New Development of Grid Generation and Image Analysis

By

BEN HILDEBRAND

DISSERTATION

Submitted in partial fulfillment of the requirements

for the degree of Doctor of Philosophy at

The University of Texas at Arlington

August, 2020

Arlington, Texas

To my mother, Janice Hildebrand,

and my father, Douglas Hildebrand.

Acknowledgments

I would like to thank my supervising professor Dr. Guojun Liao for his patience, insights, and guidance during my academic journey at the University of Texas at Arlington. He has been a crucial aid and support to my work while in the graduate program here.

I thank Dr. Jianzhong Su and Dr. David Jorgensen for serving as the Chair and Associate Chair of the Mathematics Department, respectively during my time at UT Arlington. Their aid in assigning me teaching assignments helped fund me during times of need.

I am also grateful for Dr. Hristo Kojouharov, Dr. Ren-Cang Li, and Dr. Li Wang for serving on my dissertation committee. I greatly appreciate their time and attention.

Finally, I would like to thank my parents and close friends for all the support, encouragement, and generosity they have provided me throughout the years.

Abstract

Image segmentation and registration are indispensable tools for the aid in medical diagnoses by experts.

The current gold-standard for image segmentation is manual labeling of pixels by experts which is cumbersome and inefficient. In the paper by Zhu et. al. [21] grids are generated through the deformation method for grid generation and differential properties of these grids are used in a deep learning algorithm for image segmentation. In this dissertation, we develop a new method for generating grid images based on the Variational Method. This new grid generation method generates grids based on image pixel intensities which improves upon the deformation method for grid generation in constructing such grids.

Image registration is used in quantitative analysis based on the grid representation of the registration field, but this is an ill-posed problem. Therefore, many models of regularization are used to regularize the problem. As a result there are many different models that have many different grid representations with large discrepancies. In fact, even the same model with different parameters often result in different deformations with large discrepancies. In this dissertation, we develop a platform for combining different registration fields generated by different methods with the aim of improving robustness.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Diffeomorphisms and Grid Generation

A diffeomorphism is a mapping between manifolds that is differentiable with differentiable inverse. Diffeomorphism is an active research area in differential geometry [7, 21].

Two already established methods for grid generation are the deformation method and the Variational Method.

We review both the deformation method for grid generation and the Variational Method in this dissertation. Both methods serve as aids for new developments in image segmentation and registration.

1.1.1 Deformation Method

In [9] Dacorogna and Moser first proved the existence of diffeomorphism under a Jacobian determinant constraint. Later, in [14] Liao and Anderson proposed a reformulation of the deformation method to construct diffeomorphisms numerically rep-

resented by grids and named the method *deformation method for grid generation*. In [6, 12] the Least-Squares Finite Element Method (LSFEM) was implemented to solve the divergence-curl system that arises in the formulation of the deformation method for grid generation.

1.1.2 Variational Method

Another method for grid generation based on the deformation method was developed in [7, 8] called the Variational Method. The Variational Method constructs a diffeomorphism $\boldsymbol{\phi}$ via an iterative process based on a cost functional the Sum of Squared Differences (SSD).

1.2 Image Segmentation

Image segmentation is the process of partitioning a digital image into multiple segments by labeling pixels. The goal of image segmentation is that of identifying landmarks or regions of interest within the image (e.g. identifying a tumor within a brain image). Quantitative analysis is very common in many neurological diseases such as Alzheimer's disease, epilepsy, schizophrenia, multiple sclerosis (MS), cancer, and infectious and degenerative diseases [21]. In this quantitative analysis MRI segmentation is used to quantify changes in brain structure.

In Zhu et. al. [21], the deformation method for grid generation is used to generate grids that resemble images based on pixel intensity (grid images). Differential properties of the grid images are then computed (Jacobian determinant (JD), divergence (DIV), and curl vector (CV)) and used as features in training on a convolutional neu-

ral network (CNN) to improve performance of brain segmentation. Some drawbacks to the grids generated by the deformation method in the paper by Zhu et. al. [21] are that the grids depend on a set of optimization parameters and they are lacking in accuracy.

In this dissertation, we propose a new 2D grid generation algorithm based on the Variational Method with an appropriate choice for the prescribed Jacobian determinant (JD) that relies only on the information provided in the image file; thus eliminating the need for optimization parameters as well as producing high accuracy.

1.3 Image Registration

Currently, image registration is widely used in many areas such as computer vision, biological imaging, remote sensing, and medical imaging [18]. In particular, deformable image registration in medical imaging has been an active research topic for decades [3].

Image registration is the process of aligning two or more images according to a similarity measure. Assuming that a pair of registered images differ only by Gaussian noise, registration accuracy can be evaluated by measuring pixel (voxel for 3D) intensity difference of the registered image pair [16]. A common similarity measure used to measure pixel difference is the Sum of Squared Differences (SSD) between pixel intensity of corresponding pixels between an image pair. A perfect registration results in $SSD = 0$.

There are two general types of deformable image registration: affine registration (i.e. alignment via rotation, reflection, scaling, translation, and shearing) and non-

linear registration. Affine registration is very limited in accuracy due to the limited, linear transformations available and cannot overcome the misalignment of local details. Therefore, nonlinear registration models are needed for an adequate level of registration accuracy.

One popular methodology to performing nonlinear image registration is the use of diffeomorphism to identify a deformation that accurately aligns image data.

In this dissertation, we investigate a new application of diffeomorphic averaging. We apply the grid averaging concept from [19, 20] to the registration fields (deformation grids) of an image registration algorithm that relies on a parameter θ. We then extend this idea to averaging various registration fields that have already been determined by varying registration methods with diffeomorphic guarantees. This acts as a starting point for building a platform for averaging diffeomorphic registration fields of varying registration methods to improve robustness of desirable grid representations of the registration field while retaining high alignment accuracy for a given registration pair.

1.4 Organization

This dissertation contains 6 chapters. The organization of this dissertation is as follows: chapter 1 is the introduction. chapters 2 and 3 review the deformation method for grid generation and the Variational Method for grid generation, respectively. Chapter 4 presents an appropriate prescription for the Jacobian determinant based on normalized image pixel intensities for use in the Variational Method to accurately generate grids directly from images. Chapter 5 presents a platform for

averaging grid representations for diffeomorphisms obtained from various registration algorithms as an aim to produce more robust deformations in image registration. Finally, Chapter 6 presents conclusions of the new developments.

Chapter 2

The Deformation Method for Grid Generation

2.1 Introduction

In the field of differential geometry, a diffeomorphism is a bijective map between manifolds that is smooth and has smooth inverse. Diffeomorphism is an active research area in differential geometry [7, 22]. In [9] Dacorogna and Moser first proved the existence of diffeomorphism under a Jacobian determinant constraint. Later, in [14] Liao and Anderson proposed a reformulation of the deformation method to construct diffeomorphisms numerically represented by grids, hence the name *deformation method for grid generation*. In [6, 12] the Least-Squares Finite Element Method (LSFEM) was implemented to solve the divergence-curl system that arises in the formulation of the *deformation method for grid generation*. Further development in triangular mesh generation was performed in [15] and development of higher order mesh generation in [19]. More recently, the deformation method for grid generation has been applied toward image segmentation and image registration [21, 22].

In [21] Zhu et. al. apply the deformation method for grid generation to produce grid representations of input images in order to use first-order differential operators

including the Jacobian determinant (JD), divergence (DIV), and curl vector (CV) of the corresponding generated grid. Then, these first-order differential operators are used as CNN channels with other modalities (T1-weighted, T1-IR, and T2-FLAIR) or single T1-weighted modality to improve the performance of brain segmentation.

In [22] Zhu et. al. apply the deformation method for grid generation as a means to again obtain first-order differential operators and are then used on the VoxelMorph CNN architecture of [10].

2.2 Deformation Method for Grid Generation

The goal of the deformation method is to construct a family of diffeomorphisms $\boldsymbol{\phi}(\boldsymbol{\xi}, t)$ such that $J(\boldsymbol{\phi}) = f_0(\boldsymbol{\phi}, t)$ for a given monitor function $f_0(\boldsymbol{x}, t)$, for every $t \in [t_0, T]$. In particular, we consider $t \in [0, 1]$. See Figure 2.1 for a 2D depiction of the problem we aim to solve.

Let Ω_0 and Ω_t be simply connected, bounded, moving (or fixed) domains such that $\Omega_0 \subset \mathbb{R}^n$ and $\Omega_t \subset \mathbb{R}^n$ where $n = 2$, or $n = 3$, and $t \in [0, 1]$. Let $\boldsymbol{v}(\boldsymbol{x}, t)$ be the velocity field associated with the boundary $\partial\Omega_t$. Here we required that $\boldsymbol{v}(\boldsymbol{x}, t) \cdot \boldsymbol{n} =$

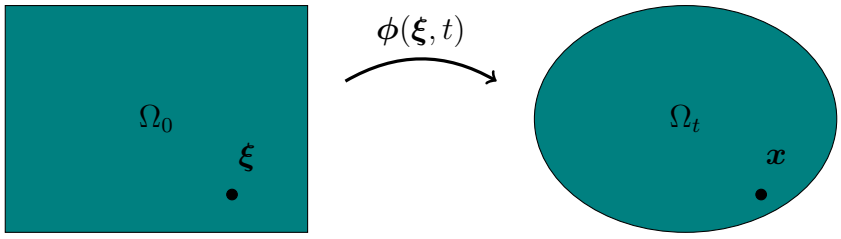


Figure 2.1: The deformation method method in 2D (moving boundary case) seeks to determine the diffeomorphism $\boldsymbol{\phi}(\boldsymbol{\xi}, t)$ from domain Ω_0 to domain Ω_t where $|\Omega_0| = |\Omega_t|$.

0 everywhere on $\partial\Omega_t$ where $\boldsymbol{n}$ is the outward normal vector of $\partial\Omega_t$. Employing this boundary condition ensures that points on the initial boundary remain on the boundary of the new domain (slippery wall condition). Given a strictly positive, scalar function that is continuously differentiable (i.e. $0 < f(\boldsymbol{x}, t) \in C^1$) with domain $\Omega_t \times [0, 1]$ with constraints

$$\begin{cases} f(\boldsymbol{x}, 0) = 1, \\ \displaystyle\int_{\Omega_t} \frac{1}{f(\boldsymbol{x}, t)} d\boldsymbol{x} = |\Omega_0|, \end{cases} \tag{2.1}$$

we determine a new diffeomorphism $\boldsymbol{\phi}(\boldsymbol{\xi}, t) : \Omega_0 \to \Omega_t$ such that

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

for every $t \in [0, 1]$. We do this in a two-step process.

First, solve the div-curl system:

$$\begin{cases} \nabla \cdot \boldsymbol{u}(\boldsymbol{x}, t) = -\dfrac{\partial}{\partial t}\left(\dfrac{1}{f(\boldsymbol{x}, t)}\right) & \text{on } \Omega_t \\ \nabla \times \boldsymbol{u}(\boldsymbol{x}, t) = \boldsymbol{0} & \text{on } \Omega_t \\ \boldsymbol{u}(\boldsymbol{x}, t) = \dfrac{\boldsymbol{v}(\boldsymbol{x}, t)}{f(\boldsymbol{x}, t)} & \text{on } \partial\Omega_t \end{cases} \tag{2.3}$$

Solving this div-curl system is done by LSFEM in [6, 12] to determine the vector field $\boldsymbol{u}(\boldsymbol{x}, t)$. Due to the complexity of LSFEM we do not go into detail of the method in this work. Once we have determined $\boldsymbol{u}(\boldsymbol{x}, t)$ we then solve the deformation ODE for each fixed $\boldsymbol{\xi} \in \Omega_0$:

$$\begin{cases} \dfrac{\partial \boldsymbol{\phi}(\boldsymbol{\xi}, t)}{\partial t} = f(\boldsymbol{\phi}(\boldsymbol{x}, t), t) \cdot \boldsymbol{u}(\boldsymbol{x}, t) & \text{on } \Omega_t \\ \boldsymbol{\phi}(\boldsymbol{\xi}, \mathbf{0}) = \boldsymbol{\xi} & \text{on } \partial\Omega_t. \end{cases} \tag{2.4}$$

This can be done by many ODE solvers (e.g. Explicit-Euler Method). Once solved, we have the diffeomorphic map $\boldsymbol{\phi}(\boldsymbol{\xi}, t)$.

See Algorithm 1 for implementation.

