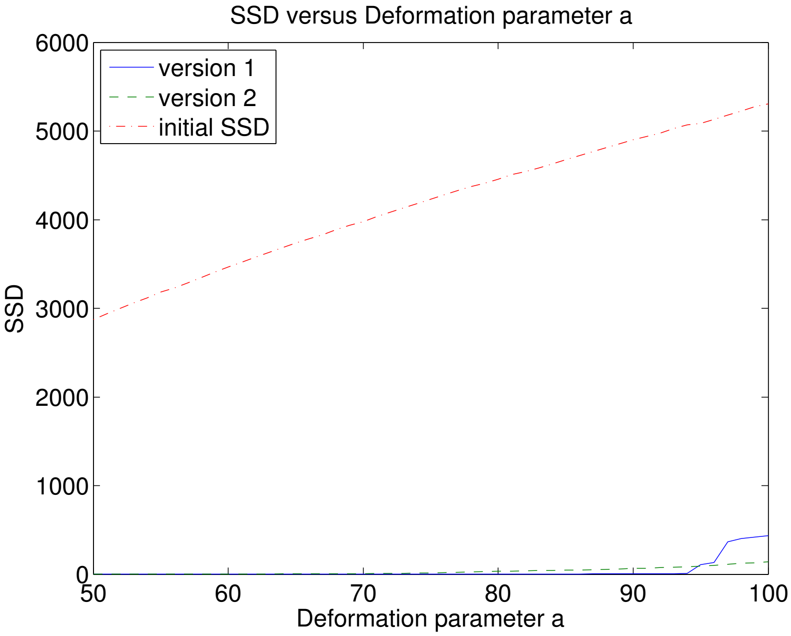
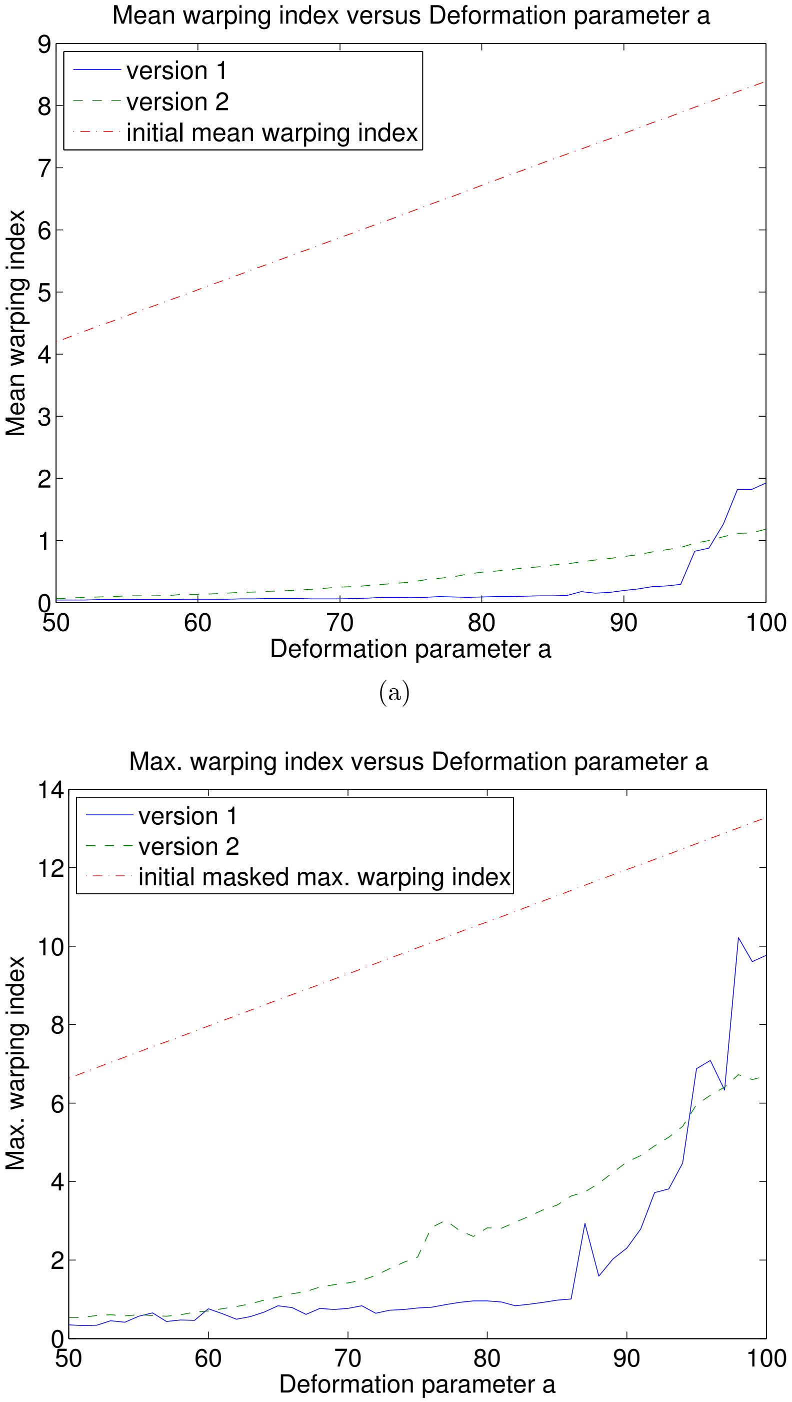


Figure 4.9. Nonrigid image registration version one versus version two experimental SSD result. The deformation parameter a ranges from 50 to 100, which means that the maximum deformation distance between R and T is ranging from 6.6391 to 13.2762 pixels.

to converge faster than version one, but after cross-referring to Fig. 4.9 and Fig. 4.10, version two converges faster is due to its limited registration accuracy. The average computation time per iteration of the registration version one and version two are 0.1341 and 0.1363 second respectively, which means that there is no significant computational complexity difference between version one and version two.

4.4.4 Conclusion

The proposed nonrigid image registration version one generally can register more accurately and is more robust than version two. It is certainly an interesting question that why the registration version two does not work as well version one, which will be one of our future objects.



(a)

(b)

Figure 4.10. Nonrigid image registration version one versus version two experimental warping index result. The deformation parameter a ranges from 50 to 100. (a) Masked mean warping index $\bar{\omega}^*$ versus a. (b) Masked maximum warping index $\omega^*_{\max}$ versus a.

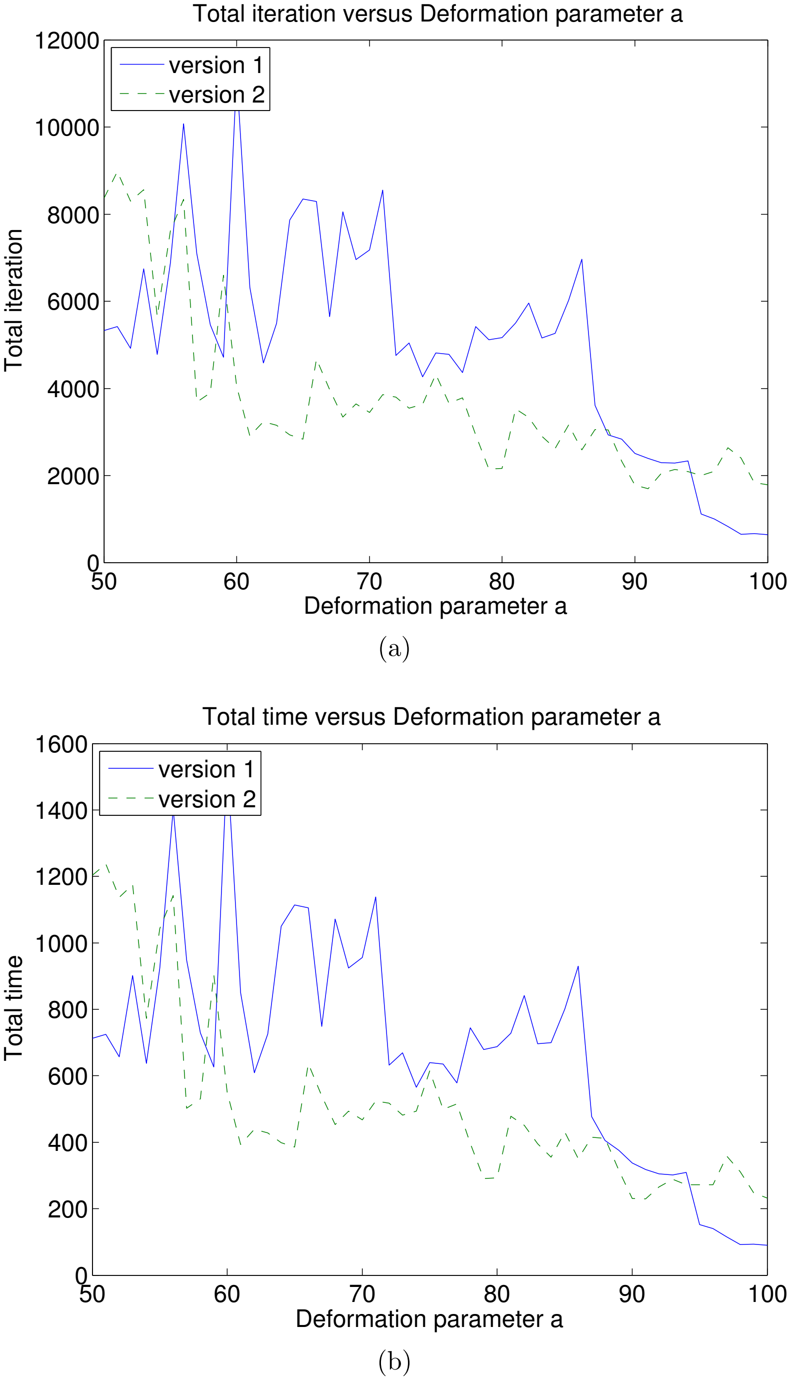


Figure 4.11. Nonrigid image registration version one versus version two computational performance experiment result. (a) Total iteration used to register R and T versus the deformation factor a. (b) Total time of registration versus a.

4.5 Experiment Four

4.5.1 Purpose

This experiment evaluates the nonrigid image registration version one adopting two optimization strategies, one updates the parameters f^1 and f^2 simultaneously (previous experiments were conducted with this strategy), the other one updates the f^1 and f^2 alternatively.

Since f^1 and f^2 are independent of each other, we assume that by adjusting f^1 and f^2 alternatively, we may obtain better registration results. Therefore, to evaluate the registration accuracy and robustness of the proposed nonrigid image registration version one using different strategies, we conducted this experiment.

4.5.2 Experimental Design

The reference images Rs and test image T used in this experiment are the same as the ones used in Sec. 4.2.2. We register T to Rs by optimizing monitor function f^1 and curl value f^2 alternatively and observe if the registration quality and robustness obtained this way is better than the ones obtained by optimizing f^1 and f^2 simultaneously.

