Variational Models and Numerical Algorithms for Effective Image Registration

by

Mazlinda Ibrahim

under the supervision of

Professor Ke Chen

Thesis submitted in accordance with the requirements
of the University of Liverpool for the
degree of Doctor in Philosophy.

July 2015

Contents

Acknowledgment

I would like to express my gratitude to my supervisor Prof. Ke Chen for his guidance and patience throughout my studies. Not to forget my second supervisor Dr. Bakhti Vasiev for his support and encouragement. I also would like to thank Prof. Alexander Movchan as the head of Department of Mathematical Sciences as well as Dr. Martyn Hughes for their advice during my doctoral studies.

In this very special moment, I would like to thank all the CMIT's members including Dr. Bryan Williams, Dr. Jianping Zhang, Dr. Lavdie Rada, Dr. Carlos Brito-Loeza and Jack Spencer for all the valuable time, advice, criticism and discussion. Indeed, this entire journey helped me to obtain better understanding of many valuable things in this life.

A gigantic thank you to my family: Ma, Abe Kie, Akak, Lina, Juwe, Udin, Kak Izan, Abang Yahaya, Abang Yie, Aqilah, Alia and Adwa. This thesis is dedicated to my late father Ibrahim Ismail, who passed away five months before I started my doctoral studies. In addition, my sincere appreciation to Dr. Bryan Williams, as my friend and unpaid English tutor. Not to forget Dr. Juriah Kamaludeen, Dr. Alia Ruzanna Aziz, Noor Hakim Ahmad, Aznida and Esraa Tarawneh for their support and encouragement. We had a great time in Liverpool for almost four years with exciting activities. I also would like to thank Hoo Yann Seong who is always next to me and ready to lend her ears. Thank you so much, Hoo!

I know that this thesis would have been impossible without financial support from the government of Malaysia. The research will be a starting point for the coming great research at the National Defense University of Malaysia.

Abstract

The goal of image registration is to align two or more images of the same scene obtained at different times, from different perspectives, or sensors such as MRI, X-ray and CT. This step is required to facilitate automatic segmentation for tumour detection or to inform further decisions in treatment planning. It is an important and challenging subject which usually involves high storage, computational cost and dealing with distorted and occluded data. The paradigm behind image registration is to find a reasonable transformation so that the template image becomes similar to the so-called given reference image. Through such transformation, information from these images can be compared or combined. This thesis deals with the mathematical modelling of image registration by way of energy minimisation of a functional.

We propose a new decomposition model for image registration which combines parametric transformation and non-parametric deformation. The first category of methods is based on a small number of parameters and for the second category the transformation is based on a functional map (or discretely a large number of parameters) with a regularisation term. We choose one cubic B-spline based model and the linear curvature model for the parametric and non-parametric parts respectively where the overall deformation consists of both global and local displacement for effective image registration. Some results for synthetic and real images will be presented to illustrate the effectiveness of the new model in contrast with the individual models.

We then propose a novel variational model for image registration which employs Gaussian curvature as a regulariser. The model is motivated by the surface restoration work in geometric processing [21]. An effective numerical solver is provided for the model using an augmented Lagrangian method. Numerical experiments show that the new model outperforms three competing models based on, respectively, the linear curvature [24], the mean curvature [19] and the diffeomorphic demon models [93] in terms of robustness and accuracy.

Finally, we present an improved model for joint segmentation and registration based on active contour without edges. The proposed model is motivated by an earlier model [58] and the linear curvature model [24]. Numerical results show that the new model outperforms the existing model for registration and segmentation of one or multiple objects in the image. The proposed model also leads to improved registration results when features exist inside the object.

List of Figures

List of Tables

Publications

A Decomposition Model and Its Algorithm Combining Cubic B-spline and Non-parametric Deformation for Effective Image Registration. Mazlinda Ibrahim and Ke Chen. *In preparation.*

A Unifying Framework for Decomposition Models of Parametric and Non-parametric Image Registration. Mazlinda Ibrahim and Ke Chen. *Submitted to Neurocomputing.*

A Novel Variational Model for Image Registration using Gaussian Curvature. Mazlinda Ibrahim, Ke Chen and Carlos Brito-Loeza. *Journal of Geometry, Imaging and Computing.* 1(4):417-446, 2014.