Algorithm 1: Deformation Method for Grid Generation

Result: Construct diffeomorphism $\boldsymbol{\phi}(\Omega_0, 1)$

Initialize: $t = 0, \boldsymbol{\phi}(\boldsymbol{\xi}, 0) = \boldsymbol{\xi}, f(\boldsymbol{x}, 0) = 1, dt = 0.1$

Prescribe: $f(\boldsymbol{x}, t)$

while $t \in [0, 1]$ **do**

  Update: $f(\boldsymbol{x}, t)$ with $t = t + dt$

  Normalize: $f(\boldsymbol{x}, t) = f(\boldsymbol{x}, t) \cdot \int_{\Omega_t} \frac{1}{f(\boldsymbol{x}, t)} d\boldsymbol{x}$

  Compute: $-\frac{\partial}{\partial t}\left(\frac{1}{f(\boldsymbol{x}, t)}\right)$

  Solve: div-curl system 2.3 by LSFEM to obtain $\boldsymbol{u}(\boldsymbol{x}, t)$ on Ω_t

  Apply: boundary conditions to $\boldsymbol{u}(\boldsymbol{x}, t)$

  Update: $\boldsymbol{\phi}(\boldsymbol{\xi}, t)$ from Ω_t to Ω_{t+dt} by solving ODE 2.4

  Update: $t = t + dt$

end

Output: $\boldsymbol{\phi}$

Chapter 3

The Variational Method with Prescribed Jacobian Determinant for Grid Generation

3.1 Introduction

In this chapter we review the Variational Method for constructing diffeomorphisms which is based on the deformation method. This is a key component of the development discussed in the following chapters. The Variational Method for determining diffeomorphisms was developed in [7, 8] and constructs a diffeomorphism $\boldsymbol{\phi}$ via an iterative process. This method aims to minimize a cost functional, the Sum of Squared Differences (SSD), defined as

$$SSD = \frac{1}{2}\int_{\Omega} \left(J(\boldsymbol{\phi}(\boldsymbol{x})) - f_0(\boldsymbol{x})\right)^2 + \alpha |curl(\boldsymbol{\phi}(\boldsymbol{x})) - \boldsymbol{g_0}(\boldsymbol{x})|^2 d\boldsymbol{x}$$

based on prescribed Jacobian determinant and curl vector, where $J(\boldsymbol{\phi}(\boldsymbol{x}))$ is the Jacobian determinant of the diffeomorphism $\boldsymbol{\phi}$ and $\alpha \geq 0$ is a weight parameter on the curl term. In [7] there are three variational methods presented: *Variational Method with Prescribed Jacobian Determinant* (version 1); *Variational Method with Prescribed*

Jacobian Determinant and Curl Vector (version 2); and version 3 generalizes the prescribed functions in version 2 (i.e. $f_0(\boldsymbol{x})$ and $\boldsymbol{g_0}(\boldsymbol{x})$ are generalized to $f_0(\boldsymbol{\phi}(\boldsymbol{x}))$ and $\boldsymbol{g_0}(\boldsymbol{\phi}(\boldsymbol{x}))$, respectively). Both version 2 and 3 consider the parameter $\alpha \geq 0$ as a weight on the prescribed curl vector term in the cost functional that is to be minimized. Version 1 is equivalent to the special case when $\alpha = 0$. We specifically focus on version 1 with a slight variant in the constraint as it applies to our generation of grids directly from images (we refer to these as *grid images*) in the next chapter of this dissertation.

3.2 2D Variational Method with Prescribed Jacobian Determinant

We now proceed in reviewing the *Variational Method with prescribed Jacobian Determinant* (version 1) from [7] in 2D, but with a variant constraint. For details on the Variational Method for 3D see chapter 3 of [7].

Let Ω be a simply connected, bounded region such that $\Omega \subset \mathbb{R}^n$ where $n = 2$ (for 2D). Let $0 < f_0(\boldsymbol{x}) \in C^1$ be a scalar function with domain Ω along with

$$\int_\Omega f_0(\boldsymbol{x}) d\boldsymbol{x} = |\Omega|. \tag{3.1}$$

We wish to find a diffeomorphism, $\boldsymbol{\phi} : \Omega \to \Omega$ that minimizes the Sum of Squared Differences (SSD) cost functional

$$SSD = \frac{1}{2} \int_\Omega \left(J(\boldsymbol{\phi}(\boldsymbol{x})) - f_0(\boldsymbol{x}) \right)^2 d\boldsymbol{x} \tag{3.2}$$

subject to the constraint

$$\begin{cases} \Delta\boldsymbol{\phi}(\boldsymbol{x}) = \boldsymbol{f}(\boldsymbol{x}) & \text{on } \Omega \\ \boldsymbol{\phi}(\boldsymbol{x}) = \boldsymbol{x} & \text{on } \partial\Omega \end{cases} \tag{3.3}$$

where $\boldsymbol{f}(\boldsymbol{x})$ is a control function.

In order to minimize SSD we apply gradient descent. To apply gradient descent we first must derive the variational gradient, $\frac{\partial SSD}{\partial \boldsymbol{f}}$. Since our framing of the problem is the 2D case ($n = 2$), we let $\boldsymbol{\phi}(\boldsymbol{x}) = (\phi_1(x, y), \phi_2(x, y))$ and $\boldsymbol{f}(\boldsymbol{x}) = (f_1(x, y), f_2(x, y))$.

Let $\delta\boldsymbol{f}$ be a variation of $\boldsymbol{f}$ that vanishes at the boundary. Then

$$\begin{aligned} SSD &= \frac{1}{2}\int_\Omega \left(J(\boldsymbol{\phi}(\boldsymbol{x})) - f_0(\boldsymbol{x})\right)^2 d\boldsymbol{x} \\ \delta SSD &= \int_\Omega \left(J(\boldsymbol{\phi}(\boldsymbol{x})) - f_0(\boldsymbol{x})\right) \delta J(\boldsymbol{\phi}(\boldsymbol{x})) d\boldsymbol{x}. \end{aligned} \tag{3.4}$$

Note that since $J(\boldsymbol{\phi}(\boldsymbol{x})) = det \begin{pmatrix} \phi_{1x} & \phi_{1y} \\ \phi_{2x} & \phi_{2y} \end{pmatrix} = \phi_{1x}\phi_{2y} - \phi_{2x}\phi_{1y}$, then by product rule $\delta J(\boldsymbol{\phi}) = \delta(\phi_{1x}\phi_{2y} - \phi_{2x}\phi_{1y}) = \delta\phi_{1x}\phi_{2y} + \phi_{1x}\delta\phi_{2y} - \delta\phi_{2x}\phi_{1y} - \phi_{2x}\delta\phi_{1y}$. Thus

$$\begin{aligned} \delta SSD &= \int_\Omega \left(J(\boldsymbol{\phi}) - f_0\right) \delta(\phi_{1x}\phi_{2y} - \phi_{2x}\phi_{1y}) d\boldsymbol{x} \\ &= \int_\Omega \left(J(\boldsymbol{\phi}) - f_0\right) (\delta\phi_{1x}\phi_{2y} + \phi_{1x}\delta\phi_{2y} - \delta\phi_{2x}\phi_{1y} - \phi_{2x}\delta\phi_{1y}) d\boldsymbol{x} \\ &= \int_\Omega \left(J(\boldsymbol{\phi}) - f_0\right) \left[(\phi_{2y}, -\phi_{2x}) \cdot \nabla\delta\phi_1 + (-\phi_{1y}, \phi_{1x}) \cdot \nabla\delta\phi_2\right] d\boldsymbol{x} \end{aligned}$$

Define $\boldsymbol{a_1} = -\left(J(\boldsymbol{\phi}) - f_0\right)(\phi_{2y}, -\phi_{2x})$ and $\boldsymbol{a_2} = -\left(J(\boldsymbol{\phi}) - f_0\right)(-\phi_{1y}, \phi_{1x})$. We get

$$\delta SSD = \int_\Omega (-\boldsymbol{a_1} \cdot \nabla\delta\phi_1 - \boldsymbol{a_2} \cdot \nabla\delta\phi_2) d\boldsymbol{x}. \tag{3.5}$$

We introduce a vector-valued function $\boldsymbol{g} = (g_1, g_2)$ satisfying

$$\begin{cases} \Delta g_1 = \nabla \cdot \boldsymbol{a_1} & \text{on } \Omega \\ \Delta g_2 = \nabla \cdot \boldsymbol{a_2} & \text{on } \Omega \\ g_1, g_2 = 0 & \text{on } \partial\Omega. \end{cases} \tag{3.6}$$

Recall two corollaries of the Divergence Theorem:

Corollary 1. *By applying the divergence theorem to the product of a scalar function v and a vector field $\boldsymbol{u}$ on domain Ω*

$$\int_\Omega (\boldsymbol{u} \cdot \nabla v + v\,(\nabla \cdot \boldsymbol{u}))\, dV = \int_{\partial\Omega} (v\boldsymbol{u} \cdot \boldsymbol{n})\, dS, \tag{3.7}$$

where $\boldsymbol{n}$ is the outward unit normal vector.

Corollary 2. *By applying the divergence theorem to the product of a scalar function v and a vector field ∇w on domain Ω*

$$\int_\Omega (\nabla w \cdot \nabla v + v(\Delta w))dV = \int_{\partial\Omega} (v\nabla \cdot \boldsymbol{n})dS, \tag{3.8}$$

where $\boldsymbol{n}$ is the outward unit normal vector.

Note that since $v = f_i, i = 1, 2$ vanishes at the boundary, then ϕ_1 and ϕ_2 also vanish at the boundary, and the integrals on the right-hand side of both corollaries vanish as a result.

$$
\begin{aligned}
\delta SSD &= \int_{\Omega} \left(-\boldsymbol{a_1} \cdot \nabla \delta\phi_1 - \boldsymbol{a_2} \cdot \nabla \delta\phi_2\right) d\boldsymbol{x} && (3.9)\\
&= \int_{\Omega} \left(\nabla \cdot \boldsymbol{a_1} \delta\phi_1 + \nabla \cdot \boldsymbol{a_2} \delta\phi_2\right) d\boldsymbol{x} && (3.10)\\
&= \int_{\Omega} \left(\Delta g_1 \delta\phi_1 + \Delta g_2 \delta\phi_2\right) d\boldsymbol{x} && (3.11)\\
&= \int_{\Omega} \left(g_1 \delta\Delta\phi_1 + g_2 \delta\Delta\phi_2\right) d\boldsymbol{x} && (3.12)\\
&= \int_{\Omega} \left((g_1, g_2) \cdot \delta\Delta\boldsymbol{\phi}\right) d\boldsymbol{x} \\
&= \int_{\Omega} \left(\boldsymbol{g} \cdot \delta\boldsymbol{f}\right) d\boldsymbol{x} && (3.13)
\end{aligned}
$$

Proceeding from 3.9 to 3.10 we apply Cor. 1 with $v = \phi_1$ and $\boldsymbol{u} = -\boldsymbol{a_1}$ or $v = \phi_2$ and $\boldsymbol{u} = -\boldsymbol{a_2}$ as needed to switch the order of ∇. Proceeding from 3.11 to 3.12 we apply Cor. 2 *twice* with $v = \phi_1$ and $w = g_1$ or $v = \phi_2$ and $w = g_2$ as needed. Finally, from 3.13 we end up with

$$
\frac{\delta SSD}{\delta \boldsymbol{f}} = \boldsymbol{g} = (g_1, g_2). \tag{3.14}
$$

See Algorithm 2 for the implementation of the Variational Method with prescribed Jacobian determinant.

Algorithm 2: Variational Method with Prescribed Jacobian Determinant

Result: Iteratively generate diffeomorphism $\boldsymbol{\phi}$

Prescribe: f_0

Initialize: $\boldsymbol{f} = \mathbf{0}$, $\boldsymbol{\phi}(\boldsymbol{x}) = \boldsymbol{x}$, $better = true$

while $\Delta t > t_{tol}$ *and* $ratio > ratio_{tol}$ *and* $iter > iter_{max}$ **do**

  if $better = true$ **then**

    Compute: $(J(\boldsymbol{\phi}(\boldsymbol{x})) - f_0)$, $\boldsymbol{a_1}$, $\boldsymbol{a_2}$, $\nabla \cdot \boldsymbol{a_1}$, $\nabla \cdot \boldsymbol{a_2}$

    Solve: Poisson equations to obtain g_1 and g_2

    Compute: $\frac{\delta SSD}{\delta \boldsymbol{f}}$ based on g_1 and g_2

  end

  Update: $f_{new} = f_{old} - \Delta t \cdot \frac{\delta SSD}{\delta \boldsymbol{f}}$

  Solve: Poisson equations to obtain ϕ_1 and ϕ_2

  Check: SSD and compute $ratio$

  if *SSD decreases* **then**

    $better = true$

    $\Delta t = \Delta t \cdot t_{up}$

    $f_{old} = f_{new}$

  else

    $better = false$

    $\Delta t = \Delta t \cdot t_{down}$

  end

end

Output: $\boldsymbol{\phi}$

Chapter 4

Grid Generation Based on Image Pixel Intensities

4.1 Introduction

Now that we have reviewed the Variational Method with Prescribed Jacobian Determinant we are ready to present a novel approach to generating grids based on images (we refer to these grids as *grid images* henceforth). In particular, we apply the variational approach of chapter 3 using image pixel intensities to determine the prescribed Jacobian determinant. Motivation for generating grids directly from images (i.e. generating grid images) arises in image segmentation.

Image segmentation is the process of partitioning a digital image into multiple segments by labeling pixels. The goal of image segmentation is that of identifying landmarks within the image (e.g. identifying a tumor within a brain image). Quantitative analysis is very common in many neurological diseases such as Alzheimer's disease, epilepsy, schizophrenia, multiple sclerosis (MS), cancer, and infectious and degenerative diseases [21]. In this quantitative analysis MRI segmentation is used to quantify changes in brain structure.

Currently, the gold-standard for image segmentation is left to experts to manually

analyze images and perform segmentation which is expensive, time consuming, and prone to human error [21]. For this reason, development of an automated process for image segmentation is highly sought after. Generating an accurate, canonical grid corresponding to an image is a key tool for automating the image segmentation process. Furthermore, in [21] grid images are generated that are aimed to resemble the input image. Features from these grids (Jacobian determinant (JD), divergence (DIV), and curl vector (CV)) are then used for data training in convolutional neural networks (CNNs) in an effort for automating the image segmentation process.

The general idea presented in this dissertation for grid generation directly from images should apply to both 2D and 3D images. Although 3D grids should be viable with similar formulation as that which is presented in this chapter, 3D grids have not yet been implemented and is left for future work.