4.5.3 Experimental Results

Robustness and Accuracy Assessment From Fig. 4.12 and Fig. 4.13, we observed that the nonrigid image registration version one optimizing f^1 and f^2 alternatively generally has the same SSD, masked mean warping index $\bar{\omega}^*$, and masked maximum warping index $\omega^*_{\max}$ when $a < 87$. When $a \geq 87$, optimizing f^1 and f^2 alternatively outperforms optimizing the two parameters simultaneously in every aspects. Notice the four abrupt spikes in Fig. 4.13, registration version one by optimizing f^1

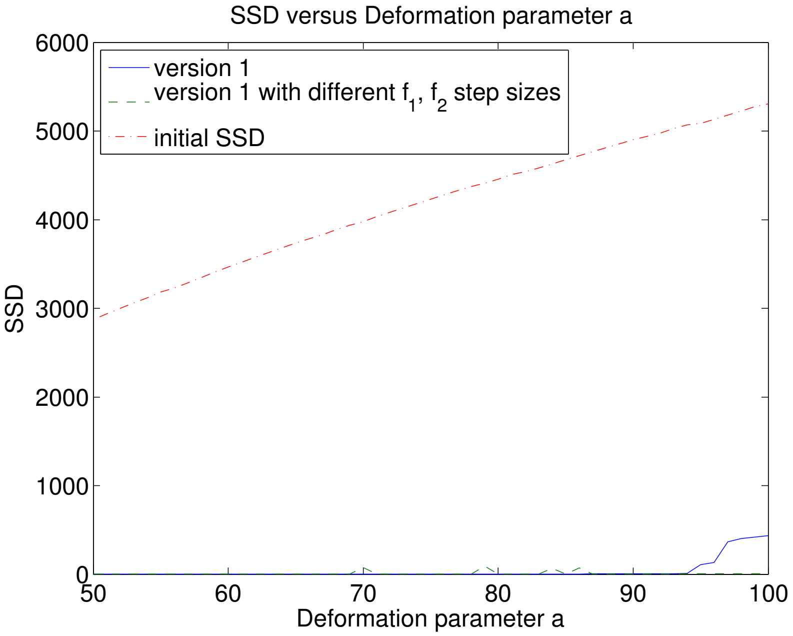
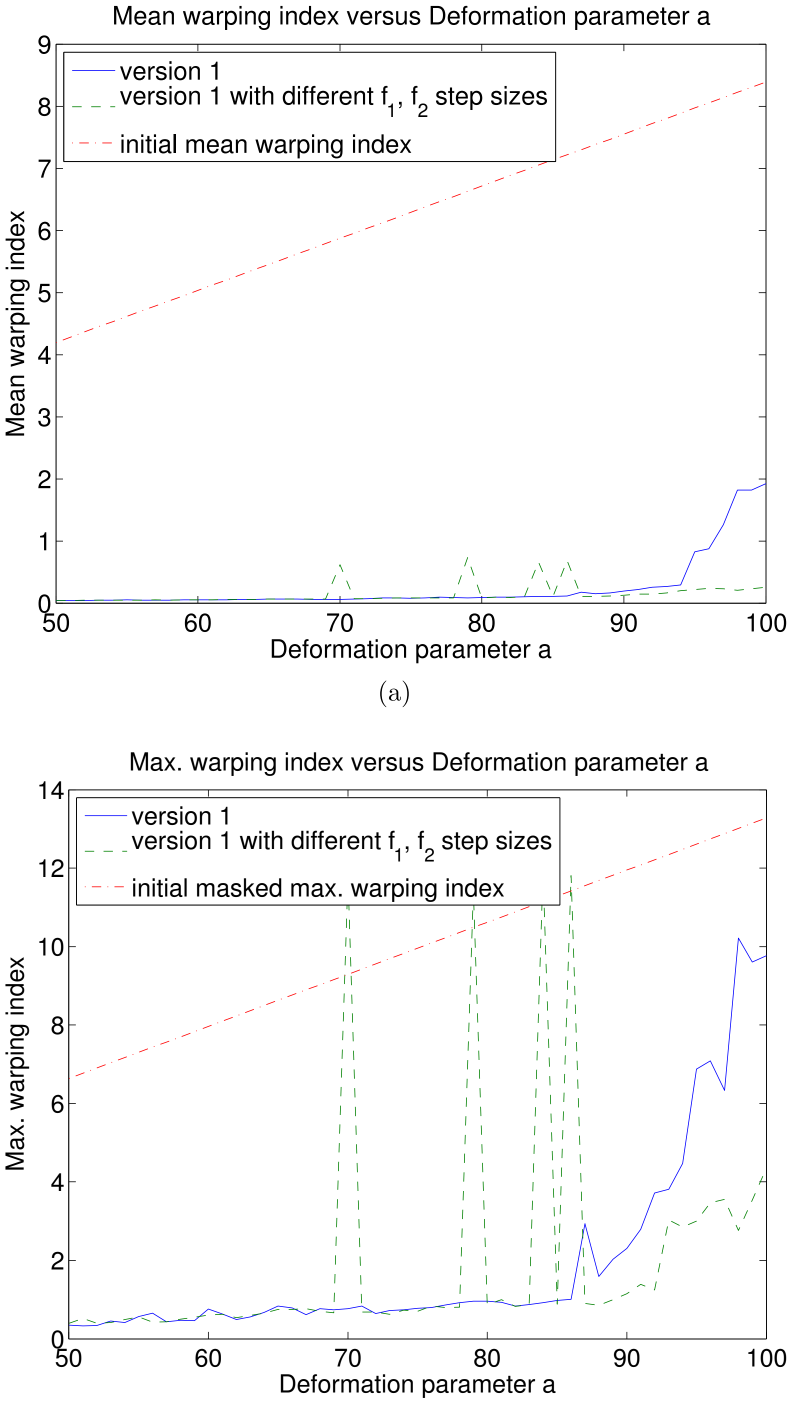


Figure 4.12. SSD versus deformation parameter a experiment result by nonrigid image registration version one with two different optimization approaches, one using the same step size to adjust f^1 and f^2, the other one using different independent step sizes to adjust f^1 and f^2. The deformation parameter a ranges from 50 to 100, which means that the maximum deformation distance between R and T is ranging from 6.6391 to 13.2762 pixels.

and f^2 alternatively seems to be prone to some local minima in the optimization process.



(a)

(b)

Figure 4.13. Nonrigid image registration version one with two different optimization approaches. The deformation parameter a ranges from 50 to 100. (a) Masked mean warping index $\bar{\omega}^*$ versus a. (b) Masked maximum warping index $\omega^*_{\max}$ versus a.

4.6 Experiment Five

4.6.1 Purpose

This experiment compares the viscous fluid image registration method [17] with the proposed nonrigid image registration version one in terms of registration accuracy, robustness, and efficiency.

4.6.2 Viscous Fluid Registration

Elastic model based transformation develops restoring forces proportional to the deformed distance. Such elastic model prevents large deformation. Christensen [17] proposed the fluid model to overcome this limitation by allowing the restoring forces to relax overtime.

To deform and register the test image to the study image, Christensen uses the Navier-Stokes equation

$$\mu\nabla^2\vec{v} + (\lambda+\mu)\vec{\nabla}(\vec{\nabla}\cdot\vec{v}) + \vec{b}(\vec{u}) = 0 \tag{4.8}$$

, a partial derivative equation (PDE), to express the velocity vector field $\vec{v}$ of the test image. $\vec{u}$ represents the displacement of a test image The first term $\mu\nabla^2\vec{v}$ is associated with constant volume viscous flow of the template, which means, it represents the viscous term of the PDE. This term constrains neighboring particles of the displacement field to deform with roughly the same velocity by spatially smoothing the velocity field. The second term $(\lambda+\mu)\vec{\nabla}(\vec{\nabla}\cdot\vec{v})$ is not zero when the regions of template grow or dissolve (mass source term). Coefficients μ and λ are viscosity coefficients. The body force $\vec{b}(\vec{u})$ can be written as

$$\vec{b}[\vec{x},\vec{u}(\vec{x},t)] = -\alpha(T[\vec{x}-\vec{u}(\vec{x},t)] - S(\vec{x}))\vec{\nabla}T|_{\vec{x}-\vec{u}(\vec{x},t)} \tag{4.9}$$

which is **∇SSD** (SSD (Sum of Squared Difference) is considered as the potential energy).

For each iteration, solve the $\vec{v}$ from (4.8) directly or by successive over relaxation (SOR), then do the explicit Euler integration to get $\vec{u}$. Iteration stops when $\vec{b}(\vec{u})$ is below a threshold or the maximum number of iteration is reached.

4.6.3 Experimental Design

The experiment conducted in this section adopts the same data set and experiment settings described in Sec. 4.2.2.

We took the Rs and T generated in Sec. 4.2.2 and registered T to Rs by the proposed nonrigid image registration version one and viscous fluid registration method ($\lambda = 0$, $\mu = 500$). To assess the quality of registration, we observed SSD, masked mean warping index $\bar{\omega}^*$, masked maximum warping index $\omega^*_{\max}$, and the selected residue images (the difference between R and registered T) after registration, where the selected mask used in $\bar{\omega}^*$ and $\omega^*_{\max}$ is displayed in Fig. 4.2.

4.6.4 Experimental Results

Robustness and Accuracy Assessment From Fig. 4.14, Fig. 4.15a, and Fig. 4.15b, we can observe that the registration version one outperforms the viscous fluid image registration method in terms of SSD, $\bar{\omega}^*$, and ω^*_{max}. Version one keeps the masked mean warping index $\bar{\omega}^*$ in the sub-pixel range, even when the initial transformation difference between R and T is as large as 12.6125 pixels. From Fig. 4.15b, the masked maximum warping index ω^*_{max} is under or around one pixel till $a = 87$.

Performance Assessment The average computation time per iteration of the registration version one and viscous fluid image registration method are 0.1341 and

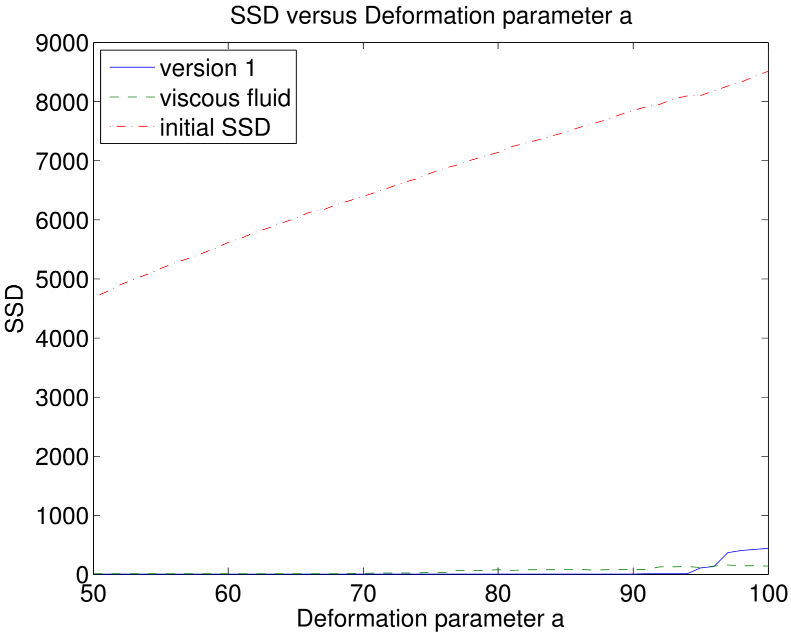
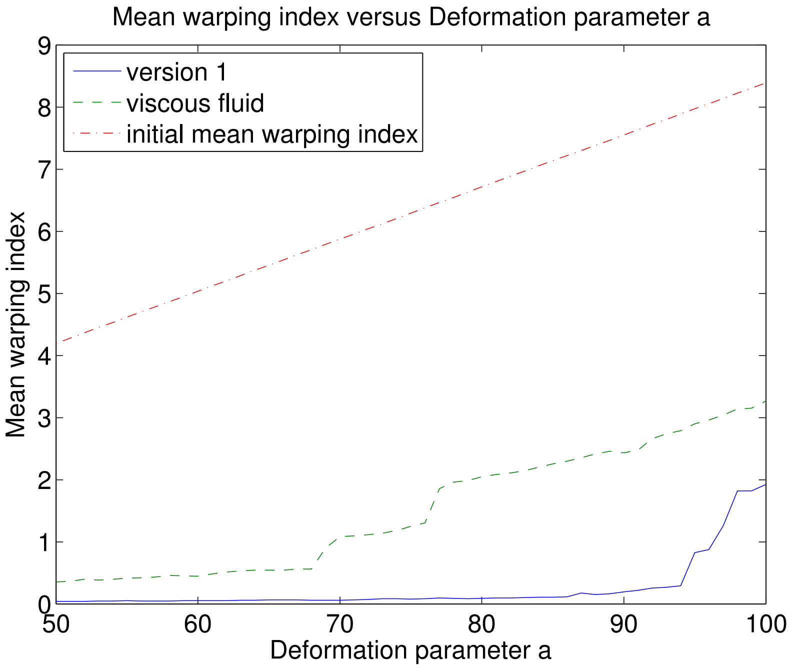


Figure 4.14. Nonrigid image registration version one versus viscous fluid registration method SSD experiment result. The deformation parameter a ranges from 50 to 100, which corresponds to the maximum deformation distance between R and T from 6.6391 to 13.2762 pixels.

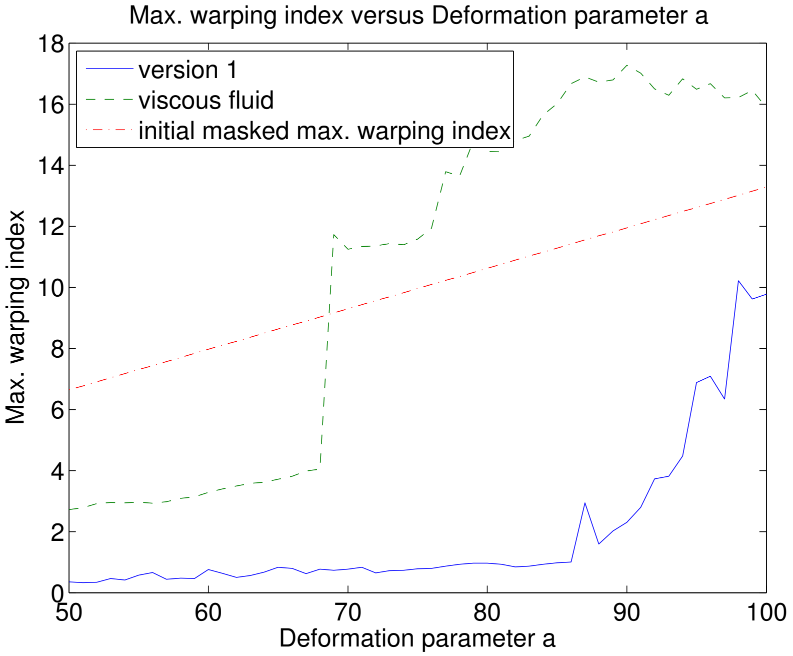
0.2580 second respectively. This is due to viscous fluid method uses SOR as its modified Navier-Stokes equation solver and the complexity is $N^{1.5}$ [45] (N: total number of image pixels) while the proposed nonrigid image registration version one using FFT based Poisson solver has the $N \log N$ [45] computation complexity.