An Improved Model for Joint Segmentation and Registration. Mazlinda Ibrahim, Ke Chen and Lavdie Rada. *In preparation.*

Presentations

A Decomposition Model Combining Parametric and Non-parametric Deformation Image Registration.

22th International Symposium on Mathematical Programming, 12-17 July 2015, Pittsburgh.

A Composition Model Combining Parametric Transformation and Non-parametric Deformation for Effective Image Registration.

SIAM Conference on Imaging Science, 12-14 May 2014, Hong Kong.

Fourth Order Variational Formulation for Image Registration.

25th Biennial Conference in Numerical Analysis, 25-28 June 2013. Glasgow.

An Improved Model for Joint Segmentation and Registration based on Linear Curvature Smoother .

3rd International Workshop on Image Processing Techniques and Applications, incorporating Mathematical Imaging with Biomedical Application, 6-8 July 2015. Liverpool.

A New Variational Model for Image Registration using Gaussian Curvature.

LMS Inverse Day on Tomographic Reconstruction from Boundary Data, 22 September 2014. Leeds.

Improved Parametric and Non-parametric Image Registration.

LMS Inverse Day on Hybrid and Multi-modal Imaging, 4 July 2014. Manchester.

Chapter 1

Introduction

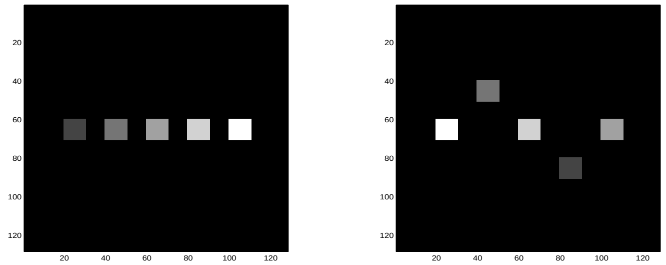
Image registration, also known as warping, fusion, motion correction or co-registration is one of the most difficult tasks among medical imaging applications. Another challenging task is image segmentation. Registration aims to automatically align images and establish correspondences between features within images which display different views of the same objects. Such images may be taken from different individuals, at different times or from different imaging machineries. After successful registration, information from different images may be compared, combined or fused for further tasks. There exits a large number of application areas for image registration including computational anatomy [1], computer aided diagnosis [92], radiation therapy [100], treatment verification [27], CCTV [76] and remote sensing [61].

For example, in medical imaging, a radiologist may be asked to combine information from computer tomography (CT) and photon emission tomography (PET) where the former modality contains patient anatomy information such as bones and organs while the latter modality is used to scan the functional data such as glucose uptake. The CT imaging process requires the patient to rest his arms due to space limitation in the CT tube. For PET scanning, the patient needs to lift his arm to minimise attenuation of the tracer. Image registration aims to align these two data of data into a unified spatial alignment. The job becomes harder since there is no motion model for each patient and internal organs move according to individuals. In [8], the authors mention that quantification and evaluation of registration results are difficult because there is not much information about the ground truth for the registration of medical images.

This thesis is about developing mathematical models for image registration. We restrict ourselves to variational or energy minimisation methods because of their flexibility. Such methods have received a lot of attention in medical imaging. In this chapter, we present an introduction to image registration with a brief discussion on how to model the problem as a minimisation problem. We also present an outline of the chapters of this thesis.

1.1 Introduction to Image Registration

Image registration is the process of finding geometric transformation between two given images respectively known as the template (target) and reference (source) images. The recovered transformation may be applied to the template image so that it becomes similar in some sense to the reference image. It is assumed that the template image is a deformed version of the reference image. Image registration has broad applications ranging from medical image analysis, video surveillance, remote sensing and satellite imagery. For mono-modality image registration, images are generated from the same imaging machinery. Thus, the same objects or features in the images are represented by the same intensity values. In this case, finding the optimal transform which links the two images remains a challenging subject. Meanwhile, for multi-modality image registration, images are generated from different imaging machineries. Thus, the same objects or patterns will have different intensity values as shown in Figure 1.1. Consequently, matching patterns is a challenge.



(a) Reference image, R.