The underlying idea for our approach in obtaining a grid image $\boldsymbol{\phi}$ is that grid cell size corresponds to a pixel (voxel for 3D) intensity value: the larger the cell size the brighter the pixel value corresponding to that cell; the smaller the cell size the darker the pixel value corresponding to that cell. For an expanding grid cell size we have $J(\boldsymbol{\phi}) > 1$ (corresponds to a *brighter* pixel value); for a shrinking grid cell size $J(\boldsymbol{\phi}) < 1$ (corresponds to a *darker* pixel value); and $J(\boldsymbol{\phi}) \approx 1$ when a pixel is gray (neither bright or dark, but rather an average of bright and dark). Of course, we also require that $J(\boldsymbol{\phi}) > 0$ everywhere to ensure $\boldsymbol{\phi}$ is a diffeomorphism. To achieve $J(\boldsymbol{\phi}) > 0$ we normalize pixel intensity within an image to avoid zero or negative pixel values, and then prescribe our JD based on pixel intensities throughout the image.

All versions of the Variational Method from [7] require a prescribed Jacobian determinant. From [19] we know that diffeomorphism constructed by Jacobian de-

terminant alone does not ensure uniqueness, but prescribed Jacobian determinant and curl vector together does. Even though uniqueness of the diffeomorphism is not guaranteed with a prescribed Jacobian determinant alone, surprisingly, we see in this chapter that if we prescribe the Jacobian determinant in a particular way, even without a prescribed curl vector, we obtain an accurate grid that resembles the input image. From the grid generated we may extract the curl vector associated with the grid (i.e. the curl vector is updated based on version 1 of the variational method once the final iteration of grid generation is complete). Many grid generation techniques require the specification of various optimization parameters (e.g. the deformation method used in [21] relies on a choice for dt, grid spacing h, and number of grid nodes N). The grid we obtain through the Variational Method with prescribed Jacobian determinant does not require strict optimization parameters. We simply base the prescription of the Jacobian determinant on the pixel intensity throughout the given input image and the grid is automatically generated. In particular, we compare results of our method with grids generated in Figure 1 of [21] and Figure 1 of [22].

4.2 Prescribing the Jacobian Determinant Based on Pixel Intensities

In this section we first derive the mathematical framework for the monitor function f_0 based on the necessary conditions for version 1 of the Variational Method, and then discretize our result.

4.2.1 Mathematical Formulation of the Prescribed Jacobian Determinant

Recall from version 1 of the Variational Method that we must have the following conditions if we desire application of the method for grid generation in 2D.

Let $\Omega \subset \mathbb{R}^2$ be simply connected and bounded with scalar function $0 < f_0(\boldsymbol{x}) \in C^1$ such that

$$\int_\Omega f_0(\boldsymbol{x})d\boldsymbol{x} = |\Omega|. \tag{4.1}$$

To satisfy these conditions, suppose we have a function $0 < I_0(\boldsymbol{x}) \in C^1$ on domain Ω such that $f_0(\boldsymbol{x}) = k \cdot I_0(\boldsymbol{x})$ for some scalar $k > 0$. Then $0 < f_0(\boldsymbol{x}) \in C^1$ and we need only determine the scalar k for which 4.1 holds. Proceeding we have

$$\begin{aligned}
\int_\Omega f_0(\boldsymbol{x})d\boldsymbol{x} &= \int_\Omega k \cdot I_0(\boldsymbol{x})d\boldsymbol{x} \\
&= k\int_\Omega I_0(\boldsymbol{x})d\boldsymbol{x} \\
&= |\Omega| \\
&\Downarrow \\
k &= \frac{|\Omega|}{\int_\Omega I_0(\boldsymbol{x})d\boldsymbol{x}}.
\end{aligned} \tag{4.2}$$

4.2 suggests if we define our monitor function f_0 as

$$f_0(\boldsymbol{x}) = \left(\frac{|\Omega|}{\int_\Omega I_0(\boldsymbol{x})d\boldsymbol{x}}\right) \cdot I_0(\boldsymbol{x}) \tag{4.3}$$

then our conditions for carrying out version 1 of the Variational Method are satisfied and we iteratively generate a diffeomorphism $\boldsymbol{\phi}$. Now we are ready to discretize and apply this monitor function as the prescribed JD for grid generation directly from images.

4.2.2 Discretization of the Prescribed Jacobian Determinant

For our purposes here, we consider 2D gray scale images and therefore our domain Ω is all points within the rectangular boundary of an image. The measure on Ω is then the total area encompassing any given image in question. That is, for an image that has m pixels by n pixels, $|\Omega| = m \cdot n$. Since pixel intensity is a function of a pixel's location we might consider $f_0(\boldsymbol{x}) = I(\boldsymbol{x})$, where $I(\boldsymbol{x})$ is the pixel intensity value at the point $\boldsymbol{x} \in \Omega$. However, it's possible that the pixel intensity at any given location might be zero (or even negative depending on the image file information). To avoid non-positive pixel intensities throughout the image, we apply the normalization $I_0(\boldsymbol{x}) = I(\boldsymbol{x}) + I_{max} + \varepsilon$ where $I_{max} = \max_{\boldsymbol{y}} |I(\boldsymbol{y})|, \forall \boldsymbol{y} \in \Omega$ and $\varepsilon > 0$ for each pixel location $\boldsymbol{x} \in \Omega$ within the image. Furthermore, it is not necessarily the case that $\sum_{\boldsymbol{x}} I_0(\boldsymbol{x}) = m \cdot n$. So we must choose to define $f_0(\boldsymbol{x})$ in such a way that the discretized summation will agree with $|\Omega| = m \cdot n$. Basing our function assignment on the formulation that lead to 4.3 we might consider the following assignment for $f_0(\boldsymbol{x})$:

$$f_0(\boldsymbol{x}) = \frac{m \cdot n}{\sum_{\boldsymbol{i}} I_0(\boldsymbol{x_i})} I_0(\boldsymbol{x}) \tag{4.4}$$

for $i = 1, 2, \ldots, m \cdot n$ and $\boldsymbol{x}_i \in \Omega$.

Since an image is essentially just a uniform grid with pixel intensities assigned at each node, the discrete summation over our function should equal the total area of the uniform grid. That is, $\sum_{\boldsymbol{x}} f_0(\boldsymbol{x}) = |\Omega|$. Performing this summation we find

$$\begin{aligned}
\sum_{\boldsymbol{x}} f_0(\boldsymbol{x}) &= \sum_{\boldsymbol{x}} \frac{m \cdot n}{\sum_i I_0(\boldsymbol{x}_i)} I_0(\boldsymbol{x}) \\
&= \frac{m \cdot n}{\sum_i I_0(\boldsymbol{x}_i)} \sum_{\boldsymbol{x}} I_0(\boldsymbol{x}) \\
&= m \cdot n \\
&= |\Omega|. \qquad (4.5)
\end{aligned}$$

Algorithm 3 demonstrates the full process of normalizing pixel intensity and generating the grid image $\boldsymbol{\phi}$.

With $I_0 > 0, \forall \boldsymbol{x} \in \Omega$ we use $f_0 = \frac{m \cdot n}{\sum I_0} I_0$ as our prescribed JD in our Variational Method for grid generation. Note that we are directly generating a grid from an image without the use of parameters; we generate the grid $\boldsymbol{\phi}$ solely based on the image file information. Also of note, one caveat of this method lies in the gradient descent portion of the algorithm. For $n \times n$ images, we can take larger (artificial) time steps (e.g $t_{up} = 1.1$ for stepping up and $t_{down} = 2/3$ when stepping down). For $m \times n$ images, the stopping criterion is met too quickly and does not produce an accurate grid so we need to use smaller time steps (e.g. $t_{up} = 1.01$ when stepping up and $t_{down} = 0.99$ when stepping down). A simple *if* statement based on image dimensions handles this phenomena.

Next we view some examples and compare to other grid image results.

Algorithm 3: Generation of Grid Image $\boldsymbol{\phi}$

Result: Normalizes image pixel intensity; prescribes JD; output $\boldsymbol{\phi}$

1 **Initialize:** $\varepsilon = 0.1$

2 **Input:** $I_0 = read(image\,file)$ (gray scale image of size $m \times n$ pixels)

3 **Update (double precision):** $I_0 = double(I_0)$

4 **Update (fixed boundary):** $I_0 = resize(I_0, m-2, n-2)$

5 **Compute:** $I_{max} = max(|I_0|)$

6 **Update:** $I_0 = I_0 + I_{max} + \varepsilon$

7 **Compute:** $k = \dfrac{m \cdot n}{\sum_{\boldsymbol{x}} I_0(\boldsymbol{x})}$

8 **Compute:** $f_0 = k \cdot I_0$

9 **Call:** Algorithm 2

4.3 Examples of Grid Images

In this section we perform numerical experiments that show the robustness and accuracy of our method. All of the grid images generated in blue in this section use Algorithm 3 with optimization parameters as follows: $t_{tol} = 10^{-15}, ratio_{tol} = 0.001, iter_{max} = 5000$. The program (written in Matlab) prompts the user for a string (the name of the image file name) and then generates the grid image based on the file requested. An *if* statement is used to handle the t_{up} and t_{down} conditions mentioned in the latter portion of the previous section based on image dimensions. All computations are performed on a machine with Intel® Core™ i7-9700K CPU @ 3.60 × 8 GHz and 16 GiB of memory.

4.3.1 Experiment 1: Generating Grid Images from Familiar Images

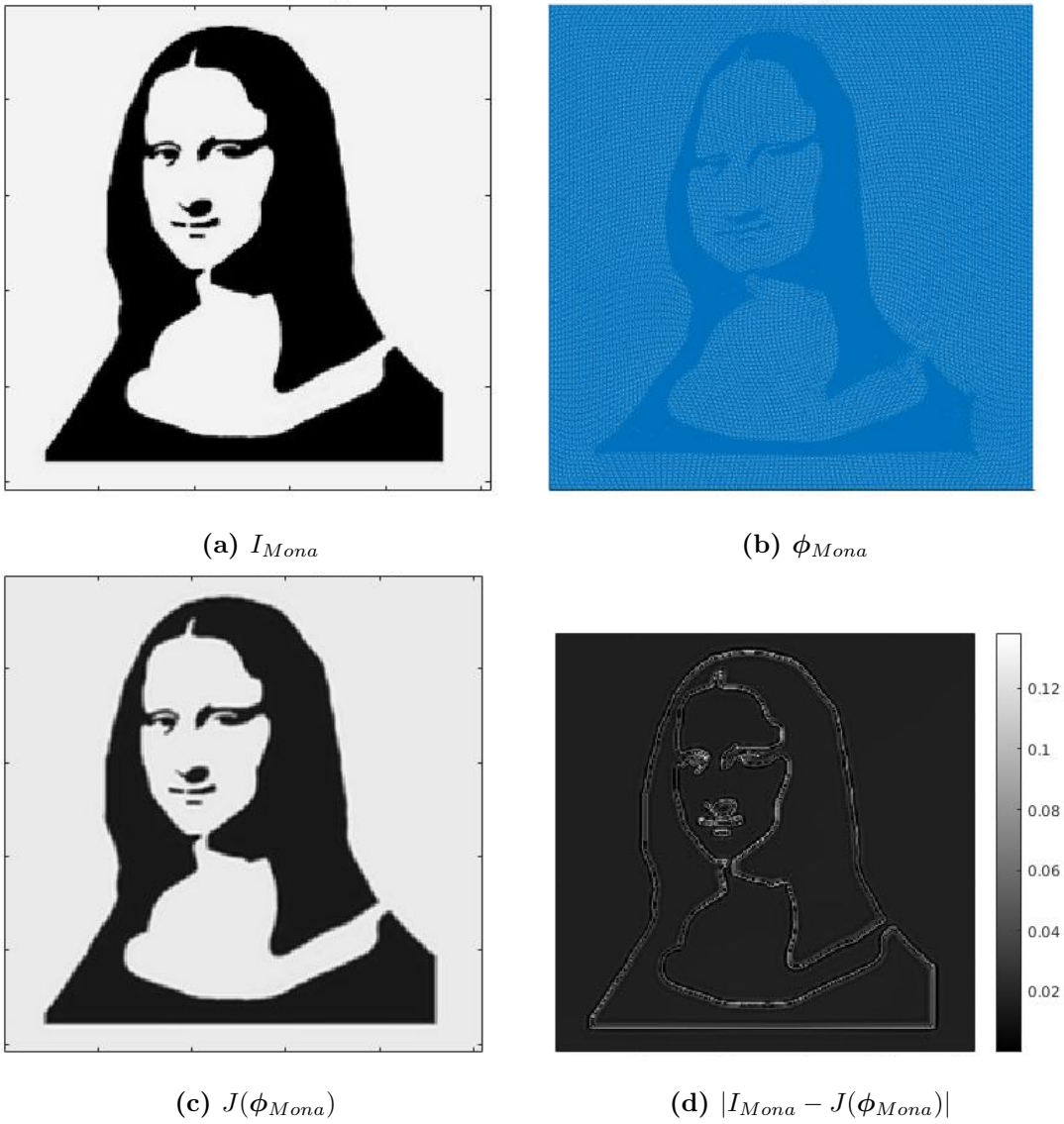
	min JD	max JD	Relative SSD	Num. Iterations	Time (sec)
ϕ_{Mona}	0.5225	1.2269	0.0062	5000	97.98
ϕ_{Pearl}	0.8381	1.7358	0.0163	5000	29.35
ϕ_J	0.8677	1.7676	0.0053	5000	110.62
ϕ_V	0.8478	1.7259	0.0053	5000	99.41

Table 4.1: Numerical summary of grid images generated by Variational Method.

We begin with an example of the Mona Lisa since this image is familiar, and easily demonstrates the accuracy of the grid generated.