4.6.5 Conclusion

The proposed nonrigid image registration method is more robust and more accurate than the viscous fluid method.

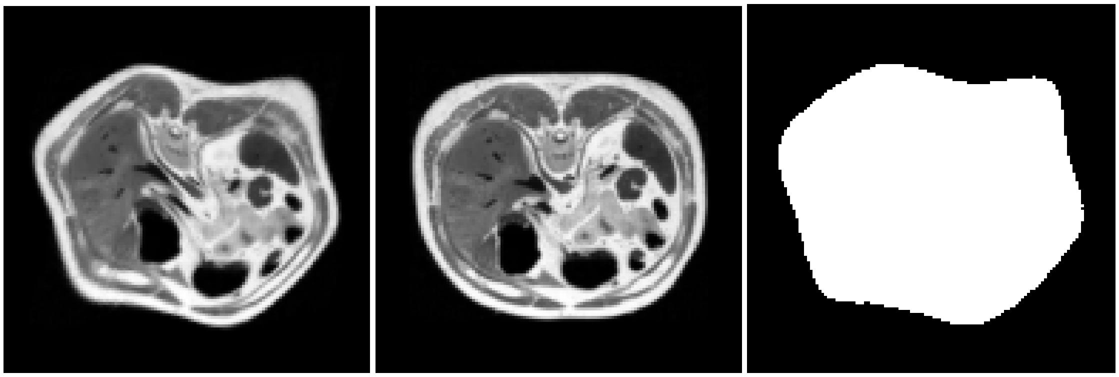


(a)



(b)

Figure 4.15. Nonrigid image registration version one versus viscous fluid registration method warping index experiment result. The deformation parameter a ranges from 50 to 100. (a) Masked mean warping index $\bar{\omega}^*$ versus a. (b) Masked maximum warping index $\omega^*_{\max}$ versus a.



(a) (b) (c)

Figure 4.16. (a) The reference images R synthetically generated with the deformation parameter $a = 50$. (b) The test image T without imposed noise. (c) The mask applied on masked mean warping index $\bar{\omega}^*$ and maximum masked mean warping index ω^*_{max}.

4.7 Experiment Six

4.7.1 Purpose

This experiment compares the noise tolerance capability of the proposed non-rigid image registration version one with the viscous fluid registration method.

4.7.2 Experimental Design

To evaluate the influence of different levels of signal to noise ratio (SNR) on the registration quality, we chose the test image T (Fig. 4.16b) and one of the reference image R (Fig. 4.16a, $a = 50$) generated in Sec. 4.2.2 and imposed different level of Gaussian noises (SNR = 30, 20, 10, and 0 dB) to the test image T. The mask used in $\bar{\omega}^*$ and ω^*_{max} is displayed in Fig. 4.16c. The SNR of an image [46, 47] is defined as

$$\text{SNR} = 20 \log_{10} \left(\frac{I_{max} - I_{min}}{s_n} \right) \text{dB}, \tag{4.10}$$

where I_{max} and I_{min} are the maximum and minimum intensity of the image, s_n is the standard deviation of the noise.

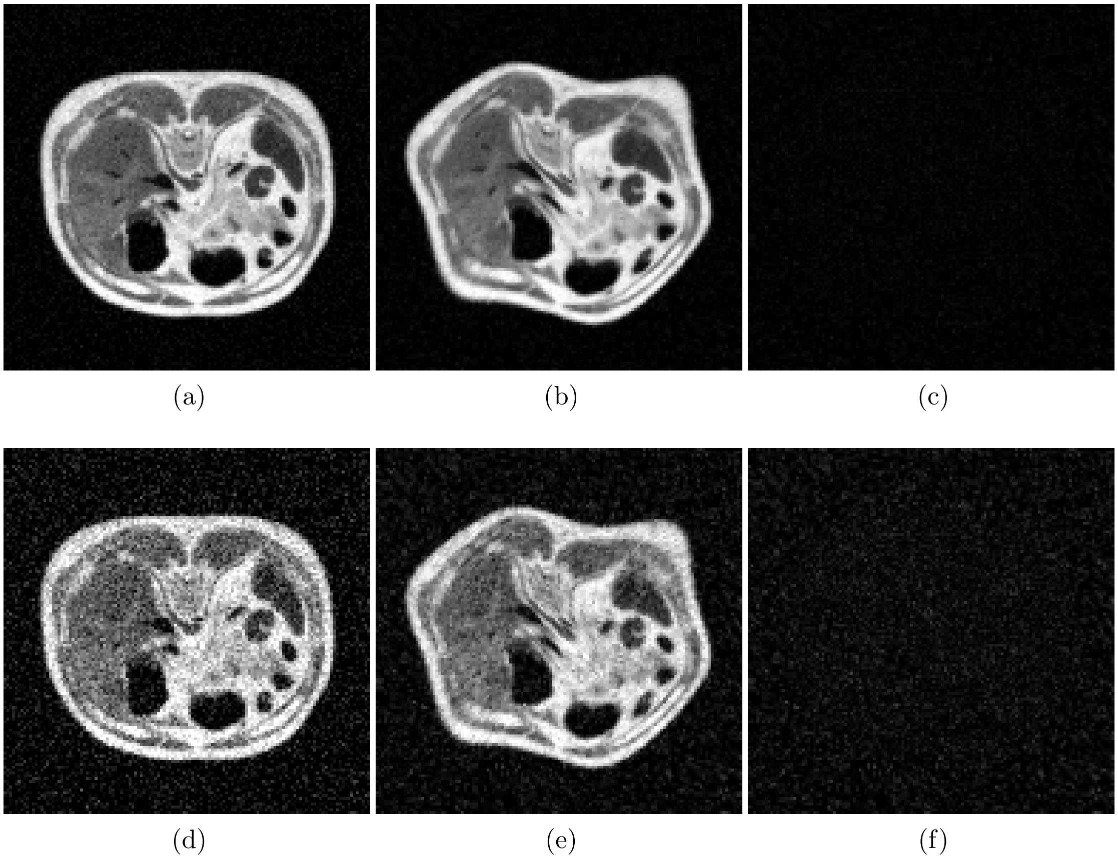
4.7.3 Experimental Results

Fig. 4.17 and Fig. 4.18 display the test images Ts imposed with SNR = 30, 20, 10, and 0 dB Gaussian noise, the register test images, and the residue images obtained by applying the proposed registration algorithm version one. We can clearly observe from the residue images that under SNR 30 to 10 Gaussian noise, there are only noise residues in the residual images, which indicates that the image registration procedure is successful. The registered images are registered properly.

Fig. 4.19 and Fig. 4.20 display the Ts imposed with SNR = 30, 20, 10, and 0 dB Gaussian noise, the register test images, and the residue images obtained by applying the viscous fluid nonrigid image registration method. From the residual images, the noise reduced significantly, which indicates that the image registration is not properly done.

Fig. 4.21 and Fig. 4.22 illustrate the SSD, $\bar{\omega}^*$, and $\omega^*_{\max}$ of the registration results versus different SNR levels. From the SSD point of view, the viscous fluid method seems to outperform the proposed registration algorithm version one, but this result is misleading. If we further observe the warping index experimental results listed in Fig. 4.22, we will find out that actually it is the proposed registration algorithm version one that can register with better accuracy under SNR = 30, 20, and 10 dB noise (the $\bar{\omega}^*$ before registration is 4.1949 pixels, the $\omega^*_{\max}$ before registration is 6.6393 pixels).

With the help of the warping index metric, we can conclude that the proposed nonrigid image registration method can register images fairly accurately under SNR = 30, 20, and 10 dB noise.


(a)

(b)

(c)

(d)

(e)

(f)

Figure 4.17. Registered test image and residue images obtained by the registration version one under SNR = 30, 20 dB noises. (a) Test image T applied with 30 dB SNR Gaussian noise. (b) Registered 30 dB SNR test image T^*_{30dB}. (c) Residue image of R and T^*_{30dB}. (d) Test image T applied with 20 dB SNR Gaussian noise. (e) Registered 20 dB SNR test image T^*_{20dB}. (f) Residue image of R and T^*_{20dB}.

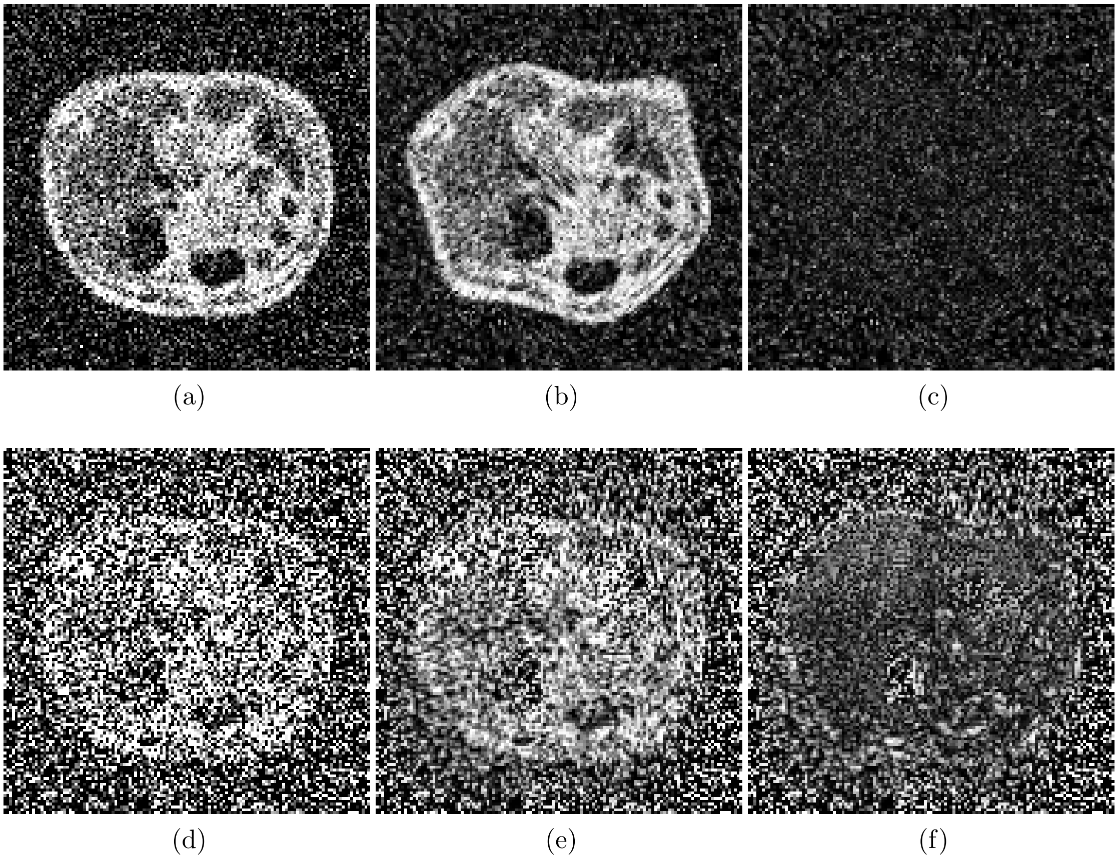


Figure 4.18. Registered test images and residue images obtained by the registration version one under SNR = 10, 0 dB noises. (a) Template image T applied with 10 dB SNR Gaussian noise. (b) Registered 10 dB SNR test image T^*_{10dB}. (c) Residue image of R and T^*_{10dB}. (d) Template image T applied with 0 dB SNR Gaussian noise. (e) Registered 0 dB SNR test image T^*_{0dB}. (f) Residue image of R and T^*_{0dB}.