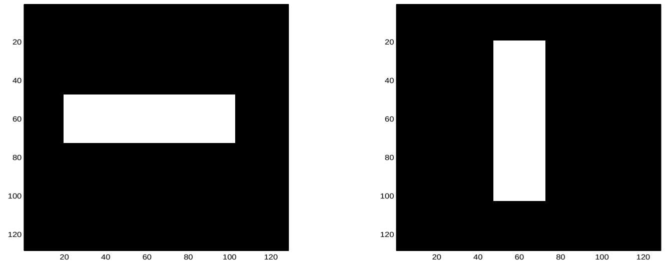
(b) Template image, T.

Figure 1.1: Illustration of reference and template images for multi-modality image registration. The same objects in the reference (a) and template images (b) have different intensity values.

There are several different ways of classifying intensity based registration methods. The methods can be divided into physical categories as rigid and non-rigid, mathematical categories as linear and nonlinear or complexity categories as parametric and non-parametric. We will follow the work done by Modersitzki [67] who classified the problem into two classes by complexity. The first class of problems is called parametric image registration, for which transformations are dependent on a finite number of parameters. As such, rigid and affine registration are parametric models where the former consists of three parameters and the latter has six parameters for two dimensional image registration. We are interested in the second class of problems which consists of the non-parametric image registration models. This class is based on functional minimisation or the variational approach. In this particular approach, we are looking for deformation (displacement) fields for every pixel in the image. It is usually based on physical processes such as elastic, motion curvature or fluid flow.

Since the goal of image registration is to transform images such that they become similar, modelling the problem involves the minimisation of a dissimilarity functional. The functional can either be a function of the image intensity difference or a distance between landmarks or markers in the images. In the former case, the registration process is given implicitly by the whole image and in the latter case, only some parts of the image are used to identify the deformation field.

In order to illustrate mono-modality image registration, we refer to Figure 1.2 where the reference image (a) consists of a rectangle and, in the template image (b), this rectangle is rotated. In image registration, we are looking for the transformation which may be applied to the rectangle in T, in order to transfigure it to be similar to the reference image R. We can use the following dissimilarity or similarity measure which is the sum of the squared difference (SSD),



(a) Reference image, R.

(b) Template image, T.

Figure 1.2: Illustration of reference and template images. The template image (b) is a rotated version of the reference image (a).

$$\mathcal{D}^{\text{SSD}}(T, R, \boldsymbol{\varphi}(x_1, x_2)) = \frac{1}{2}\int_\Omega (T(\boldsymbol{\varphi}(x_1, x_2)) - R(x_1, x_2))^2 \, \mathrm{d}\Omega \tag{1.1}$$

where $\boldsymbol{\varphi}$ is the transformation, $\mathrm{d}\Omega = \mathrm{d}x_1 \, \mathrm{d}x_2$ and Ω is the image domain. The distance in equation (1.1) is the L_2 norm and it should be noted that minimising (1.1) with respect to the transformation $\boldsymbol{\varphi}$ is an ill-posed problem. This is because the solution given by

$$\min_{\boldsymbol{\varphi}} \mathcal{D}^{\text{SSD}}(T, R, \boldsymbol{\varphi}) \tag{1.2}$$

is not unique. Referring to Figure 1.2, the solution can be either a rotation of $\frac{\pi}{2}$ or $-\frac{3\pi}{2}$ about the centre of the image. Thus, we need a priori information about the transformation so that unwanted solutions may be penalised. Using the well-known Tikhonov regularisation, we incorporate a regularisation term $\mathcal{S}(\boldsymbol{\varphi})$ into (1.1). The joint energy minimisation functional corresponding to (1.1) consists of a weighted sum including fitting and regularisation terms as follows,

$$\min_{\boldsymbol{\varphi}} \left\{ \mathcal{J}(T, R, \boldsymbol{\varphi}) = \mathcal{D}^{\text{SSD}}(T, R, \boldsymbol{\varphi}) + \gamma \mathcal{S}(\boldsymbol{\varphi}) \right\} \tag{1.3}$$

where γ is a positive constant parameter which balances the trade-off between the fitting and regularisation energies. This thesis focuses on mathematical models of image registration based on this type of variational approach.

1.2 Thesis Outline

The remaining chapters of this thesis are organised as follows.

Chapter 2 - Mathematical Preliminaries

In this chapter, we present some mathematical tools which will be used throughout this thesis and which the reader may wish to consult while reading subsequent chapters. A brief review is given of definitions, theorems and examples of some important and relevant mathematical topics including normed linear spaces, variations of a functional, functions of bounded variation, inverse problems and regularisation, the discretisation of partial differential equations using finite differences, and iterative solutions of linear and nonlinear equations. Finally, we will conclude this chapter with an introduction to the multigrid method for elliptic PDEs.