See Figure 4.1 for the original Mona Lisa image (256 by 256 pixels) side-by-side with our generated grid image (horizontal and vertical step size of $h = 1$). See Figure 4.2 for an enlarged view of the grid shown in 4.1b. Due to the grid being saved in Matlab as an image and being rescaled the grid lines in the image appear quite dense, however, the minimum Jacobian determinant is positive indicating a diffeomorphic map (no mesh folding occurs; see Table 4.1). Figure 4.3 shows an enlarged view of a courser grid with slightly bigger step size ($h = 2$).

In the grids generated we clearly see that darker pigments show contracting grid cell size and lighter pigments show expanding grid cell size (black-ish areas contract;



(a) I_{Mona} **(b)** ϕ_{Mona}

(c) $J(\phi_{Mona})$ **(d)** $|I_{Mona} - J(\phi_{Mona})|$

Figure 4.1: Grid image of Mona Lisa (256×256 pixels): (a) Original image, I_{Mona}, (b) Grid image, ϕ_{Mona}, (c) Image formed by $J(\phi_{Mona})$, (d) Absolute error between I_{Mona} and $J(\phi_{Mona})$.

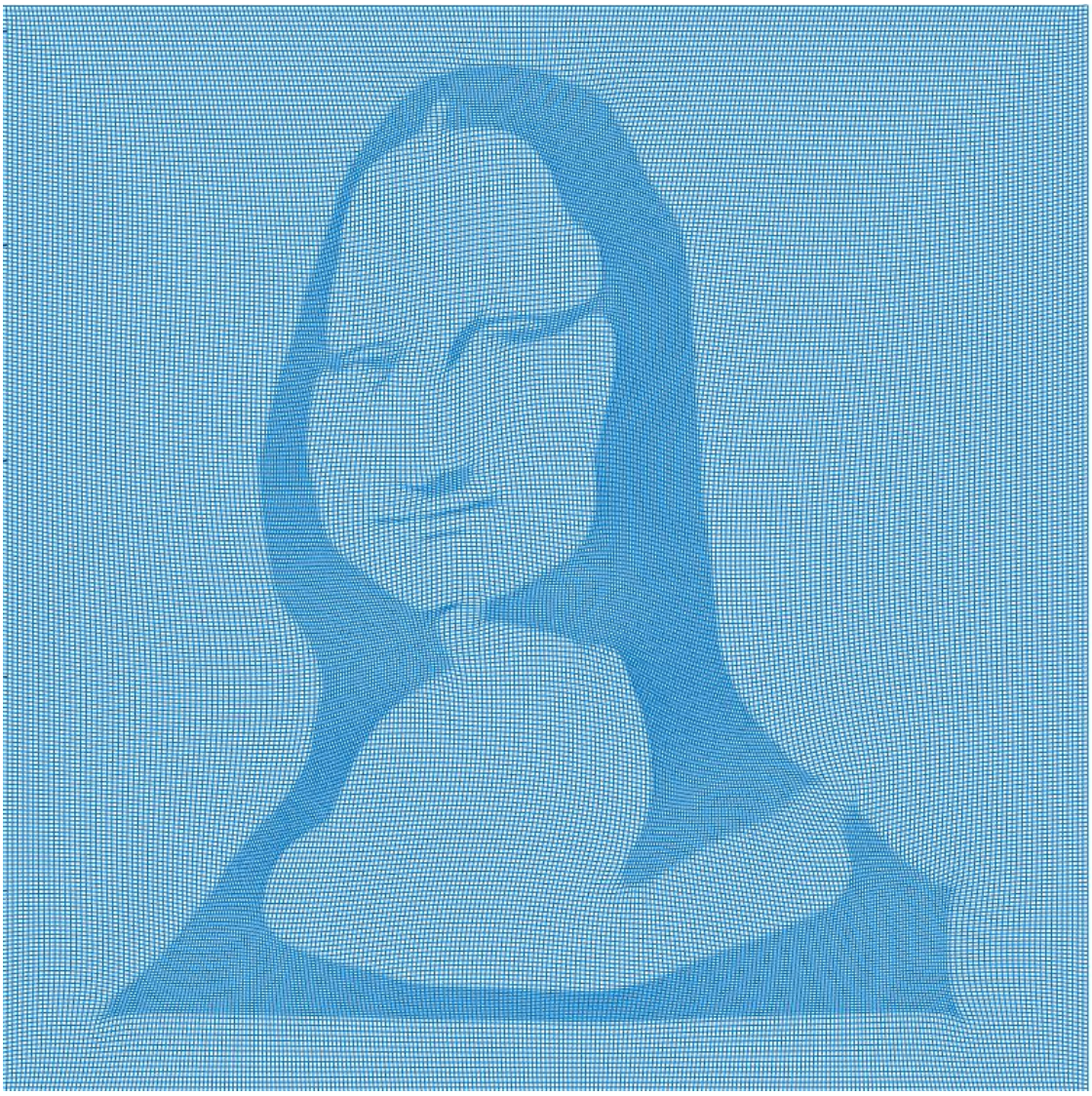


Figure 4.2: Enlarged view of Figure 4.1b based on Mona Lisa image (256 by 256 pixels) with grid step size of $h = 1$.

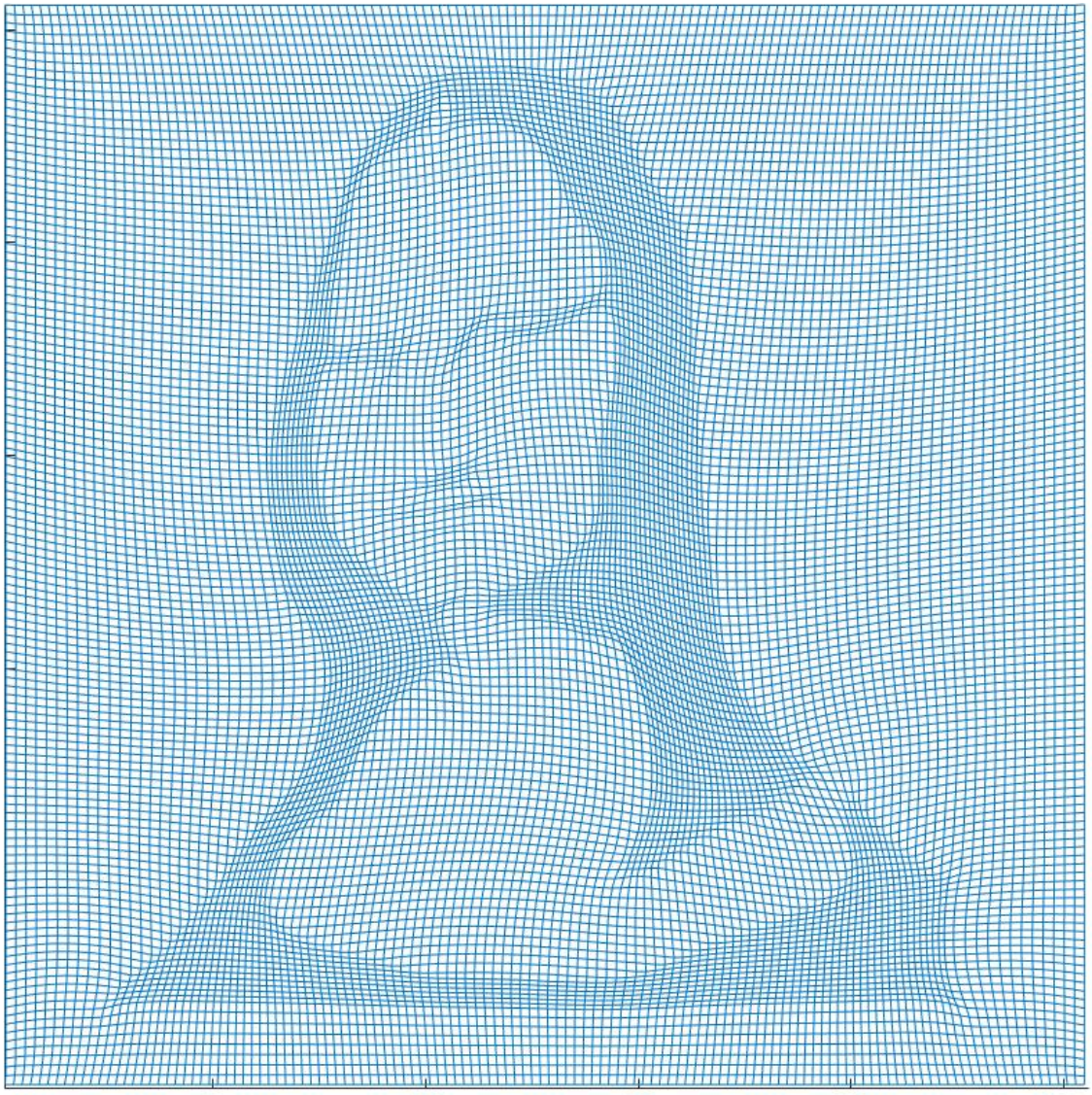
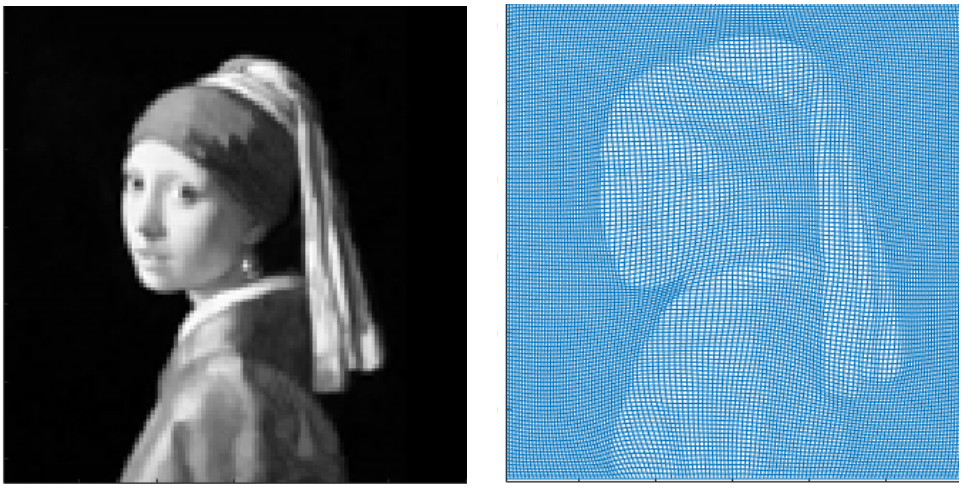


Figure 4.3: Enlarged course grid based on Mona Lisa image (256 by 256 pixels) with plotted grid step size of $h = 2$.



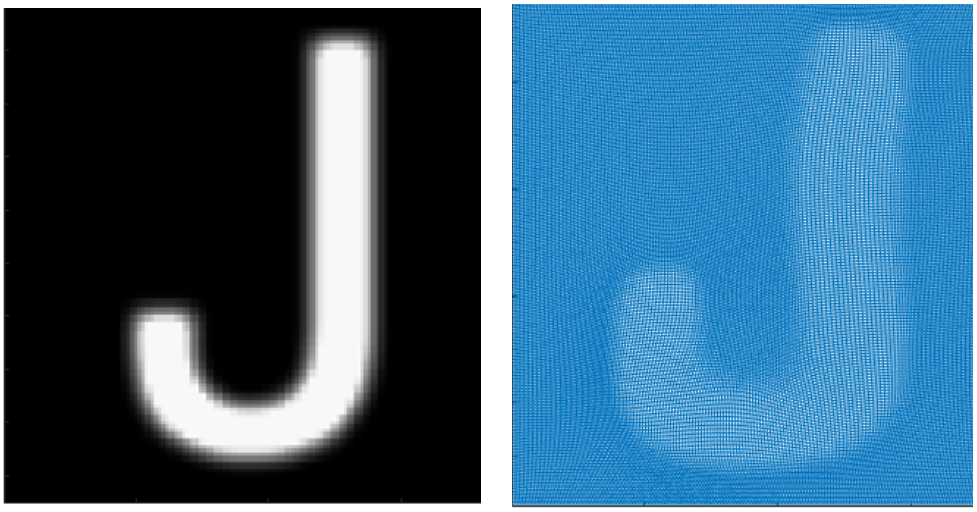
(a) I_{Pearl} **(b)** ϕ_{Pearl}

Figure 4.4: Girl with the Pearl Earing (128 by 128 pixels): (a) Original Image, I_{Pearl}, (b) Grid Image, ϕ_{Pearl}.

white-ish areas expand). Although the grid may not match the image perfectly in a visual sense, we do see strong resemblance to the input image, and all major landmarks in the image are easily seen (e.g. eyes, nose, mouth, hair, etc). Furthermore, the image can be recovered by plotting the JD of the grid ($J(\phi)$) and we see visually the accuracy of the grid's cell size correspondence to the image pixel values. Figure 4.1d shows the absolute error between pixel values and JD.