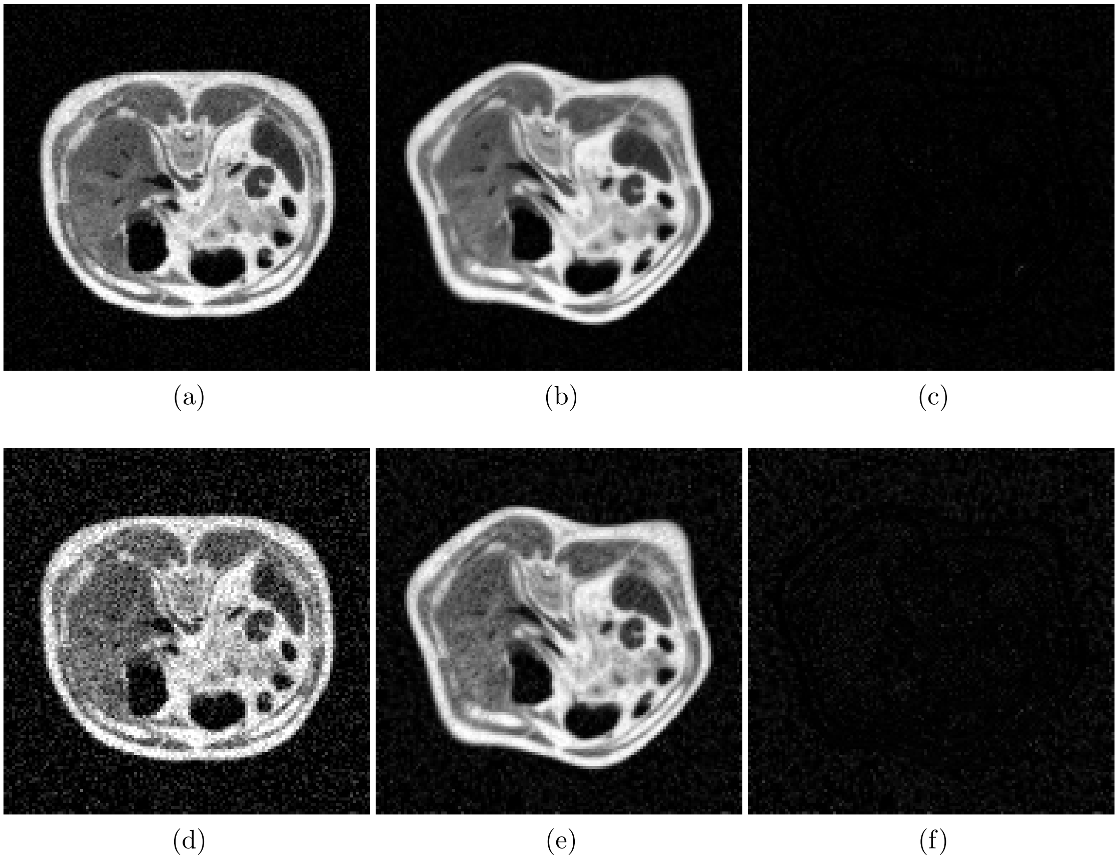


Figure 4.19. Registered test images and residue images obtained by the viscous registration method under SNR = 30, 20 dB noises. (a) Template image T applied with 30 dB SNR Gaussian noise. (b) Registered 30 dB SNR test image T^*_{30dB}. (c) Residue image of R and T^*_{30dB}. (d) Template image T applied with 20 dB SNR Gaussian noise. (e) Registered 20 dB SNR test image T^*_{20dB}. (f) Residue image of R and T^*_{20dB}.

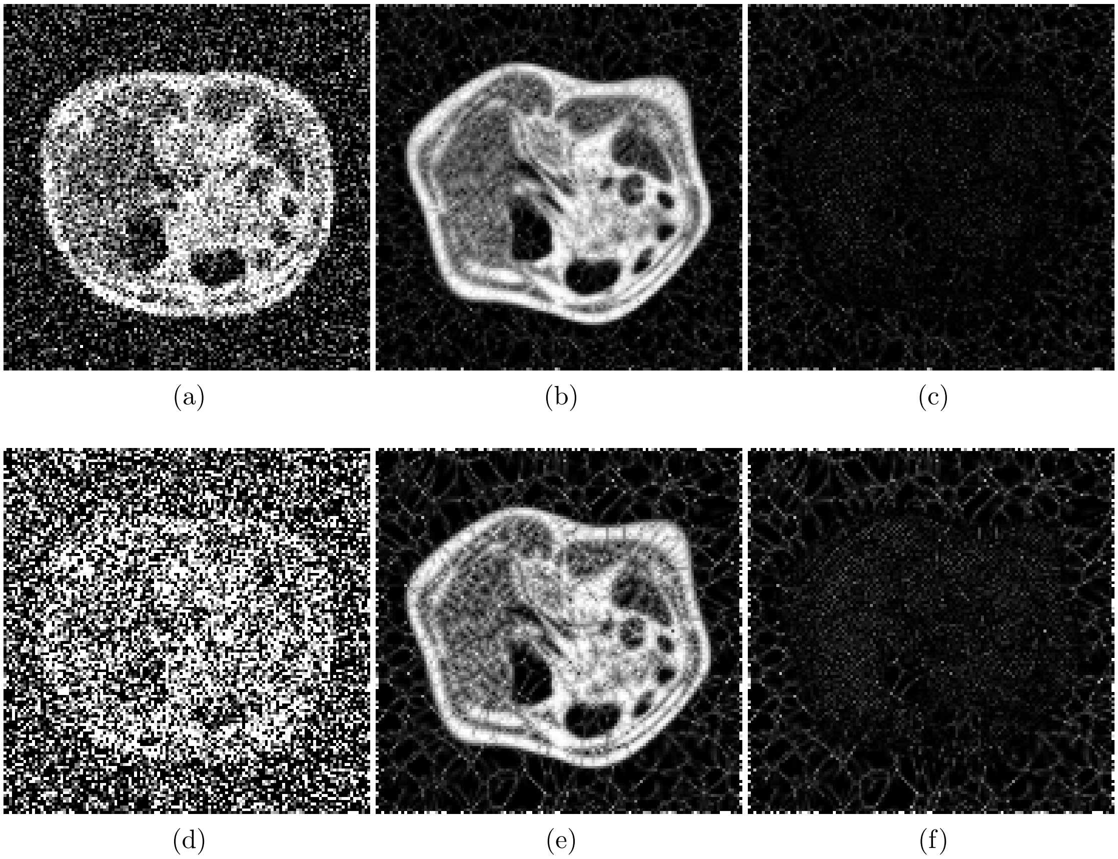


Figure 4.20. Registered test images and residue images obtained by the viscous fluid registration method under SNR = 10, 0 dB noises. (a) Template image T applied with 10 dB SNR Gaussian noise. (b) Registered 10 dB SNR test image T^*_{10dB}. (c) Residue image of R and T^*_{10dB}. (d) Template image T applied with 0 dB SNR Gaussian noise. (e) Registered 0 dB SNR test image T^*_{0dB}. (f) Residue image of R and T^*_{0dB}.

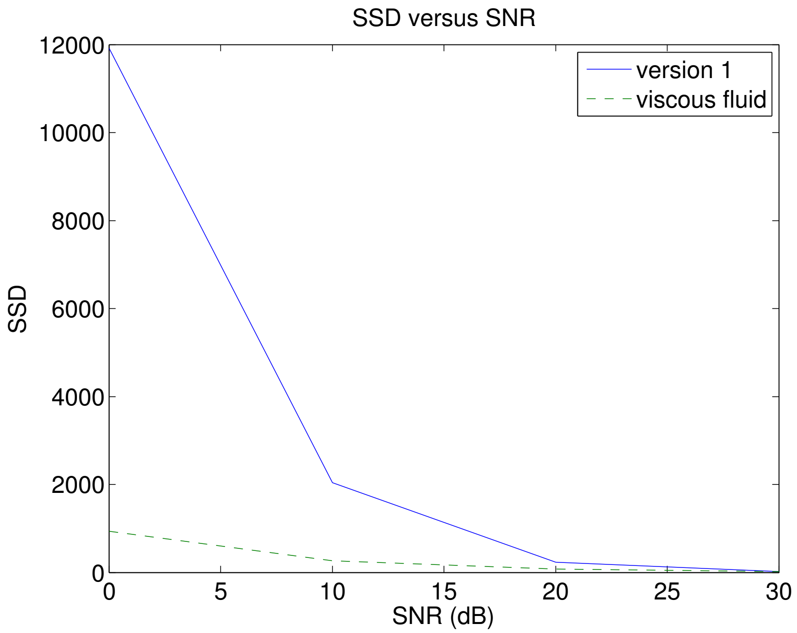
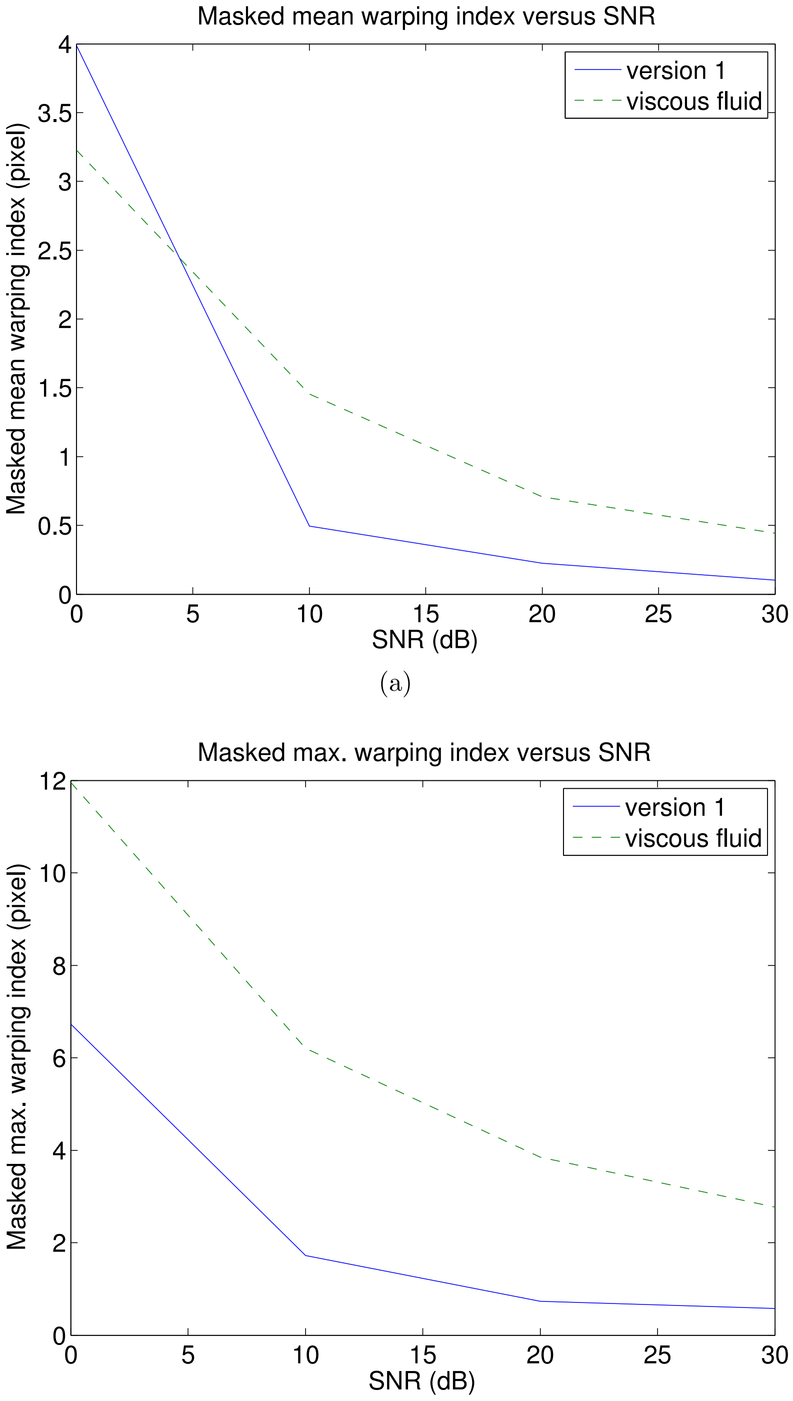


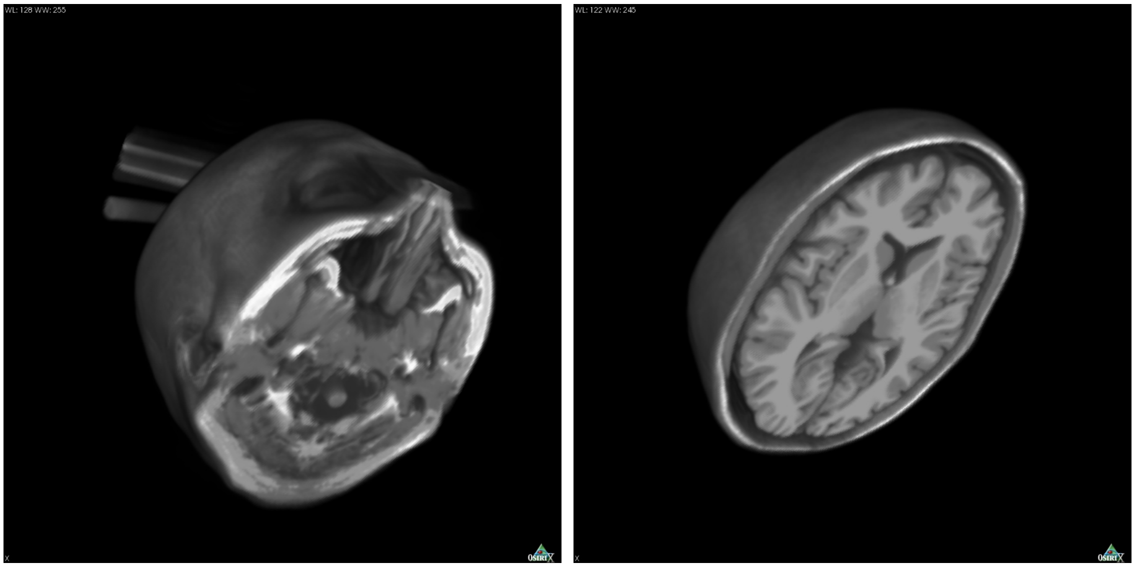
Figure 4.21. Nonrigid image registration version one versus viscous fluid noise tolerance SSD experiment result.