Chapter 3 - Mathematical Models for Image Registration and Segmentation

In this chapter, we present a brief review of mathematical models for image registration. We start with the mathematical setting for an image and the mathematical formulation for image registration. We introduce similarity measures and parametric image registration models. Then, we cover some existing models for non-parametric image registration based on the variational formulation such as the linear elastic, nonlinear elastic and fluid registration techniques. We also introduce and describe a solution scheme for the minimisation of such models. We finally present some work in image segmentation which is useful for Chapter 7.

Chapter 4 - A Decomposition Model Combining Parametric and Non-parametric Deformation

In this chapter, we introduce a decomposition model for mono-modality image registration which combines parametric transformation and non-parametric deformation. We choose cubic B-spline and linear curvature to model parametric and non-parametric transformations respectively. Numerical results are presented at the end which show that the decomposition model outperforms individual models.

Chapter 5 - Multi-modality Image Registration using the Decomposition Model

In this chapter, we extend the decomposition model of parametric and non-parametric transformations introduced in Chapter 4 to multi-modality images. In this case, the reference and template images come from different imaging modalities. For example, the reference image may be a computer tomography (CT) scan which is good for the quantification of cancerous tissues for determining the dose calculation in treatment planning and the template image may be a magnetic resonance (MRI) image which is much better for the visualisation of soft tissues compared to the CT scanner. Given the very different resulting images, the intensity values are not directly comparable and so the use of traditional similarity measures such as the sum of the squared difference is no longer valid. We explore two similarity measures for multi-modality images given by mutual information and the normalised gradient field in order to build registration models for such images. We use three data sets in order to test these two similarity measures with the decomposition model.

Chapter 6 - A Novel Variational Model for Image Registration using Gaussian Curvature

In this chapter, we propose a novel regularisation term for non-parametric image registration based on the Gaussian curvature of the surface obtained from the displacement field. Direct solution of the resulting Euler-Lagrange PDEs is difficult due to high nonlinearity. We provide a numerical solver for the model using the augmented Lagrangian method. Numerical experiments are shown to illustrate the performance of the proposed model in comparison with the linear curvature, diffeomorphic demon and mean curvature models. Our new model turns out to be more robust than these three competing models for the tested images because Gaussian curvature is a more natural physical quantity for a surface compared to linear and mean curvatures.

Chapter 7 - An Improved Model for Joint Segmentation and Registration

In this chapter, we present an improved model for joint segmentation and registration using the active contour without edges method for segmentation and the linear curvature model for image registration. The proposed model improves on the original Guyader and Vese model [58] for image segmentation and registration. Here we use the Chan-Vese model for segmentation [12] where the image is modelled as a piecewise constant function.

Chapter 8 - Conclusion and Future Research

In the final chapter, we present our conclusions and outline possible future research directions arising from the work presented in this thesis.

Chapter 2

Mathematical Preliminaries

This chapter covers some basic mathematical tools that will be used throughout the thesis. We begin with an introduction to normed linear spaces with some useful examples. Next, we review some relevant theory about calculus of variation. We discuss inverse problems and regularisation before moving on to the discretisation of partial differential equations (PDEs) by finite difference methods and define some notation. Finally, we discuss iterative methods and iterative solutions of nonlinear equations, finishing with an introduction to multigrid methods.

2.1 Normed Linear Spaces

Definition 2.1.1 ***Linear Vector Space***. *Let* $\mathbf{V}$ *be a set with the operations of multiplication and addition defined. Let* $\mathbf{u}$ *and* $\mathbf{v}$ *be any two elements of* $\mathbf{V}$, $\mathbf{u}, \mathbf{v} \in \mathbf{V}$. *Let the sum of these two elements be denoted by* $\mathbf{u} + \mathbf{v}$ *and the scalar multiplication of* $\mathbf{u}$ *with an element* $c \in \mathbf{F}$ *of a scalar field* $\mathbf{F}$ *by* $c\mathbf{u}$. *Then* $\mathbf{V}$ *is called a vector space of a scalar field* $\mathbf{F}$ *if all of the following ten axioms are satisfied.*

1. *Closure under addition:* $\mathbf{u} + \mathbf{v} \in \mathbf{V}$.