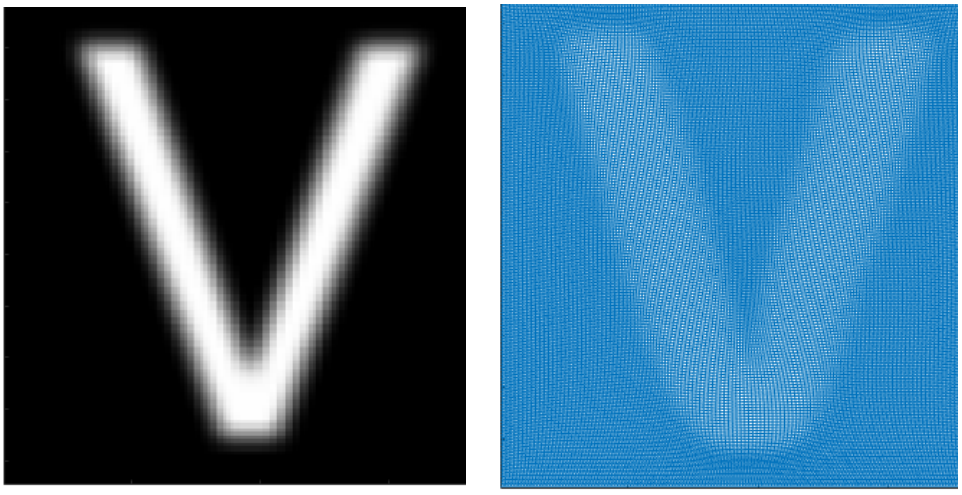
Figures 4.4, 4.5, and 4.6 show additional experiments with The Girl with the Pearl Earing; an image of the capital letter J; and an image of the capital letter V, respectively. Each grid image shows remarkable resemblance to each of the corresponding input images.

See Table 4.1 for numerical summary associated with each image-grid pair. Here the relative SSD indicates the ratio of the SSD between $J(\phi)$ and the image pixel intensity with the initial SSD between the JD of the initialized uniform grid (before



(a) I_J **(b)** ϕ_J

Figure 4.5: Letter J (192 by 190 pixels): (a) Original Image, I_J, (b) Grid Image, ϕ_J.



(a) I_V **(b)** ϕ_V

Figure 4.6: Letter V (191 by 190 pixels): (a) Original Image, I_V, (b) Grid Image, ϕ_V.

any iterations are computed) and the image pixel intensity (i.e. Relative SSD = $SSD_{final}/SSD_{initial}$).

Seeing the success of the grid images generated here, we now move on to experiments involving slices of brain images.

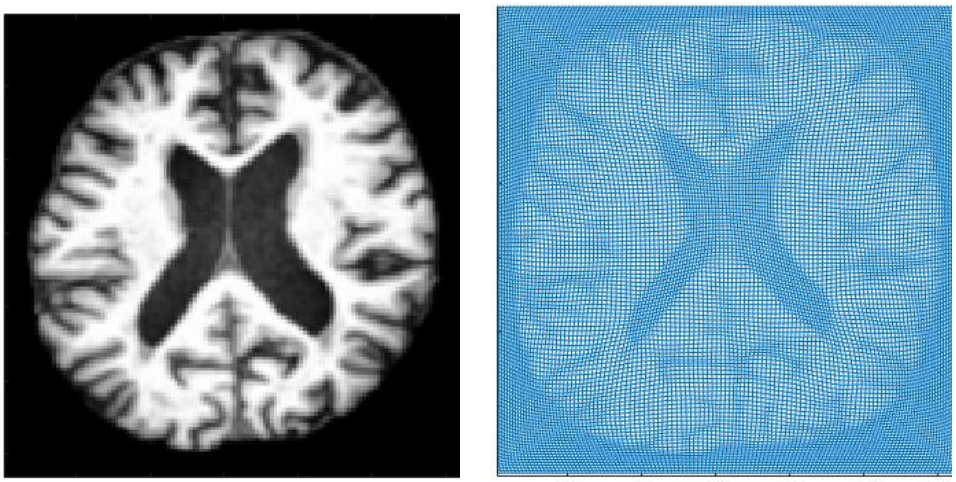
4.3.2 Experiment 2: Comparing Grid Images

Seeing the success of our grid generation with familiar images we now proceed to comparing results with grids generated in [21] and [22]. In both of these papers grids are generated via the deformation method for grid generation as discussed in chapter 2 (see Algorithm 1 for implementation). The images from which the grids were generated are 2D brain slices from the IBSR public dataset and MRBrainS18 Challenge dataset.

Using the 2D slice from the IBSR public dataset as our input image we obtain a grid with strong resemblance of the original image as expected from our previous experiment. The grid image highlights important features within the image (e.g. white matter (WM), gray matter (GM), and cerebral spinal fluid (CSF)). See Figure 4.7 for a side-by-side view.

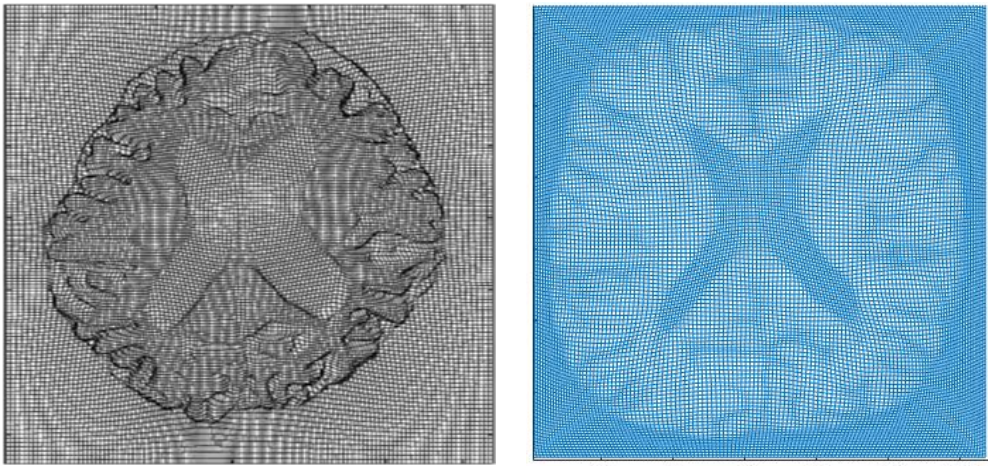
Next we compare the grid generated by our novel approach (see Figure 4.7b) with the grid generated by the deformation method of chapter 2 in Fig. 1 of [21]. In comparing these two grid images it is visually clear that the novel approach to generating grid images is an improvement to that of the deformation method. See Figure 4.8 for the side-by-side comparison.

Taking the same approach as with the sample slice from the IBSR public dataset



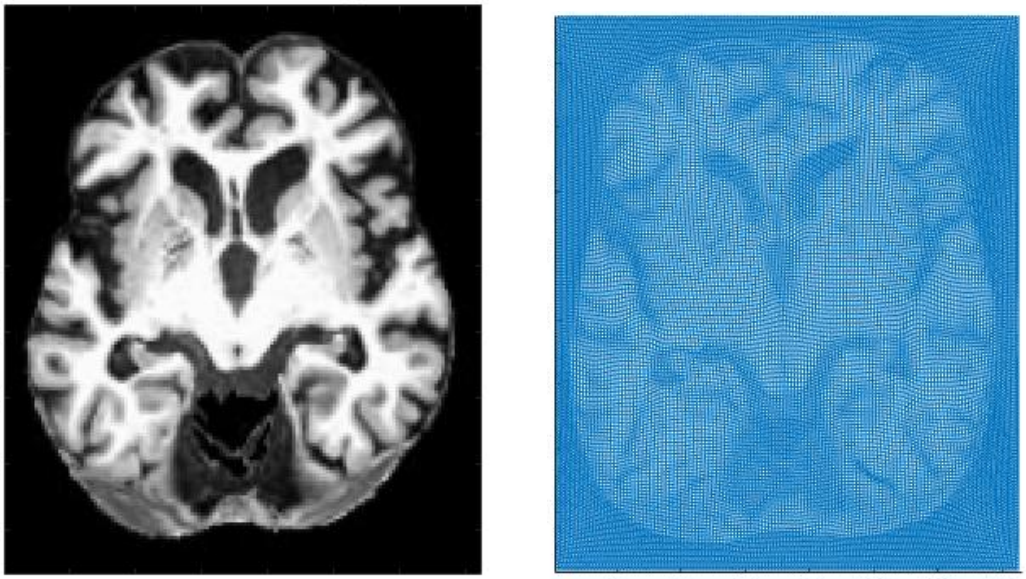
(a) I_{IBSR} **(b)** ϕ_{IBSR}

Figure 4.7: Grid generated by Variational Method based on pixel intensity of IBSR slice: (a) Original Image, I_{IBSR}, (b) Grid Image, ϕ_{IBSR}.



(a) $\phi_{IBSR\,def}$ **(b)** ϕ_{IBSR}

Figure 4.8: Comparison of grids: (a) Grid generated by deformation method in Fig. 1 of [21], $\phi_{IBSR\,def}$, (b) Grid generated by Variational Method, ϕ_{IBSR}.

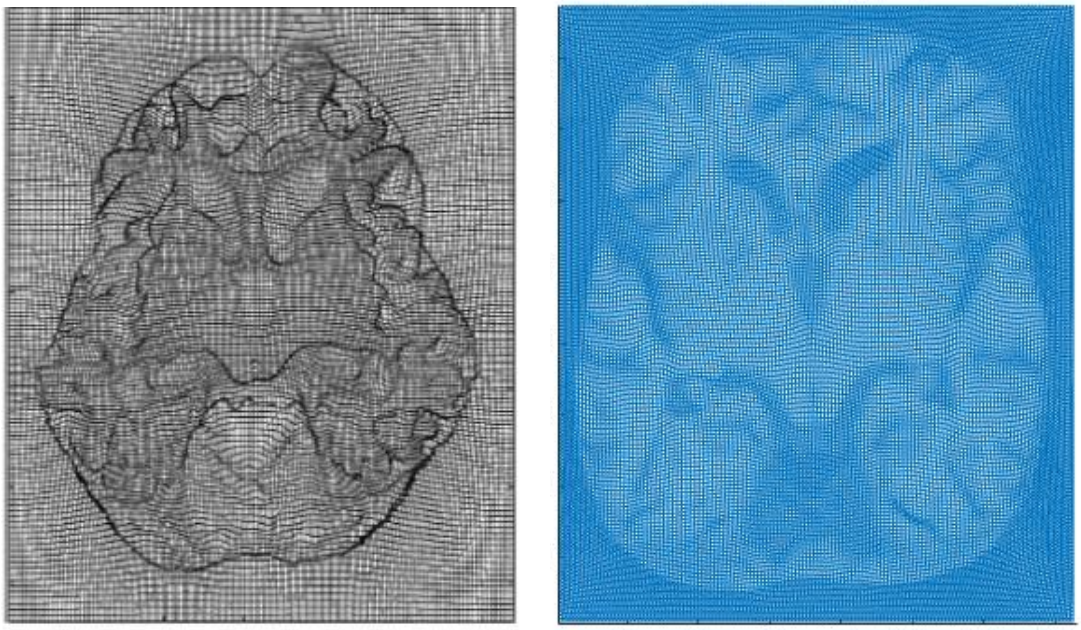


(a) $I_{MRBrainS18}$ **(b)** $\phi_{MRBrainS18}$

Figure 4.9: Grid generated by Variational Method based on pixel intensity of MRBrainS18 slice: (a) Original image, $I_{MRBrainS18}$, (b) Grid Image, $\phi_{MRBrainS18}$.

we proceed in comparing the Variational Method approach for generating the grid image with the deformation method approach for generating the grid image. We first generate the grid image using the Variational Method for the sample slice from MRBrainS18 Challenge dataset. See Figure 4.9 for a side-by-side view of the input image and grid image as generated by the Variational Method. Now comparing grids, Figure 4.10 shows a side-by-side comparison of the grid images generated by both methods.

In our comparison we again visually see that the grid image as generated by our Variational Method is an improvement to the deformation method in highlighting the features of the original brain image including WM, GM, and CSF. See Table 4.2 for



(a) $\phi_{MRBrainS18\,def}$ **(b)** $\phi_{MRBrainS18}$

Figure 4.10: Comparison of grids: (a) Grid generated by deformation method in Fig. 1 of [22], $\phi_{MRBrainS18\,def}$, (b) Grid generated by Variational Method, $\phi_{MRBrainS18}$.

Grid Image	min JD	max JD	Relative SSD	Num. Iterations	Time (sec)
ϕ_{IBSR}	0.7048	1.4746	0.0096	5000	29.43
$\phi_{MRBrainS18}$	0.7181	1.5147	0.0103	2438	43.23

Table 4.2: Numerical summary of grid images generated by Variational Method.

the numerical summary. Notice that the number of iterations for the MRBrainS18 slice is 2438. The artificial time step fell below the tolerance and stopped at that iteration.

Chapter 5

Averaging Image Registration Deformation Fields

5.1 Introduction

Currently, image registration is widely used in many areas such as computer vision, biological imaging, remote sensing, and medical imaging [18]. In particular, deformable image registration in medical imaging has been an active research topic for decades [3]. In particular, brain image analysis is based on accurate and robust registration. Unfortunately, the current methods do not provide this [18]. In this chapter we devise a new platform to combine registration deformations generated by different methods or the same method with varying parameters. Different methods such as those based on physical models, for example, the hyper-elastic or fluid flow models, make their own assumptions on underlying tissue properties. Each model has its own strengths and shortcomings. Therefore, it makes sense to combine their registration fields. In so doing, we hope that this maintains high accuracy.

Image registration is the process of aligning two or more images according to a similarity measure. Assuming that a pair of registered images differ only by Gaussian noise, registration accuracy can be evaluated by measuring pixel (voxel for 3D)

intensity difference of the registered image pair [16]. A common similarity measure used to measure pixel difference is the Sum of Squared Differences (SSD) between pixel intensity of corresponding pixels between an image pair. A perfect registration results in $SSD = 0$. Another common measure of registration accuracy is the Jaccard similarity coefficient (JSC) defined by

$$JSC = \frac{|DT_r \cap R_r|}{|DT_r \cup R_r|} \tag{5.1}$$

where DT_r and R_r represent the segmented regions of interest (landmarks within the image) in the deformed template (after registration) and the reference, respectively [18]. Essentially, the JSC identifies how well regions of interest are aligned. A perfect registration yields $JSC = 1$.