(a)

(b)

Figure 4.22. Nonrigid image registration version one versus viscous fluid method noise tolerance warping index experiment results. (a) Masked mean warping index $\bar{\omega}^*$ versus SNR. (b) Masked maximum warping index $\omega^*_{\max}$ versus SNR.



(a) (b)

Figure 4.23. (a) Original brain MRI image. (b) The cropped brain MRI image used as the test image.

4.8 Experiment Seven

4.8.1 Purpose

The purpose of this experiment is to compare the registration quality and efficiency of two proposed implementations of the 3D nonrigid image registration version one — one optimizes f^1 to f^4, the other one optimizes f^1 to f^3 where $f^1 =$ monitor function $- 1$ and f^2 to f^4 are three curl components.

4.8.2 Experimental Design

3D Data Preparation We adopted a simulated T1 3D 181 × 217 × 181 brain MRI image with 0% noise from the BrainWeb[48, 49, 50, 51, 52], the simulated brain database, of the McConnell Brain Imaging Centre as our experiment material displayed in Fig. 4.23a. We segmented the 3D image and took the 55 to 105 slices as Fig. 4.23b shows, then each of the retrieved slices was resized into a 65 × 65 image.

We then padded this image with five layers zeros above and eight layers zeros below in the z direction so that it can be a $65 \times 65 \times 65$ image.

This processed image served as our test image T. Similar to the procedure described in Sec. 4.2.2, we warped the test image T by applying the thin-plate splines principal warps method with $7 \times 7 \times 7$ control points to generate a reference image R. The positions $\boldsymbol{x} = \{x_1, x_2, x_3\}$, $x_m = \{1, 2, \ldots, 65\}$, $m = 1$, 2, or 3, of the control points are determined by the transformation function $f(x_m, a)$ defined in Eq. 4.11, where a is referred to as the deformation factor.

$$
\begin{aligned}
f(x_m, a) &= x_m - \frac{65\sin(\frac{4\pi(x_1-1)}{64})g(x_2)^{0.3}}{a} \cdot \frac{65\sin(\frac{4\pi(x_1-1)}{64})g(x_3)^{0.3}}{a}, \quad m = 1,2,3 \\
g(x_{m'}) &= \begin{cases} \dfrac{x_{m'}-1}{10}, & x_{m'} \leq 33 \\ 3 - \dfrac{x_{m'}-33}{32}, & x_{m'} \geq 33 \end{cases}, \quad m' = 2,3
\end{aligned}
\tag{4.11}
$$

4.8.3 Experiment Results

Figure 4.24 displays the selective slices of the reference image R, registered test image using nonrigid image registration version one that optimizes four parameters f^1 to f^4, and the residual difference between the them. The result is quite accurate, there is almost no difference between the registered T and R images.

Figure 4.25 displays the selective slices of the reference image R, registered test image using nonrigid image registration version one that optimizes three parameters f^1 to f^3, and the residual difference between the them. It does not register correctly, the residue is very obvious.

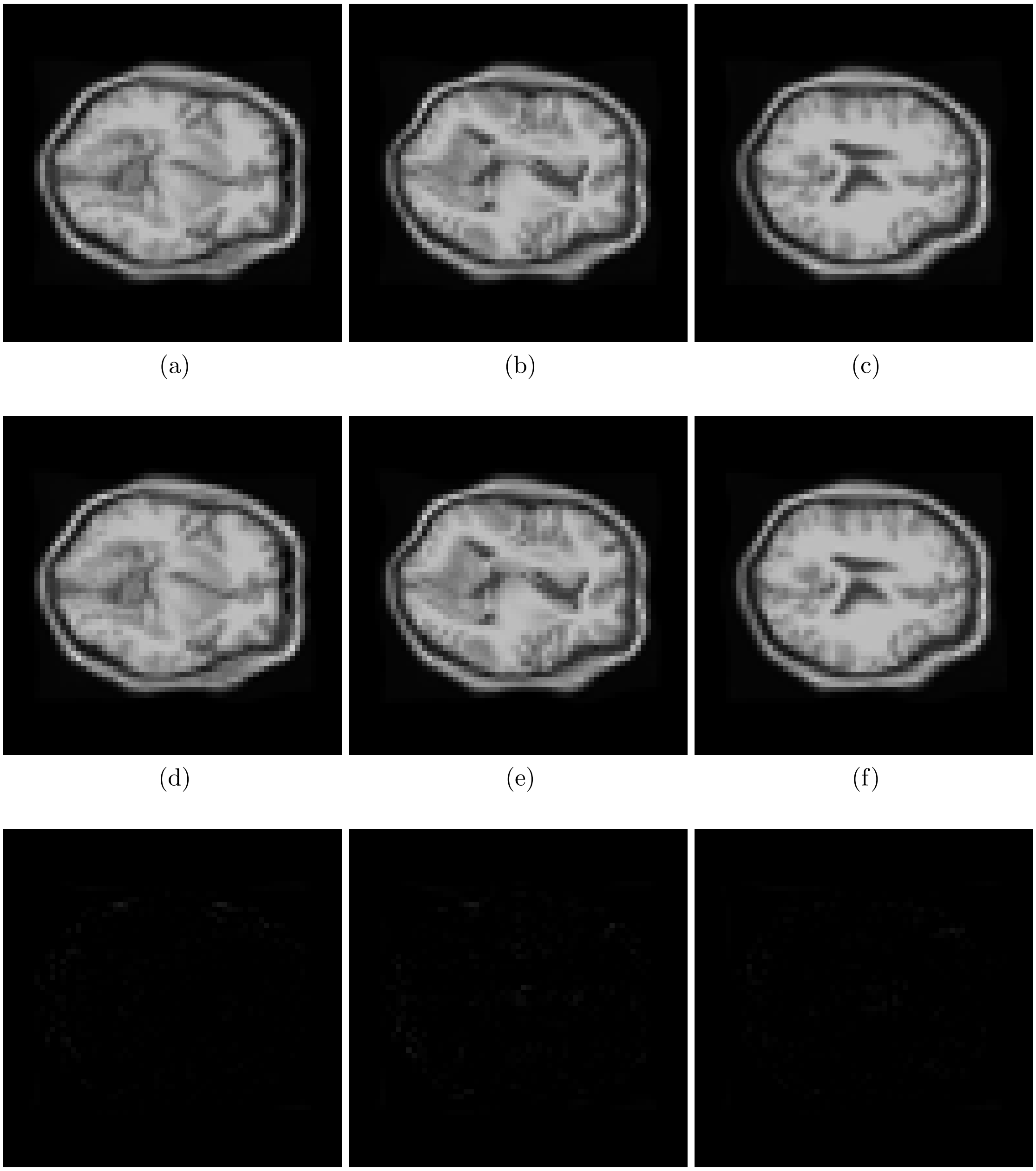


Figure 4.24. (a) to (c) are the slice 17, 33, and 49 of the synthetically generated reference image R respectively. (d) to (f) are the slice 17, 33, and 49 of the test image T registered to the R using nonrigid image registration method version one that optimizes f^1 to f^4. (g) to (i) shows the residual difference between (a) to (c) and (d) to (f).

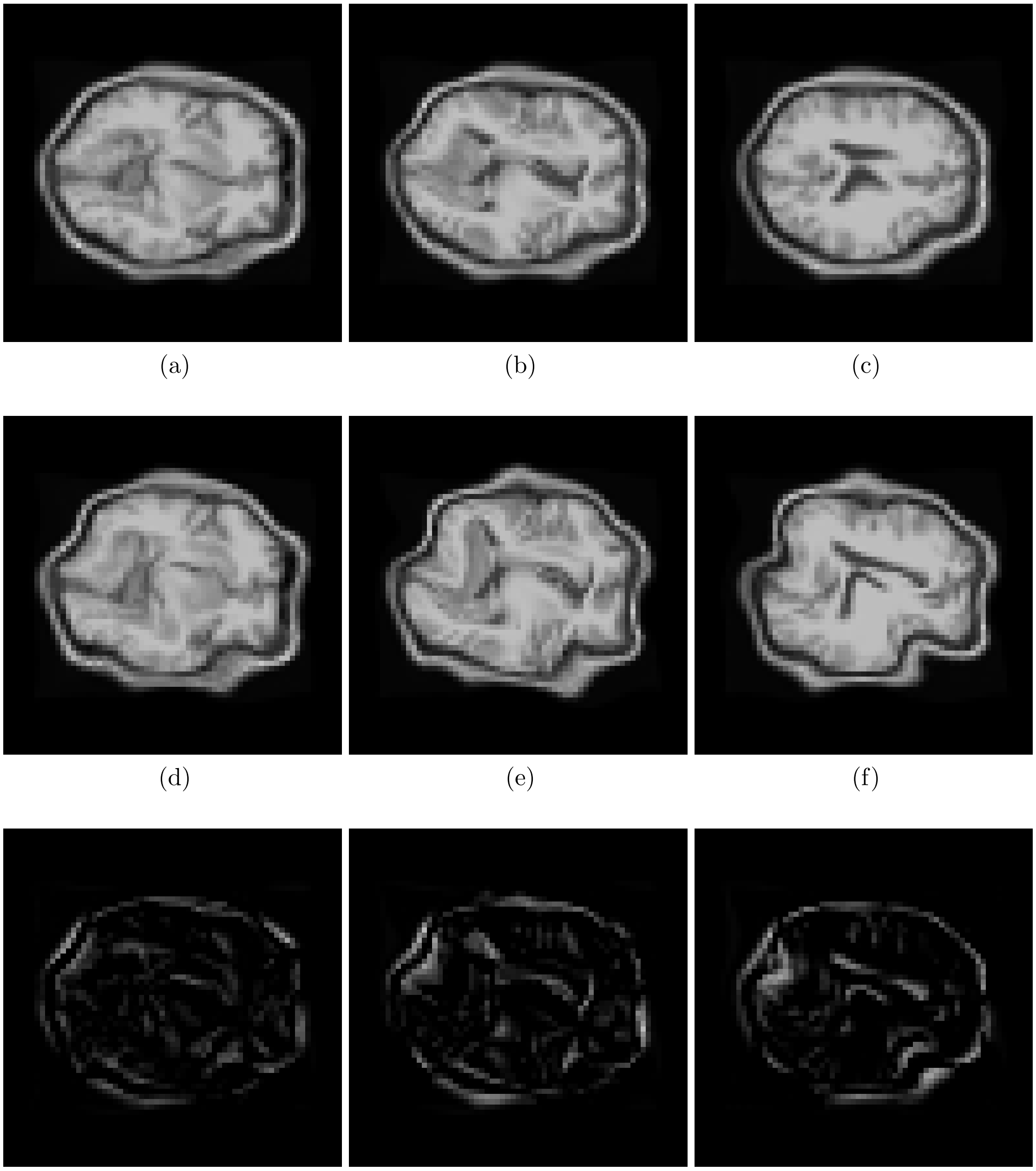
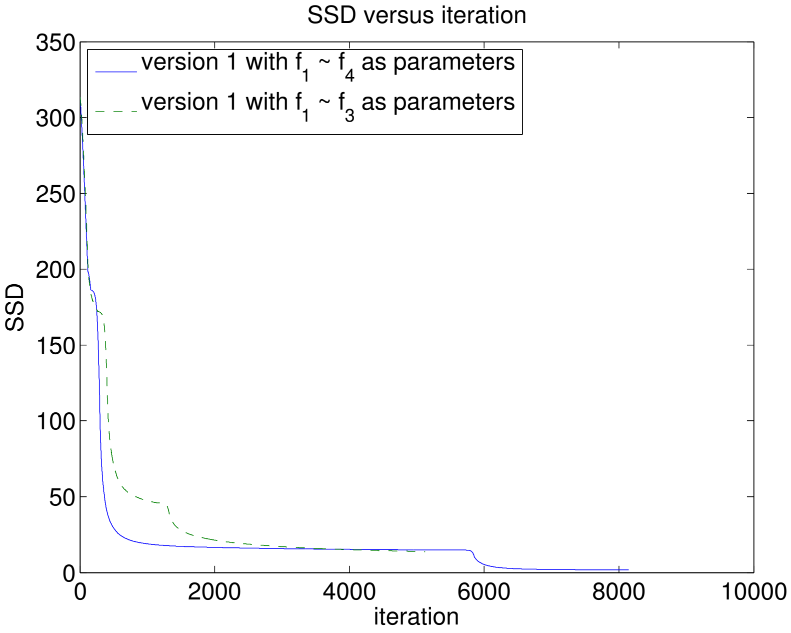


Figure 4.25. (a) to (c) are the slice 17, 33, and 49 of the synthetically generated reference images R respectively. (d) to (f) are the slice 17, 33, and 49 of the test images T registered to the R using nonrigid image registration method version one that optimizes f^1 to f^3. (g) to (i) shows the residual difference between (a) to (c) and (d) to (f).