2. *Commutativity under addition:* $\mathbf{u} + \mathbf{v} = \mathbf{v} + \mathbf{u}$.

3. *Associativity under addition:* $(\mathbf{u} + \mathbf{v}) + \mathbf{w} = \mathbf{u} + (\mathbf{v} + \mathbf{w})$ *for all* $\mathbf{w} \in \mathbf{V}$.

4. *Existence of an identity element of addition: There exists an element* $\mathbf{0} \in \mathbf{V}$, *such that* $\mathbf{0} + \mathbf{u} = \mathbf{u}$ *for all* $\mathbf{u} \in \mathbf{V}$.

5. *Existence of additive inverse: For all* $\mathbf{u} \in \mathbf{V}$, *there exists an element* $-\mathbf{u} \in \mathbf{V}$, *such that* $\mathbf{u} + -\mathbf{u} = \mathbf{0}$.

6. *Closure under scalar multiplication: For* $c \in \mathbf{F}$, *we have* $c\mathbf{u} \in \mathbf{V}$.

7. *Distributivity: If* $\mathbf{u}, \mathbf{v} \in \mathbf{V}$ *and* $c \in \mathbf{F}$, *then* $c(\mathbf{u} + \mathbf{v}) = c\mathbf{u} + c\mathbf{v}$.

8. *Distributivity under scalar multiplication: If* $c, d \in \mathbf{F}$, *then* $(c + d)\mathbf{u} = c\mathbf{u} + d\mathbf{u}$.

9. *Associativity under scalar multiplication: If* $c, d \in \mathbf{F}$, *then* $c(d\mathbf{u}) = (cd)\mathbf{u}$.

10. *Existence of an identity element of scalar multiplication: There exists an element* $1 \in \mathbf{F}$, *such that* $1\mathbf{u} = \mathbf{u}$ *for all* $\mathbf{u} \in \mathbf{V}$.

Example 2.1.2 *The linear vector spaces denoted by*

- $\mathbb{R}^d$ *and* $\mathbb{C}^d$ *for all* $d \in \mathbb{N}$,
- $\mathbf{F}[x]$*, a polynomial function over a field* $\mathbf{F}$,

are normed linear spaces. We can verify these using Definition 2.1.

Theorem 2.1.3 ***Linear Subspace****. A subset* $\mathbf{W}$ *of a vector space* $\mathbf{V}$ *is a linear subspace if and only if it satisfies the following three properties:*

1. *There exists* $\mathbf{0} \in \mathbf{W}$.
2. $\mathbf{W}$ *is closed under addition.*
3. $\mathbf{W}$ *is closed under scalar multiplication.*

Proof *The first property ensures that* $\mathbf{W}$ *is not a null set since there exists at least one element of* $\mathbf{W}$*. From the definition of vector space and since an element of* $\mathbf{W}$ *is also an element of* $\mathbf{V}$*, the vector space operations are well defined.*

Definition 2.1.4 ***Norm and Seminorm****. A norm* $\|\cdot\|$ *on a vector space* $\mathbf{V}$ *is a function* $\|\cdot\| : \mathbf{V} \to \mathbb{R}$ *that satisfies the following properties for all* $\mathbf{u}, \mathbf{v} \in \mathbf{V}$ *and* $\lambda \in \mathbf{F}$ *where* $\mathbf{F}$ *is a scalar field.*

1. *Faithfulness:* $\|\mathbf{u}\| = 0$ *if and only if* $\mathbf{u} = 0$ *and* $\|\mathbf{u}\| > 0$ *if* $\mathbf{u} \neq 0$.
2. *Homogeneity:* $\|\lambda\mathbf{u}\| = |\lambda|\|\mathbf{u}\|$.
3. *Subadditivity:* $\|\mathbf{u} + \mathbf{v}\| \leq \|\mathbf{u}\| + \|\mathbf{v}\|$.

A semi-norm is defined similarly as above but with the first property being replace by $\|\mathbf{u}\| \geq 0$*. A normed space is a linear vector space with a norm defined on it. For a normed linear space, the distance (also known as a metric) between two elements* $\mathbf{u}, \mathbf{v} \in \mathbf{V}$ *is denoted by* $d(\mathbf{u}, \mathbf{v}) = \|\mathbf{u} - \mathbf{v}\|$.