There are two general types of image registration: affine registration (i.e. alignment via rotation, reflection, scaling, translation, and shearing) and nonlinear registration. Affine registration is usually performed as a preprocessing step before nonlinear registration is performed since these methods are relatively cheap computationally. Nonlinear registration models are not so cheap computationally and there are many different nonlinear models that have been proposed.

In all the many nonlinear registration models that exist, there is no current gold-standard model. Some popular nonlinear registration methods include Diffusion model [11], Large Deformation Diffeomorphic Metric Mapping (LDDMM) [4], DARTEL [1], diffeomorphic demons [17], the hyper-elastic model [5], and standard symmetric normalization (SyN) [2]. In [18] Zhang and Chen propose a new registration method by reformulating the Lam and Lui Beltrami measure as developed in [13]

as a direct regularizer for controlling the JD. Furthermore, Zhang and Chen compare their new registration method with various other registration models (see Table 5 of Zhang and Chen [18]). From Zhang and Chen's comparison of their proposed method with other methods it's clear that different registration models produce quite different grid representations of the registration field when considering SSD, min JD, max JD, and JSC.

In [19, 20] diffeomorphic registration fields are averaged to obtain a new diffeomorphic registration field by averaging the JD and CV of the deformation fields and then prescribing these as the JD and CV in the Variational Method. That is, for N registration fields $(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \ldots, \boldsymbol{\phi}_N)$ one obtains an average deformation field $\boldsymbol{\phi}_{avg}$ by prescribing JD as $J(\boldsymbol{\phi}_{avg}) = \frac{1}{N}\sum_i J(\boldsymbol{\phi}_i)$ and CV as $curl(\boldsymbol{\phi}_{avg}) = \frac{1}{N}\sum_i curl(\boldsymbol{\phi}_i)$ in version 2 of the Variational Method with $\alpha = 1$.

In this chapter, we investigate a new application of diffeomorphic averaging. We apply the grid averaging concept to varying parameters of an image registration algorithm that relies on a parameter θ. We do this as a starting point for building a platform for averaging diffeomorphic registration fields of varying registration methods to improve robustness of grid representation of registration fields.

5.2 Review of Image Registration Field Averaging

In this section we review the image averaging process of [19, 20] based on a template (moving) image. We do this for clarification of what is meant by averaging image registration deformation fields (registration fields). The general idea is illustrated in Figure 5.1.

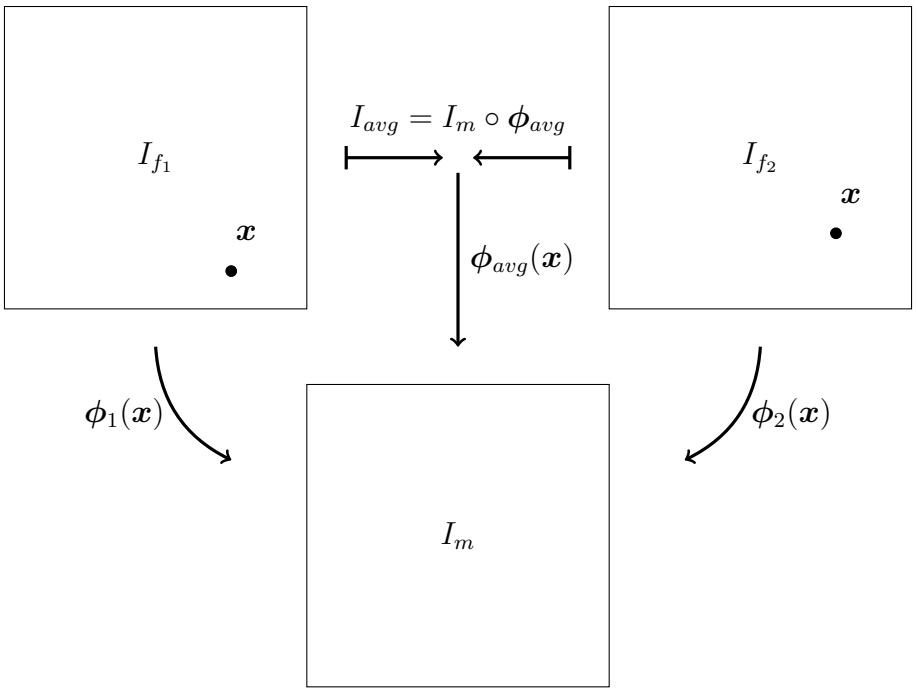


Figure 5.1: Averaging an image pair (I_{f_1} and I_{f_2}) based on template (moving) image I_m by averaging deformation fields ($\boldsymbol{\phi}_1$ and $\boldsymbol{\phi}_2$).

Given two fixed images, I_{f_1} and I_{f_2}, we desire to find an "average" image or a fusion of the two images, I_{avg}. We perform registration for the pair I_{f_1} and I_m and the registration field that registers these two images is denoted $\boldsymbol{\phi}_1$ (i.e. $I_{f_1} = I_m \circ \boldsymbol{\phi}_1$). We then perform registration again, but this time for the image pair I_{f_2} and I_m and the registration field that registers these two images is denoted $\boldsymbol{\phi}_2$ (i.e. $I_{f_2} = I_m \circ \boldsymbol{\phi}_2$). To obtain the fused images we average the registration fields $\boldsymbol{\phi}_1$ and $\boldsymbol{\phi}_2$ by version 2 of the Variational Method. That is, we prescribe JD and CV as $J(\boldsymbol{\phi}_{avg}) = \frac{1}{2}(J(\boldsymbol{\phi}_1) + J(\boldsymbol{\phi}_2))$ and $curl(\boldsymbol{\phi}_{avg}) = \frac{1}{2}(curl(\boldsymbol{\phi}_1) + curl(\boldsymbol{\phi}_2))$ and minimize SSD

$$SSD = \frac{1}{2}\int_\Omega \left(J(\boldsymbol{\phi}(\boldsymbol{x})) - f_0(\boldsymbol{x})\right)^2 + \alpha |curl(\boldsymbol{\phi}(\boldsymbol{x})) - \boldsymbol{g_0}(\boldsymbol{x})|^2 d\boldsymbol{x} \tag{5.2}$$

with $\alpha = 1$ subject to the constraint

$$\begin{cases} \Delta\boldsymbol{\phi}(\boldsymbol{x}) = \boldsymbol{f}(\boldsymbol{x}) & \text{on } \Omega \\ \boldsymbol{\phi}(\boldsymbol{x}) = \boldsymbol{x} & \text{on } \partial\Omega. \end{cases} \tag{5.3}$$

With the obtained deformation field $\boldsymbol{\phi}_{avg}$ we then have $I_{avg} = I_m \circ \boldsymbol{\phi}_{avg}$.

5.3 Experiments in Averaging Registration Fields for Varying Parameters of a Given Registration Model

In [19, 20], given N diffeomorphisms $(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \ldots, \boldsymbol{\phi}_N)$ we construct a new diffeomorphism $\boldsymbol{\phi}_{avg}$ by prescribing JD and CV with $J(\boldsymbol{\phi}_{avg}) = \frac{1}{N}\sum_i J(\boldsymbol{\phi}_i)$ and $curl(\boldsymbol{\phi}_{avg}) =$

$\frac{1}{N}\sum_i curl(\boldsymbol{\phi}_i)$, respectively, which generates what we refer to as an average deformation field. The previous section illustrates the idea for $N = 2$.

We consider a single image pair and look for two different registrations Reg_1 with registration field $\boldsymbol{\phi}_1$ and Reg_2 with registration field $\boldsymbol{\phi}_2$. We average these two registration fields and re-sample the moving image on the new average registration field $\boldsymbol{\phi}_{avg}$. In doing this we summarize some key features about the deformation field, namely, min JD, max JD, and SSD for further analysis.

5.3.1 Experiment 1

We first perform image registration with an image pair of The Girl with the Pearl Earing and a distorted version of the Girl with the Pearl Earing (see Figure 5.2). We designate the non-distorted image as the fixed image I_f and the distorted image as the moving image I_m. The registration algorithm depends on a parameter θ. For the first registration, Reg_1, we assign $\theta = 0.5$ and obtain the registration field $\boldsymbol{\phi}_1$. See Figure 5.3 for the deformation field $\boldsymbol{\phi}_1$ and the re-sampled moving image $I_m \circ \boldsymbol{\phi}_1$. See Table 5.1 for min JD, max JD, and SSD.

Girl with the Pearl Earing	min JD	max JD	SSD
$\boldsymbol{\phi}_1$	0.3952	1.7431	2277.66
$\boldsymbol{\phi}_2$	0.6114	1.7318	2275.77
$\boldsymbol{\phi}_{avg}$	0.5079	1.7347	2271.95

Table 5.1: Girl with the Pearl Earing registration field summary.

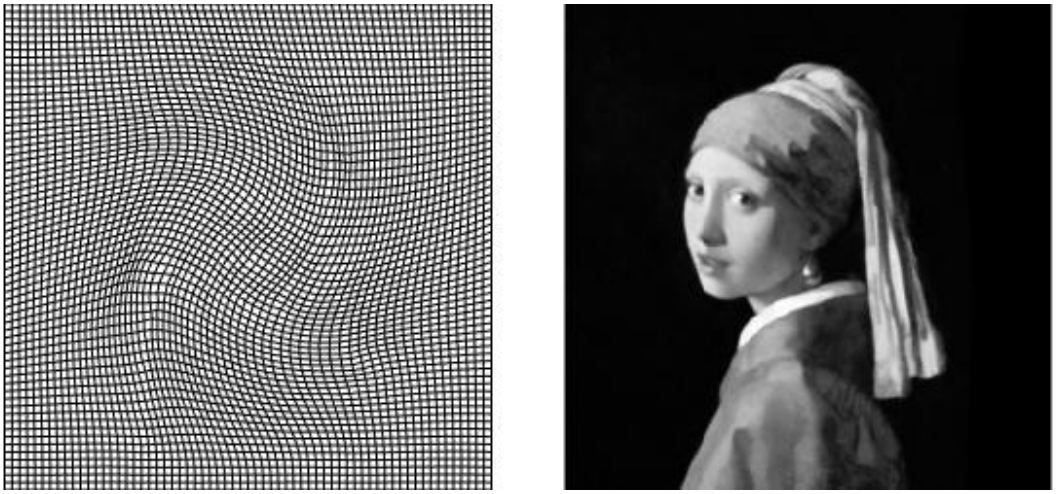
(a) I_f **(b)** I_m

Figure 5.2: Image pair to be registered: (a) Fixed image, (b) Moving image.

Next, we perform another registration, Reg_2, with parameter $\theta = 0.75$ and obtain registration field $\boldsymbol{\phi}_2$ and re-sampled moving image $I_m \circ \boldsymbol{\phi}_2$. See Figure 5.4 for the deformation field $\boldsymbol{\phi}_2$ and the re-sampled moving image $I_m \circ \boldsymbol{\phi}_2$ and Table 5.1 for min JD, max JD, and SSD.

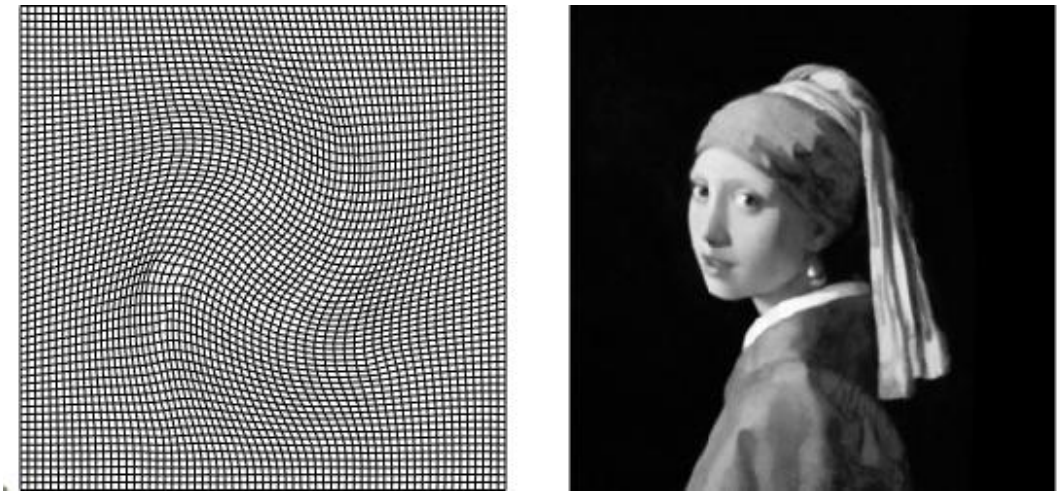
Finally, we average these two registration fields by prescribing $J(\boldsymbol{\phi}_{avg}) = \frac{\boldsymbol{\phi}_1 + \boldsymbol{\phi}_2}{2}$ and $curl(\boldsymbol{\phi}_{avg}) = \frac{curl(\boldsymbol{\phi}_1) + curl(\boldsymbol{\phi}_2)}{2}$ and then re-sample the moving image on the new average registration field $\boldsymbol{\phi}_{avg}$. See Figure 5.5 for the deformation field $\boldsymbol{\phi}_{avg}$ and the re-sampled moving image $I_m \circ \boldsymbol{\phi}_{avg}$. See Table 5.1 for min JD, max JD, and SSD.

Comparing each registration field and each re-sampling in Table 5.1 we see that the SSD actually *improves* upon *both* registration fields. However, this is not the only significance in performing this experiment. We also see that the *range* of the JD also changes. In particular, the range of the JD narrows. We note this important observation here and continue on to our next experiment.