Table 4.1. 3D nonrigid image registration without controlling the solvability constraint experiment results.

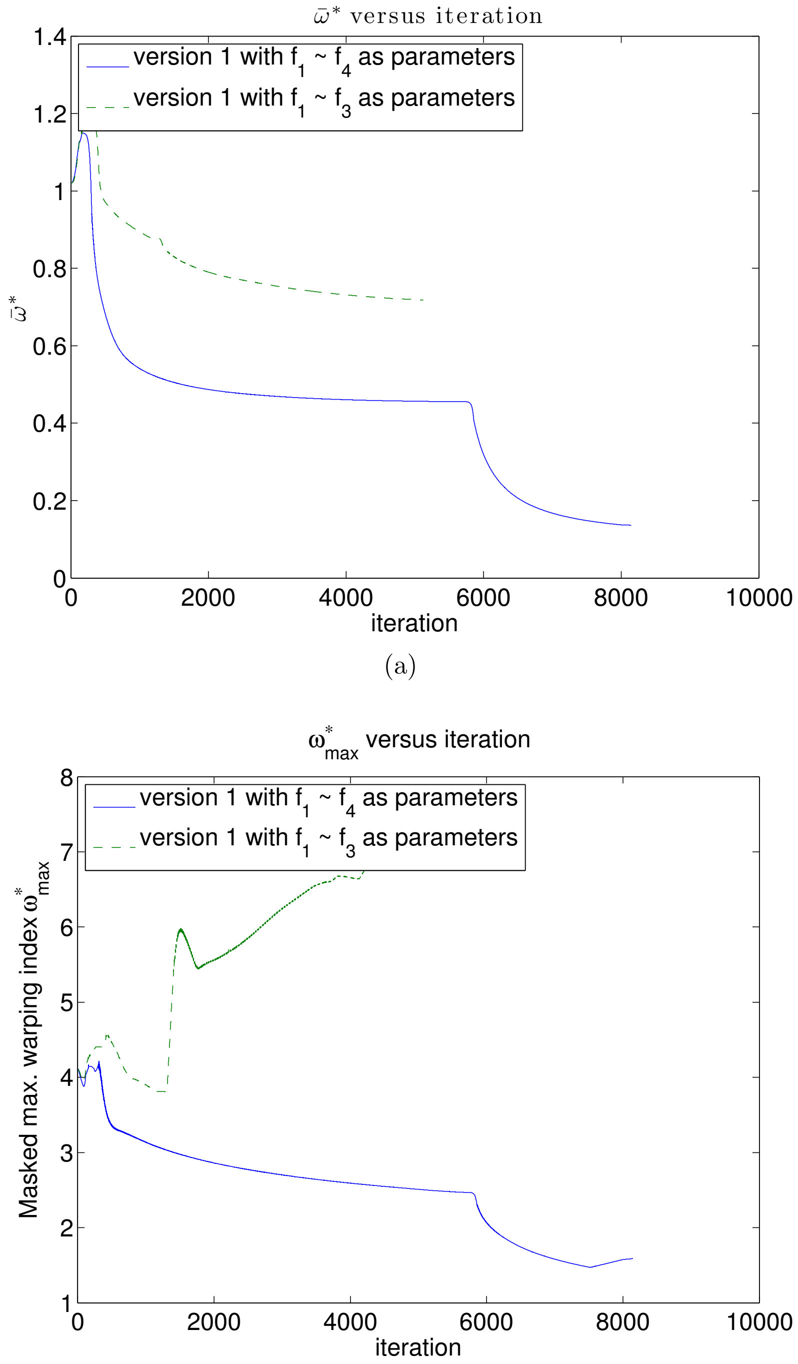
	Optimize f^1 to f^4	Optimize f^1 to f^3
SSD before registration	313.2860	313.2860
$\bar{\omega}^*$ before registration	1.0200	1.0200
$\bar{\omega}^*_{\text{max}}$ before registration	4.1165	4.1165
SSD after registration	1.8277	13.8077
$\bar{\omega}^*$ after registration	0.1360	0.7180
$\bar{\omega}^*_{\text{max}}$ after registration	1.5910	7.8739
Total iteration	8148	5126
Computation time per iteration	9.8372	8.2559
Minimum determinant of Jacobian $\det(J)$	0.1841	-0.2135
Minimum monitor function	0.1617	0.1000



(a)

Figure 4.26.3D registration without solvability SSD experiment result.

The quantitative experiment results of the 3D nonrigid image registration of the 3D data pairs prepared in the previous section are presented in Table 4.1 and figures from Fig. 4.26 to Fig. 4.27.



(a)

(b)

Figure 4.27. 3D registration experiment result. (a) Masked mean warping index $\bar{\omega}^*$ versus iteration. (b) Masked maximum warping index ω^*_{max} versus iteration.

In Fig. 4.26 and Fig. 4.27, the abrupt changes of SSD, $\bar{\omega}^*$, and $\omega^*_{\max}$ values are due to the knot space changing of the multiresolution strategy. From these figures, the version one algorithm optimizing four parameters f^1 to f^4 outperforms the one optimizing three parameters f^1 to f^3.

Table 4.1 presents the initial values of the registration quality assessment metrics (SSD, masked mean warping index $\bar{\omega}^*$, and masked maximum warping index $\omega^*_{\max}$), the quality assessment metrics after registration, total registration iteration, and mean computation time per iteration. From Table 4.1, the registration by optimizing f^1 to f^4 outperforms the registration by optimizing f^1 to f^3 from the registration quality point of view, since registration by optimizing four components can achieve lower SSD, $\bar{\omega}^*$, and $\omega^*_{\max}$.

We also observed the determinant of Jacobian of the registered deformation field in Table 4.1. The registration optimizing with three parameters f^1 to f^3 has the $\det(J)$ deviated from the monitor function and the negative -0.2135 indicates the generated mesh is folding. On the other hand, the $\det(J)$ of the registration optimizing with four parameters is closely related to the monitor function.

Registration algorithm version one implemented by optimizing four components outperforms the one by optimizing three components, because by increasing one dimension of the searching space, it is easier to overcome the local minima.

4.9 Experiment Eight

4.9.1 Purpose

Sec. 4.8 conducted a 3D image registration experiment by proposed algorithm version one that optimizes f^1 to f^4 without enforcing the solvability constraint [34] — $\nabla \cdot (\nabla \times \boldsymbol{\eta}) = 0$, where f^1 is the monitor function and f^2 to f^4 are the three curl components of the intermediate vector field $\boldsymbol{\eta}$ used in the proposed nonrigid image registration algorithm version one.

The purpose of this experiment is to observe if the solvability constraint is necessary for the nonrigid image registration algorithm version one. We have evaluated the registration quality and efficiency of two different implementations to enforce the solvability constraint.

First Approach After completing gradient descent optimization steps — $f^{m\prime} = \mu \cdot df^m$, $m = 2, 3, 4$, we can normalize these three components by $f^{m\prime} = f^{m\prime} - (\nabla \cdot \langle f^2, f^3, f^4 \rangle)\, x_{m-1}/3$, where $m = 2, 3, 4$ and $\boldsymbol{x} = \{x_1, x_2, x_3\}$ is the position coordinate of the monitor function and curl value.

Proof of the First Approach. If we normalize f^2 to f^4 in the following way

$$f^{2\prime} = f^2 - \frac{x_1(\nabla \cdot \langle f^2, f^3, f^4 \rangle)}{3} \tag{4.12}$$

$$f^{3\prime} = f^3 - \frac{x_2(\nabla \cdot \langle f^2, f^3, f^4 \rangle)}{3} \tag{4.13}$$

$$f^{4\prime} = f^4 - \frac{x_3(\nabla \cdot \langle f^2, f^3, f^4 \rangle)}{3}, \tag{4.14}$$

then

$$\nabla \cdot \left\langle f^{2\prime}, f^{3\prime}, f^{4\prime} \right\rangle = \frac{\partial}{\partial x_1} \left(f^2 - \frac{x_1(\nabla \cdot \langle f^2, f^3, f^4 \rangle)}{3} \right)$$

$$+\frac{\partial}{\partial x_2}\left(f^3-\frac{x_2(\nabla\cdot\langle f^2,f^3,f^4\rangle)}{3}\right)$$
$$+\frac{\partial}{\partial x_3}\left(f^4-\frac{x_3(\nabla\cdot\langle f^2,f^3,f^4\rangle)}{3}\right)=0.$$

□

Second Approach In this approach, after completing gradient descent optimization steps — $f^{m\prime}=\mu\cdot df^m$, $m=2,3$, we can normalize these three components by $f^{4\prime}=-x_3\left(\nabla\cdot\langle f^2,f^3,0\rangle\right)$, where x_3 is the third component of the position coordinate of the monitor function and curl value.

Proof of the Second Approach. If we normalize f^2 to f^4 in the following way

$$f^{2\prime}=f^2 \tag{4.15}$$

$$f^{3\prime}=f^3 \tag{4.16}$$

$$f^{4\prime}=-x_3(\nabla\cdot\left\langle f^2,f^3,0\right\rangle), \tag{4.17}$$

then

$$\nabla\cdot\left\langle f^{2\prime},f^{3\prime},f^{4\prime}\right\rangle=\frac{\partial f^2}{\partial x_1}+\frac{\partial f^3}{\partial x_2}-\frac{\partial x_3(\nabla\cdot\langle f^2,f^3,0\rangle)}{\partial x_3}=0.$$

□

4.9.2 Experimental Design

We use the same reference image R and test image T generated in Sec. 4.8.2. We compared the two implementations of the registration algorithm version one that enforce the solvability constraint with the one without any solvability constraint in terms of SSD, $\bar{\omega}^*$, and $\omega^*_{\max}$.

Table 4.2. 3D nonrigid image registration optimizing f^1 to f^4 by two different approaches to enforce the $\nabla \cdot \langle f^2, f^3, f^4 \rangle = 0$ constraint.

	First Approach	Second Approach
SSD before registration	313.2860	313.2860
$\bar{\omega}^*$ before registration	1.0200	1.0200
$\bar{\omega}^*_{\text{max}}$ before registration	4.1165	4.1165
SSD after registration	27.4268	97.6646
$\bar{\omega}^*$ after registration	0.5219	1.5392
$\bar{\omega}^*_{\text{max}}$ after registration	4.1822	10.3124
Total iteration	961	1385
Computation time per iteration	8.4372	8.4291
Minimum determinant of Jacobian $\det(J)$	0.1273	-0.2178
Minimum monitor function	0.1061	0.1000

4.9.3 Experiment Results and Analysis

Figure 4.28 and Figure 4.29 display the selective slices of the reference image R, registered test image using nonrigid image registration version one that optimizes four parameters f^1 to f^4 using the two approaches to enforce the solvability constraint, and the residual difference between the them. Both results are not satisfying, but the first approach to enforce the solvability constraint is better than the second approach.

From Table 4.1 and Table 4.2, enforcing the solvability constraint does not result in better registration quality (SSD, $\bar{\omega}^*$, and ω^*_{max}). This is because we did not consider the solvability constraint during the computation of $d\text{ssd}/df^m$, $m = 1 \sim 4$.