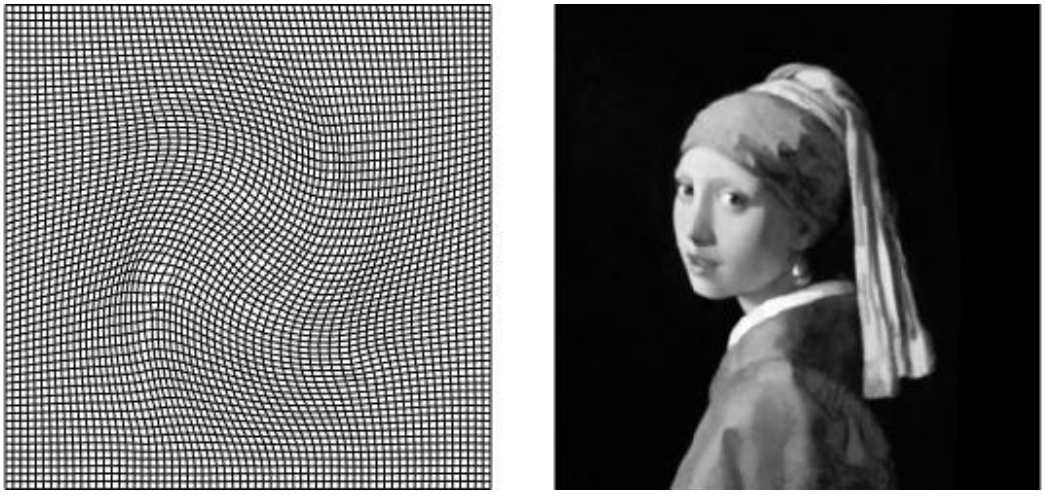
(a) ϕ_1 **(b)** $I_m \circ \boldsymbol{\phi_1}$

Figure 5.3: Girl with the Pearl Earing registration, Reg_1: (a) Registration field $\boldsymbol{\phi}_1$ generated with registration parameter $\theta = 0.5$, (b) Moving image I_m re-sampled on registration field $\boldsymbol{\phi}_1$.



(a) $\boldsymbol{\phi_2}$ **(b)** $I_m \circ \boldsymbol{\phi}_2$

Figure 5.4: Girl with the Pearl Earing registration, Reg_2: (a) Registration field $\boldsymbol{\phi}_2$ generated with registration parameter $\theta = 0.75$, (b) Moving image I_m re-sampled on registration field $\boldsymbol{\phi}_2$.



(a) ϕ_{avg} **(b)** $I_m \circ \phi_{avg}$

Figure 5.5: Girl with the Pearl Earing re-sampled on average registration field: (a) Average registration field ϕ_{avg}, (b) Moving image I_m re-sampled on registration field ϕ_{avg}.

5.3.2 Experiment 2

We perform the experiment again just as in Experiment 1, but now with a new image pair. Here we use an image pair of the Mona Lisa and the same distortion as that in Example 1 applied. Once again, the fixed image I_f is the non-distorted image and the distorted image is the moving image I_m. See Figure 5.6 for the image pair in question.

We use the same registration algorithm as before, again, Reg_1 with parameter $\theta = 0.5$ (see Figure 5.7) and Reg_2 with parameter $\theta = 0.75$ (see Figure 5.8). Finally we average the grids based on average JD and average CV just as before and re-sample the moving image I_m on the average deformation field ϕ_{avg} (see Figure 5.9). Table 5.2 shows the min JD, max JD, and SSD for each deformation field and re-sampling.

Mona Lisa	min JD	max JD	SSD
ϕ_1	0.5926	1.9438	6546.51
ϕ_2	0.1844	2.4445	6622.42
ϕ_{avg}	0.4906	1.9949	6561.91

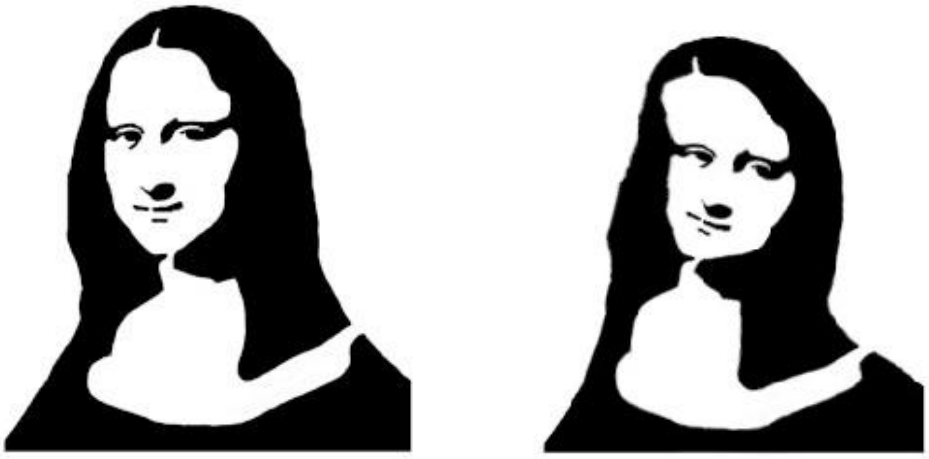
Table 5.2: Mona Lisa registration field summary.

In this second experiment we see that we improve the SSD from the second registration, but not from the first registration. However, the SSD is only slightly worse which means the accuracy is still reasonably high. The accuracy is not necessarily all we should be concerned with though.

Although accuracy is important, we also desire a registration field that is a "good" grid to represent our deformation. If we closely examine the grids generated representing each registration field, which grid is most desirable?

Let's examine the different deformations more closely. Figure 5.10 shows all three deformation grids together.

With a cursory glance we notice in Figure 5.10b that there are some grid cells that are peculiarly small and some grid cells that seem peculiarly large in areas when compared to 5.10a and 5.10b. We get a sense for this from the range of JD listed in Table 5.2. The deformation field ϕ_2 in 5.10b does not have the same cell size consistency as the other two (i.e. the distribution of the JD is not as consistent as that of ϕ_1 and ϕ_{avg}). We get a sense for this since the range of JD for ϕ_2 is larger than the other two registration fields. Investigating further, we plot the distribution for the JD of each grid in 5.11. We see in 5.11b that the distribution of JD for ϕ_2

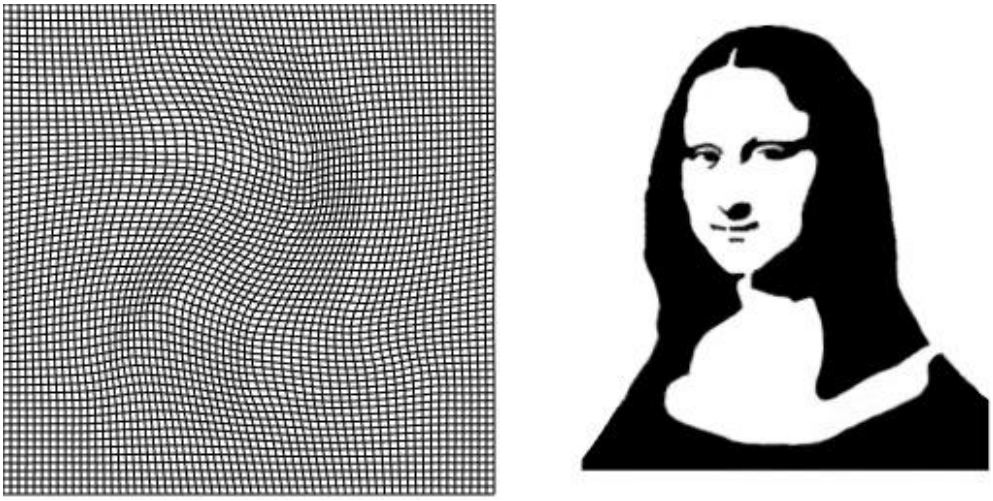


(a) I_f **(b)** I_m

Figure 5.6: Image pair to be registered: (a) Fixed image, (b) Moving image.

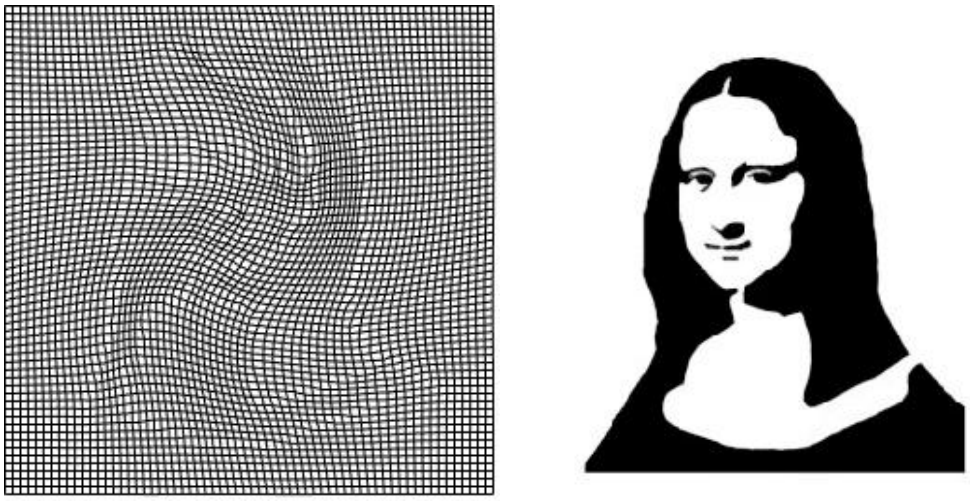
sharply drops off on the right side and has several values that continue beyond 1.5 and is *skewed right*. The distribution for JD for $\boldsymbol{\phi}_1$ in 5.11a appears to be approximately symmetric. The distribution for the JD of $\boldsymbol{\phi}_{avg}$ looks to have picked up some slack in averaging the JD of the two distributions $\boldsymbol{\phi}_1$ and $\boldsymbol{\phi}_2$. A similar phenomena occurs in our first example. That is, the range of the JD in Example 1 with the registration of the Girl with the Pearl Earing narrows after averaging just as has happened in averaging in this example has narrowed the range of the JD.

Considering the findings here and in Example 1, we propose a platform for building new deformation fields from existing ones to improve robustness in grid representations of registration fields in the next section.



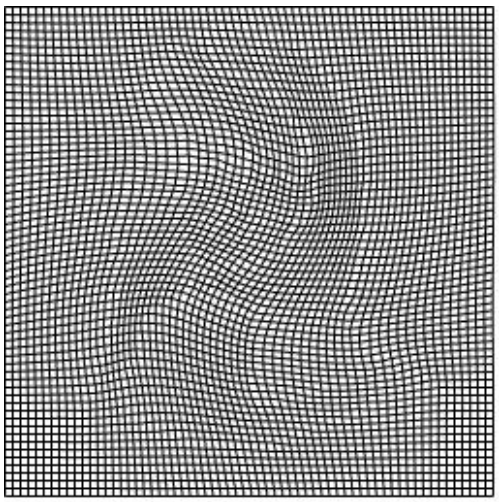
(a) ϕ_1 **(b)** $I_m \circ \phi_1$

Figure 5.7: Mona Lisa registration, Reg_1: (a) Registration field ϕ_1 generated with registration parameter $\theta = 0.5$, (b) Moving image I_m re-sampled on registration field ϕ_1.



(a) ϕ_2 **(b)** $I_m \circ \phi_2$

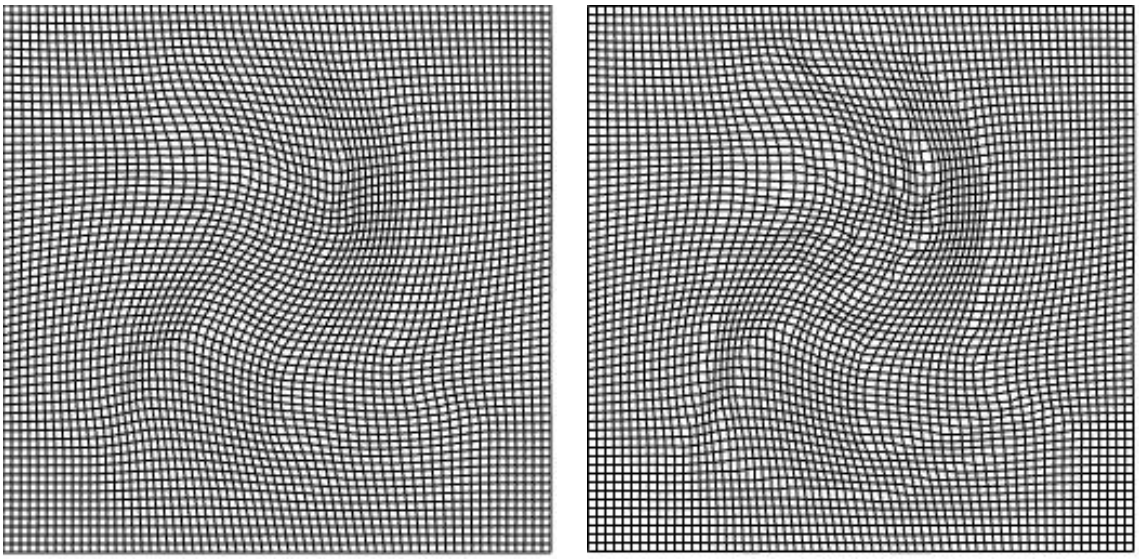
Figure 5.8: Mona Lisa registration, Reg_2: (a) Registration field ϕ_2 generated with registration parameter $\theta = 0.75$, (b) Moving image I_m re-sampled on registration field ϕ_2.



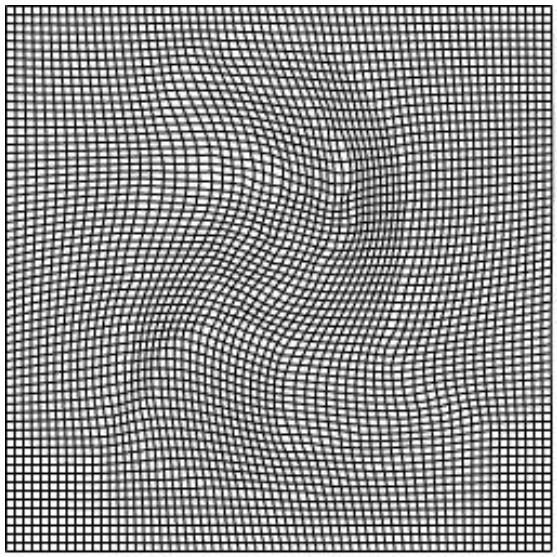
(a) ϕ_{avg}

(b) $I_m \circ \phi_{avg}$

Figure 5.9: Mona Lisa re-sampled on average registration field: (a) Average registration field ϕ_{avg}, (b) Moving image I_m re-sampled on registration field ϕ_{avg}.

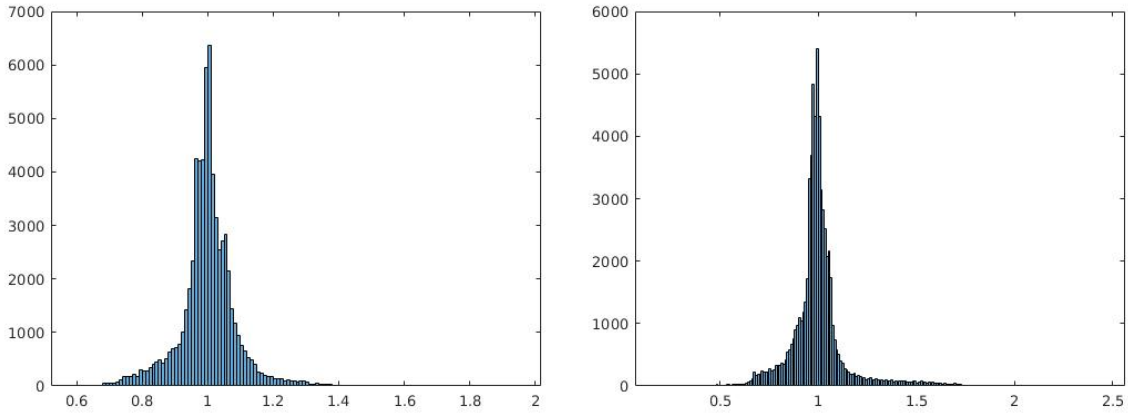


(a) ϕ_1 **(b)** ϕ_2

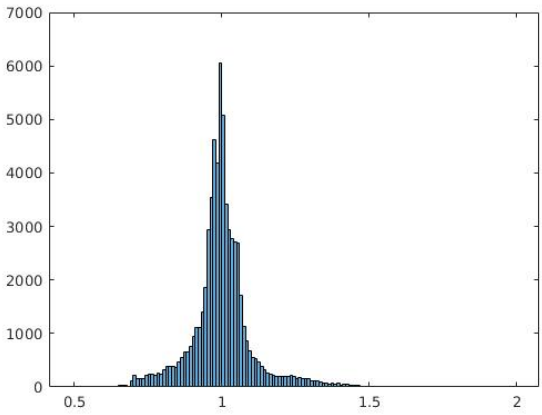


(c) ϕ_{avg}

Figure 5.10: Mona Lisa registration deformation fields: (a) Deformation field from Reg_1, (b) Deformation field from Reg_2, (c) Deformation field formed by averaging deformation fields from Reg_1 and Reg_2.



(a) $J(\phi_1)$ **(b)** $J(\phi_2)$



(c) $J(\phi_{avg})$

Figure 5.11: Distribution of JD of deformation fields for Mona Lisa registration: (a) Distribution of $J(\phi_1)$, (b) Distribution of $J(\phi_2)$, (c) Distribution of $J(\phi_{avg})$.

5.4 Construction of Robust Deformation Fields from Existing Ones

With the results of Example 1 and Example 2 of the previous section in mind, we now propose a new methodology of obtaining new registration fields from established ones, but leave the implementation of the discussion in this section for future work. Specifically, in the previous two examples, we averaged grids of an existing image registration algorithm with varying parameters and saw that the range of the JD became narrower after averaging. Furthermore, the spread of the distribution of the JD also narrowed and became more symmetric after averaging in Example 2 while retaining high accuracy. We now extend this concept to averaging established deformation fields from *differing* image registration models.

We will use Table 5 of Zhang and Chen [18] as a guide for this novel idea of constructing desirable registration grids from already constructed registration grids. Table 5.3 shows a sample of Table 5 of [18]. In [18], a new method based on Beltrami coefficient (namely, New 3) is proposed and compared to various image registration models. In their conclusion, Zhang and Chen highly recommend their New 3 model as a new robust method. Their accuracy as measured by relative SSD and JSC is quite impressive, however, examining Table 5 of Example 5.4 in [18] we notice that the min JD and max JD are quite far apart, and, therefore, this suggests the grid representation of the registration field (even though no visual of the grid itself is provided) has quite different cell size at least somewhere within the grid representation of the registration field (i.e. a less desirable grid represents the registration field). In

Registration Method	min JD	max JD	SSD (%)	JSC (%)
New 3	0.0051	49.9309	3.47	95.34
Hyper-elastic model	0.4181	3.6192	4.44	93.51
LDDMM	0.0319	20.8164	5.18	93.79
DDemons	0.1846	2.6309	18.89	87.40

Table 5.3: Sample of Table 5 of [18].

fact, for their proposed New 3 model we see from Table 5.3 that min JD is very small and max JD is quite large. Such a small min JD means the grid representation itself comes close to folding and therefore close to losing the topology in at least one region of the grid. Also, since max JD is so large there is at least one very large grid cell within the grid indicating a large local deformation within a region of the grid. The visual grid representation is likely not desirable based on this observation.

Further examining Table 5 of the same paper, we notice that min JD and max JD for say the hyper-elastic model [5] and the Large Deformation Diffeomorphic Metric Mapping (LDDMM) [4] models result in quite different grid representations of their respective deformation fields. The hyper-elastic model has a much narrower spread than both New 3 and LDDMM models for this particular registration.

Although there are other registration methods in the original Table 5 of [18], we focus on the four methods in the sample table of Table 5.3 as a starting point for our future considerations. We do not currently have the grid representations for these

particular registrations, but codes are available on-line and URLs are designated for both hyper-elastic model and LDDMM in Zhang and Chen [18].

The expectation is that averaging over the grid representations for each of the models listed in Table 5.3 should produce a narrower distribution of the JD while maintaining high accuracy and thus determining a more robust registration field than that of those constructed by the LDDMM and New 3 models.

Another consideration is averaging over a less accurate registration model with higher accuracy models (e.g. averaging the registration fields of New 3, Hyper-elastic model, and DDemons). Even though the DDemons registration algorithm has the second-worst accuracy in Table 5 of Zhang and Chen [18], perhaps the accuracy will adjust higher while the JD distribution narrows.

Further investigation is of course required, but we now have a platform for our future investigation.

Chapter 6

Conclusion

We conclude this dissertation with a summary of our new developments.

Firstly, we propose a new method for generating grids directly from images. This is done by first normalizing pixel intensity to be greater than zero: $I(\boldsymbol{x}) > 0$. Numerically, we can do this in the following way: $I_0(\boldsymbol{x}) = I(\boldsymbol{x}) + I_{max} + \varepsilon$ where $I_{max} = \max_{\boldsymbol{y}} |I(\boldsymbol{y})|, \forall \boldsymbol{y} \in \Omega$ and $\varepsilon > 0$. Once normalized, we prescribe the JD based on the normalized pixel intensity for the 2D Variational Method: $f_0(\boldsymbol{x}) = \frac{m \cdot n}{\sum_i I_0(\boldsymbol{x}_i)} \cdot I_0(\boldsymbol{x})$ for $i = 1, 2, \ldots, m \cdot n$ and $\boldsymbol{x}_i \in \Omega$. Due to the significant visual improvement of the grid images compared to those produced by the deformation method, we propose this new method of generating grid images to be used in place of the deformation method for the improvement of the image segmentation process. In addition to improved visual performance, the new variational approach for generating grid images does not rely heavily on optimization parameters as the deformation method does.

Secondly, based on promising results of experiments involving a single registration algorithm with varying parameters, we developed a platform for constructing more robust registration grids by means of version 2 of the Variational Method with prescribed JD and CV and parameter $\alpha = 1$ based on averages of existing registra-

tion fields (i.e. $J(\boldsymbol{\phi}_{avg}) = \frac{1}{N}\sum_i J(\boldsymbol{\phi}_i)$ and $curl(\boldsymbol{\phi}_{avg}) = \frac{1}{N}\sum_i curl(\boldsymbol{\phi}_i)$). The goal of this work is to level out the distribution of the JD so that the distribution has a narrower spread and ideally is approximately symmetric while maintaining high accuracy when the moving image is re-sampled on the average registration field. In performing this grid averaging we hope to produce more robust grid deformations for image registration.

Bibliography

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

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

[3] Balakrishnan, G., Zhao, A., Sabuncu, M. R., Guttag, J., and Dalca, A. V. Voxelmorph: a learning framework for deformable medical image registration. *IEEE transactions on medical imaging 38*, 8 (2019), 1788–1800.

[4] Beg, M. F., Miller, M. I., Trouvé, A., and Younes, L. Computing large deformation metric mappings via geodesic flows of diffeomorphisms. *International journal of computer vision 61*, 2 (2005), 139–157.

[5] Burger, M., Modersitzki, J., and Ruthotto, L. A hyperelastic regularization energy for image registration. *SIAM Journal on Scientific Computing 35*, 1 (2013), B132–B148.

[6] Cai, X.-X., Fleitas, D., Jiang, B., and Liao, G. Adaptive grid generation based onthe least-squares finite-element method. *Computers & Mathematics with Applications 48*, 7-8 (2004), 1077–1085.

[7] Chen, X. *Numerical Construction of Diffeomorphism and the Applications to Grid Generation and Image Registration*. Dissertation, University of Texas at Arlington, 2016.

[8] Chen, X., and Liao, G. New variational method of grid generation with prescribed jacobian determinant and prescribed curl. *arXiv preprint arXiv:1507.03715* (2015).

[9] Dacorogna, B., and Moser, J. On a partial differential equation involving the jacobian determinant. In *Annales de l'Institut Henri Poincare (C) Non Linear Analysis* (1990), vol. 7, Elsevier, pp. 1–26.

[10] Dalca, A. V., Balakrishnan, G., Guttag, J., and Sabuncu, M. R. Unsupervised learning for fast probabilistic diffeomorphic registration. In *International Conference on Medical Image Computing and Computer-Assisted Intervention* (2018), Springer, pp. 729–738.

[11] Fischer, B., and Modersitzki, J. Fast diffusion registration. *Contemporary Mathematics 313* (2002), 117–128.

[12] Fleitas, D. L. *The Least-squares Finite Element Method For Grid Deformation And Meshfree Applications*. Dissertation, University of Texas at Arlington, 2005.

[13] Lam, K. C., and Lui, L. M. Landmark-and intensity-based registration with large deformations via quasi-conformal maps. *SIAM Journal on Imaging Sciences 7*, 4 (2014), 2364–2392.

[14] Liao, G., and Anderson, D. A new approach to grid generation. *Applicable analysis 44*, 3-4 (1992), 285–298.

[15] Liao, G. G., Chen, X., Cai, X., Hildebrand, B., and Fleitas, D. A new method for triangular mesh generation. *arXiv preprint arXiv:1507.03699* (2015).

[16] Song, J. H. *Methods for evaluating image registration.* Dissertation, 2017.

[17] Vercauteren, T., Pennec, X., Perchant, A., and Ayache, N. Diffeomorphic demons: Efficient non-parametric image registration. *NeuroImage 45*, 1 (2009), S61–S72.

[18] Zhang, D., and Chen, K. A novel diffeomorphic model for image registration and its algorithm. *Journal of Mathematical Imaging and Vision 60*, 8 (2018), 1261–1283.

[19] Zhou, Z. *Image Analysis Based on Differential Operators with Applications to Brain MRIs.* Dissertation, University of Texas at Arlington, 2019.

[20] Zhou, Z., Hildebrand, B., Chen, X., and Liao, G. Computational technologies for brain morphometry. *arXiv preprint arXiv:1810.04833* (2018).

[21] Zhu, Y., Zhou, Z., Liao, G., Yang, Q., and Yuan, K. Effects of differential geometry parameters on grid generation and segmentation of mri brain image. *IEEE Access 7* (2019), 68529–68539.

[22] Zhu, Y., Zhou, Z., Liao, G., and Yuan, K. Deformable registration using average geometric transformations for brain mr images. *arXiv preprint arXiv:1907.09670* (2019).

Biographical Statement

Ben Hildebrand was born in Spokane, WA, USA, in 1983. He received his B.S. in Physics and B.S. in Mathematics from Washington State University in 2010. He received his M.S. and Ph.D. in Computational and Applied Mathematics from the University of Texas at Arlington in 2015 and 2020, respectively. During the years of his doctoral studies, he worked as a full-time math tutor at Tarrant County College, Southeast Campus until Spring 2018 when he began his full-time lecture career also at Tarrant County College, Southeast Campus. His current research interests are statistical learning and data science.