The final $\det(J)$ of the second approach to enforce the solvability constraint is out of sync with the monitor function, which is similar to the $\det(J)$ result of the "optimizing f^1 to f^3" column illustrated in Table 4.1. This indicates that the out of sync problem is not caused by enforcing the solvability constraint or not.

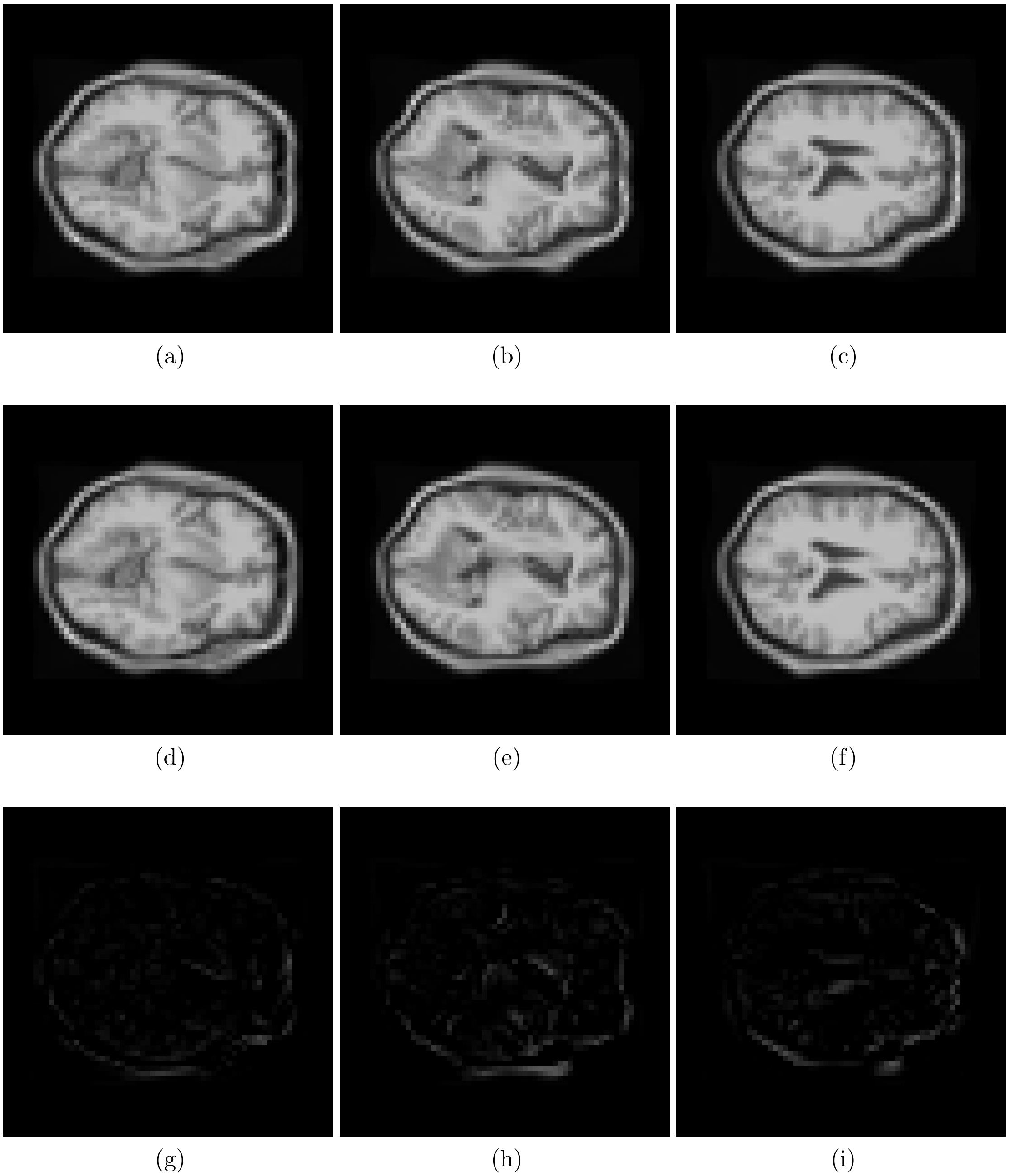


Figure 4.28. (a) to (c) are the slice 17, 33, and 49 of the synthetically generated reference images R respectively. (d) to (f) are the slice 17, 33, and 49 of the test images T registered to the R using nonrigid image registration method version one that optimizes f^1 to f^4 with the first constraint approach. (g) to (i) shows the residual difference between (a) to (c) and (d) to (f).

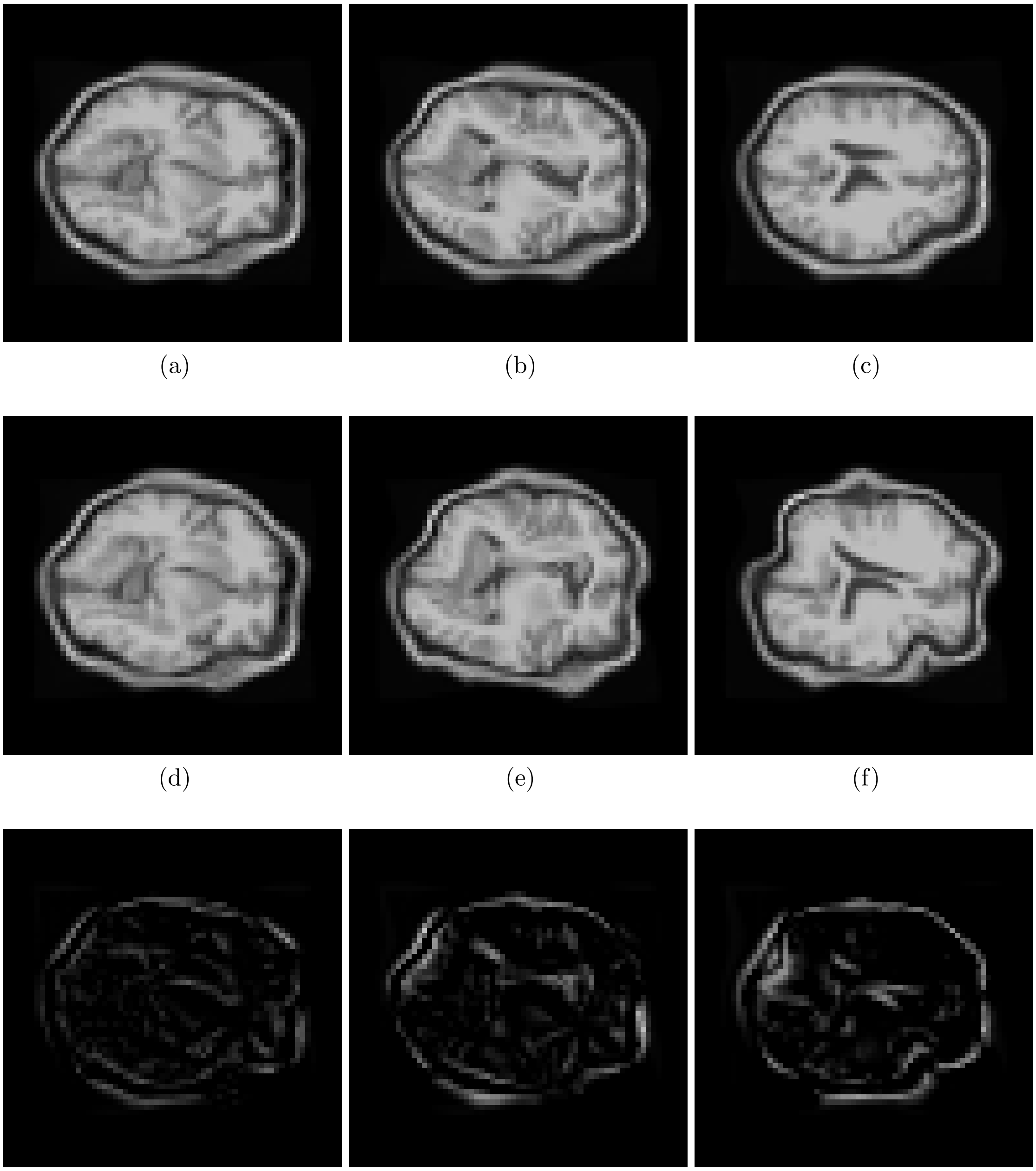


Figure 4.29. (a) to (c) are the slice 17, 33, and 49 of the synthetically generated reference images R respectively. (d) to (f) are the slice 17, 33, and 49 of the test images T registered to the R using nonrigid image registration method version one that optimizes f^1 to f^4 with the second constraint approach. (g) to (i) shows the residual difference between (a) to (c) and (d) to (f).

Summarizing the experiments conducted in Sec. 4.8 and Sec. 4.9, the proposed registration algorithm version one works best if we optimize the monitor function and three curl components without the solvability constraint.

Theoretically, without the solvability constraint, the solved intermediate vector field $\boldsymbol{\eta}$ may not be accurate. On the other hand, if the gradient decent optimization process can find the global optima, then it may lead the whole system to fulfill the solvability constraint. The observed absolute mean of $\nabla \cdot (\nabla \times \boldsymbol{\eta})$ of the registration algorithm version one without the solvability constraint is 0.0162, which is quite close to 0.

The reason of why enforcing the solvability constraint does not result in good outcome is because that when the solvability constraint is enforced, it alters df^2, df^3, and df^4 suggested by the gradient decent optimization process. Therefore, it may mislead the registration process and generate the poor result.

CHAPTER 5

SUMMARY

5.1 Accomplishments and Issues

Two versions of the proposed nonrigid image registration algorithms have been implemented. The third version was not implemented due to the difficulty of the optimization derivation explained in Sec. 3.4.3. The problem could be solved if the intermediate vector field $\boldsymbol{\eta}$ is solved on the moving grid generated by the deformation function $\phi(\boldsymbol{\xi})$ instead of on the original grid $\boldsymbol{\xi}$, but by doing so, more computational difficulty will occur which can be an interesting topic in the future works.

In the proposed algorithms, a div-curl solver is essential for obtaining the intermediate vector field $\boldsymbol{\eta}$. The LSFEM inverse filtering method was first introduced to avoid the large matrix computation and the SOR method was adopted to obtain an approximated LSFEM inverse filter. Later, a more accurate way to solve the $\boldsymbol{\eta}$ by the Poisson solver was introduced. Therefore, we reformed the div-curl system of $\boldsymbol{\eta}$ into a Poisson's equation set, which turned out to be more accurate and more efficient.

To register efficiently, Chapter 3 derives a high performance gradient descent optimizer that implements the gradient of SSD with respect to the monitor function and curl values in the form of convolution.

The registration quality and computational performance comparison of the proposed nonrigid image registration with the LSFEM inverse filtering and Poisson solver approach was conducted in Sec. 4.2. As expected, using the Poisson solver outperforms the inverse filtering approach.

Sec. 4.4 compares the proposed nonrigid image registration algorithm version one and two. Version one appears to outperform version two, but the cause of this result is still unknown at this time. However, it is worthwhile to discover that why version two cannot work as well as version one does in the future.

The noise tolerance capability of the proposed registration algorithm is demonstrated in Sec. 4.7. With the capability to register images quite accurately (mean error under 0.5 pixel) under 10 dB SNR noise indicates that the proposed algorithm has an excellent noise tolerance feature.

Theoretically, the solvability constraint is necessary for the 3D image registration of the proposed registration algorithm, but the experiments conducted in Sec. 4.8 and Sec. 4.9 suggest that it is not necessary to enforce this constraint in the proposed algorithm. This is because of the solvability constraint self-retaining feature of the gradient decent optimization process, in other words, if the gradient decent optimization can accurately optimize the monitor function and curl components, the solvability constraint should be automatically fulfilled.

Summarizing the experiments conducted in Chap. 4, we demonstrated the excelled registration quality, robustness, and good noise tolerance of the proposed algorithm.

5.2 Contributions

Traditional image registration models minimize the cost functional comprising a similarity functional and a regularization functional.

$$\textit{cost functional} = \textit{similarity functional} + \beta \times \textit{regularization functional}$$

Chapter 3 formulates three versions of the nonrigid image registration algorithms based on the well-established deformation based grid generation method that can simplify the traditional variational model. The main feature of the proposed algorithm is the capability to control the area/volume of the resulting deformation field, therefore, the regularization term can be discarded.

In the proposed model, there is only one term in the cost functional — the similarity functional. It is favorable to avoid the regularization functional because the regularization parameter β is application specific, can only be determined from experiments, and varies with different tissues. If the β is set too low, the resulting deformation field may suffer from grid folding; if the β is set too high, then the regularization term weighted too much, and the registration result becomes poor. Therefore, containing the similarity functional only in the cost functional is one of the advantages of the proposed model over the traditional model.

Possible Applications The proposed algorithm can be easily applied to do the volume preserving [53] nonrigid image registration for incompressible content since the proposed algorithm can control the volume through setting the boundary on the monitor function. Another interesting application is for the cardiac image registration [54]. The proposed algorithms is capable to register accurately in large deformation cases, which fit in with the large deformation feature vastly seen in the cardiac imaging field.

Other possible applications are fusion of images from different modalities [6, 7], super-resolution in positron emission tomography (PET) imaging [8], visualizing diffusion tensor MR images (DTI) [9, 10], atlas based segmentation [11, 12], geometric correction of functional magnetic resonance imaging (fMRI) [13], pattern recognition

[14], etc. The purposed image registration in these applications is to establish the correspondence among the pixels/voxels of image pairs.

5.3 Future Works

In the future, the proposed algorithm is going to be applied to the possible applications mentioned in the previous section.

University of Iowa has started the Non-rigid Image Registration Evaluation Project (NIREP) [55]. It extends prior image registration validation projects and provides image validation database for evaluating nonrigid image registration algorithms. This is a good opportunity to collaborate with them to extensively validate the registration algorithm proposed in this dissertation.

Currently, the registration is performed on the original grid points, it is intriguing to explore the possibility to compute on the deformed grid points. By doing this, the registration accuracy may be further improved.

REFERENCES

[1] M. Droske and M. Rumpf, "A variational approach to nonrigid morphological image registration," *SIAM Journal on Applied Mathematics*, vol. 64, no. 2, pp. 668–687, 2004.

[2] L. Gottesfeld Brown, "A survey of image registration techniques," *ACM computing surveys*, vol. 24, no. 4, pp. 325–376, 1992.

[3] J. Maintz and M. Viergever, "A survey of medical image registration," *Medical Image Analysis*, vol. 2, no. 1, pp. 1–36, 1998.

[4] B. Zitová and J. Flusser, "Image registration methods: a survey," *Image and Vision Computing*, vol. 21, no. 11, pp. 977–1000, 2003.

[5] A. Gholipour, N. Kehtarnavaz, R. Briggs, M. Devous, and K. Gopinath, "Brain Functional Localization: A Survey of Image Registration Techniques," *Medical Imaging, IEEE Transactions on*, vol. 26, no. 4, pp. 427–451, 2007.

[6] R. Narayanan, "Diffeomorphic transformations for automatic multi-modality image registration," Ph.D. dissertation, The University of Michigan, 2007.

[7] S. Warfield, F. Talos, C. Kemper, E. Cosman, A. Tei, M. Ferrant, B. Macq, W. Wells III, P. Black, F. Jolesz, *et al.*, "Augmenting intraoperative mri with preoperative fmri and dti by biomechanical simulation of brain deformation," *Proceedings of SPIE*, vol. 5029, p. 77, 2003.

[8] J. Kennedy, O. Israel, A. Frenkel, R. Bar-Shalom, and H. Azhari, "Super-Resolution in PET Imaging," *Medical Imaging, IEEE Transactions on*, vol. 25, no. 2, pp. 137–147, 2006.

[9] J. Blaas, C. Botha, C. Majoie, A. Nederveen, F. Vos, and F. Post, "Interactive visualization of fused fmri and dti for planning brain tumor resections," *Proceedings of SPIE*, vol. 6509, p. 65091P, 2007.

[10] D. Laidlaw, E. Ahrens, D. Kremers, M. Avalos, R. Jacobs, and C. Readhead, "Visualizing diffusion tensor images of the mouse spinal cord," *Visualization'98. Proceedings*, pp. 127–134, 1998.

[11] Y. Zhou and J. Bai, "Multiple abdominal organ segmentation: An atlas-based fuzzy connectedness approach," *Information Technology in Biomedicine, IEEE Transactions on*, vol. 11, no. 3, pp. 348–352, 2007.

[12] R. Kikinis, M. Shenton, D. Iosifescu, R. McCarley, P. Saiviroonporn, H. Hokama, A. Robatino, D. Metcalf, C. Wible, C. Portas, *et al.*, "A digital brain atlas for surgical planning, model-driven segmentation, and teaching," *IEEE Transactions on Visualization and Computer Graphics*, vol. 2, no. 3, pp. 232–241, 1996.

[13] D. Wu, J. Suri, V. Magnotta, T. Przebinda, and V. DeBrunner, "Improved method for correction of systematic bias introduced by the sub-voxel image registration process in functional magnetic resonance imaging (fmri)," *Medical Imaging 2006: Physiology, Function, and Structure from Medical Images. Edited by Manduca, Armando; Amini, Amir A. Proceedings of the SPIE,*, vol. 6144, pp. 271–279, 2006.

[14] T. Lin and Y. Zheng, "Node-matching-based pattern recognition method for retinal blood vessel images," *Optical Engineering*, vol. 42, p. 3302, 2003.

[15] R. Bajcsy and S. Kovačič, "Multiresolution elastic matching," *Computer Vision, Graphics, and Image Processing*, vol. 46, no. 1, pp. 1–21, 1989.

[16] J.-P. Thirion, "Image matching as a diffusion process: an analogy with maxwell's demons," *Medical Image Analysis*, vol. 2, no. 3, pp. 243–260, 1998.

[17] G. Christensen, R. Rabbitt, and M. Miller, “Deformable templates using large deformation kinematics,” *Image Processing, IEEE Transactions on*, vol. 5, no. 10, pp. 1435–1447, 1996.

[18] B. Fischer and J. Modersitzki, “Curvature Based Image Registration,” *Journal of Mathematical Imaging and Vision*, vol. 18, no. 1, pp. 81–85, 2003.

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

[20] M. Bern and D. Eppstein, “Mesh generation and optimal triangulation,” *Computing in Euclidean Geometry*, vol. 1, pp. 23–90, 1992.

[21] K. Miller and R. Miller, “Moving finite elements. i,” *SIAM Journal on Numerical Analysis*, vol. 18, no. 6, pp. 1019–1032, 1981.

[22] K. Miller, “Moving finite elements. ii,” *SIAM Journal on Numerical Analysis*, vol. 18, no. 6, pp. 1033–1057, 1981.

[23] N. N. Carlson and K. Miller, “Design and application of a gradient-weighted moving finite element code i: in one dimension,” *SIAM J. Sci. Comput.*, vol. 19, no. 3, pp. 728–765, 1998.

[24] ——, “Design and application of a gradient-weighted moving finite element code ii: in two dimensions,” *SIAM J. Sci. Comput.*, vol. 19, no. 3, pp. 766–798, 1998.

[25] G. Liao and D. Anderson, “A new approach to grid generation,” *Applicable Analysis*, vol. 44, no. 3, pp. 285–298, 1992.

[26] G. Liao, T. Pan, and J. Su, “A numerical grid generator based on moser’s deformation method,” *Numerical Methods for Partial Differential Equations*, vol. 10, no. 1, pp. 21–31, 1994.

[27] X. Cai, B. Jiang, and G. Liao, “Adaptive grid generation based onthe least-squares finite-element method,” *Computers and Mathematics with Applications*, vol. 48, no. 7-8, pp. 1077–1085, 2004.

[28] J. Thompson, Z. Warsi, and C. Mastin, *Numerical grid generation: foundations and applications.* Elsevier North-Holland, Inc. New York, NY, USA, 1985.

[29] T. Baker, "Mesh generation: Art or science?" *Progress in Aerospace Sciences*, vol. 41, no. 1, pp. 29–63, 2005.

[30] G. Liao, T. Pan, and J. Su, "A numerical grid generator based on moser's deformation method," *Numerical Methods for Partial Differential Equations*, vol. 10, no. 1, pp. 21–31, 1994.

[31] B. Semper and G. Liao, "A moving grid finite-element method using grid deformation," *Numerical Methods for Partial Differential Equations*, vol. 11, pp. 603–615, 1995.

[32] P. Bochev, G. Liao, and G. dela Pena, "Analysis and computation of adaptive moving grids by deformation," *Numerical Methods for Partial Differential Equations*, vol. 12, no. 4, pp. 489–506, 1996.

[33] L. Jie, "New development of the deformation method," PhD dissertation, The University of Texas at Arlington, 2006.

[34] B. Jiang, *The Least-Squares Finite Element Method: Theory and Applications in Computational Fluid Dynamics and Electromagnetics.* Springer, 1998.

[35] Z. Gülboy and N. Gelçkinli, "On FIR filters having approximate FIR inverses with a specified LMS error," *Signal Processing*, vol. 19, no. 1, pp. 9–15, 1990.

[36] D. Young, "Iterative methods for solving partial difference equations of elliptic type," *Transactions of the American Mathematical Society*, vol. 76, no. 1, pp. 92–111, 1954.

[37] H. Chen, C. Hsieh, and G. Liao, "Non-rigid image registration using adaptive grid generation: preliminary results," *Biomedical Imaging: From Nano to Macro, 2007. ISBI 2007. 4th IEEE International Symposium on*, pp. 580–583, 2007.

[38] J. Betts and S. Campbelt, "Discretize then Optimize," *Mathematics for Industry: Challenges and Frontiers: a Process View: Practice and Theory*, pp. 140–157, 2005.

[39] W. Pratt, *Digital Image Processing: PIKS Inside.* John Wiley & Sons, Inc. New York, NY, USA, 2001.

[40] R. Hockney, "A Fast Direct Solution of Poisson's Equation Using Fourier Analysis," *Journal of the ACM (JACM)*, vol. 12, no. 1, pp. 95–113, 1965.

[41] G. SKOLLERMO, "A Fourier method for the numerical solution of Poisson's equation," *Mathematics of Computation*, vol. 29, no. 131, pp. 697–711, 1975.

[42] J. Kybic and M. Unser, "Fast parametric elastic image registration," *Image Processing, IEEE Transactions on*, vol. 12, no. 11, pp. 1427–1442, November 2003.

[43] [Online]. Available: http://www.nlm.nih.gov/research/visible/visible_human.html

[44] F. L. Bookstein, "Principal warps: thin-plate splines and the decomposition of deformations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 6, pp. 567–585, 1989.

[45] J. Demmel, *Applied Numerical Linear Algebra.* Society for Industrial Mathematics, 1997.

[46] A. Netravali and B. Haskell, *Digital Pictures: Representation, Compression, and Standards.* Plenum Pub Corp, 1995.

[47] M. Rabbani and P. Jones, *Digital Image Compression Techniques.* SPIE Press, 1991.

[48] [Online]. Available: http://www.bic.mni.mcgill.ca/brainweb/

[49] C. CA, K. Kwan, and R. Evans, "BrainWeb: online interface to a 3-D MRI simulated brain database," *NeuroImage*, vol. 5, no. 4, 1997.

[50] R. Kwan, A. Evans, and G. Pike, "MRI simulation-based evaluation of image-processing andclassification methods," *Medical Imaging, IEEE Transactions on*, vol. 18, no. 11, pp. 1085–1097, 1999.

[51] K. Remi, A. Evans, and G. Pike, "An Extensible MRI Simulator for Post-Processing Evaluation," in *Proceedings of the 4th International Conference on Visualization in Biomedical Computing.* Springer, 1996, pp. 135–140.

[52] D. Collins, A. Zijdenbos, V. Kollokian, J. Sled, N. Kabani, C. Holmes, and A. Evans, "Design and construction of a realistic digital brain phantom," *Medical Imaging, IEEE Transactions on*, vol. 17, no. 3, pp. 463–468, 1998.

[53] T. Rohlfing, C. R. M. Jr., D. A. Bluemke, and M. A. Jacobs, "Volume-preserving non-rigid registration of mr breast images using free-form deformation with an incompressibility constraint," *IEEE Trans. Med. Imaging*, vol. 22, no. 6, pp. 730–741, 2003.

[54] T. Mäkelä, P. Clarysse, O. Sipilä, N. Pauna, Q. Pham, T. Katila, and I. Magnin, "A review of cardiac image registration methods," *IEEE Trans Med Imaging*, vol. 21, no. 9, pp. 1011–1021, 2002.

[55] [Online]. Available: http://www.nirep.org

BIOGRAPHICAL STATEMENT

Chih-Yao Hsieh received his B.Sc. degree in Physics in 2000 from the National Taiwan Normal University in Taiwan and his M.Sc. degree in Computer Science from the University of Texas at Arlington in 2003. He is now pursuing a Ph.D. degree in Computer Science at the University of Texas at Arlington, Texas, United States. His current research interests include biomedical imaging, remote sensing, image registration, image fusion, numerical methods, and pervasive computing.