Example 2.1.5 ***p-norm****. Let* $\mathbf{x} \in \mathbb{R}^n$*, then for any real number* $p \geq 1$ *we define the p-norm of* $\mathbf{x}$ *as*

$$\|\mathbf{x}\|_p = \left(\sum_{i=1}^{n} |x_i|^p\right)^{\frac{1}{p}}.$$

For $p = 2$*, we recover the Euclidean norm defined by*

$$\|\mathbf{x}\|_{\mathbb{R}^d} = \sqrt{\mathbf{x} \cdot \mathbf{x}} = \sqrt{\sum_{i=1}^{n} x_i^2}.$$

The Euclidean scalar product is denoted by $\mathbf{x} \cdot \mathbf{y}$ *and is defined by*

$$\mathbf{x} \cdot \mathbf{y} = \|\mathbf{x}\|\|\mathbf{y}\| \cos\theta$$

where θ *is the measure of the angle between* $\mathbf{x}$ *and* $\mathbf{y}$.

Example 2.1.6 ***L^p-norm or L_p-norm*** *. Let f be a function defined on a domain Ω and $1 \leq p \leq \infty$. We define the L^p-norm of f on Ω as*

$$\|f(x)\|_p = \left(\int_\Omega |f(x)|^p \, \mathrm{d}x \right)^{\frac{1}{p}} .$$

Note that since f may have arbitrarily many components, this is a generalisation of Example 2.1.5.

Example 2.1.7 ***L^∞-norm****. The special case of the L^p-norm from Example 2.1.5 where $p = \infty$ is defined as*

$$\|f(x)\|_\infty = \sup_x |f(x)|. \tag{2.1}$$

Definition 2.1.8 ***Inner Product****. An inner product on a linear vector space $\mathbf{V}$ defined over the scalar field $\mathbf{F}$ is a function $\langle \cdot, \cdot \rangle_{\mathbf{V}}$ defined on $\mathbf{V} \times \mathbf{V}$ which satisfies the following properties:*

1. *Positive definiteness: $\langle \mathbf{u}, \mathbf{u} \rangle_{\mathbf{V}} > \mathbf{0}$ for $\mathbf{u} \neq \mathbf{0}$ and $\mathbf{u} \in \mathbf{V}$.*
2. *Conjugate symmetry: $\langle \mathbf{u}, \mathbf{v} \rangle_{\mathbf{V}} = \overline{\langle \mathbf{v}, \mathbf{u} \rangle_{\mathbf{V}}}$ for all $\mathbf{u}, \mathbf{v} \in \mathbf{V}$.*
3. *Linearity under scalar multiplication: $\langle \lambda \mathbf{u}, \mathbf{v} \rangle_{\mathbf{V}} = \lambda \langle \mathbf{u}, \mathbf{v} \rangle_{\mathbf{V}}$ for all $\mathbf{u}, \mathbf{v} \in \mathbf{V}$ and $\lambda \in \mathbf{F}$.*
4. *Linearity under vector addition: $\langle \mathbf{u} + \mathbf{v}, \mathbf{w} \rangle_{\mathbf{V}} = \langle \mathbf{u}, \mathbf{v} \rangle_{\mathbf{V}} + \langle \mathbf{v}, \mathbf{w} \rangle_{\mathbf{V}}$ for all $\mathbf{u}, \mathbf{v}, \mathbf{w} \in \mathbf{V}$.*

Example 2.1.9 *Examples of inner products are*

- *The standard inner product*

  $$\langle \mathbf{x}, \mathbf{y} \rangle_{\mathbb{R}^d} = \mathbf{y}^\top \mathbf{x} = \sum_{i=1}^{d} x_i y_i$$

  for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$.

- *If $C[a, b]$ is the vector space of real-valued continuous functions defined on the interval (a, b), then*

  $$\langle f, g \rangle = \int_a^b f(t) g(t) \ \mathrm{d}t$$

  is an inner product on $C[a, b]$.

Definition 2.1.10 ***Support of a Function****. Let Ω be a nonempty open set in $\mathbb{R}^n$, and let f be a continuous real or complex valued function on Ω. The support of f is defined as*

$$supp(f) = \overline{\{x \in \Omega : f(x) \neq 0\}}. \tag{2.2}$$

Definition 2.1.11 ***Compactly Support****. A function f in $\Omega \subset \mathbb{R}^n$ has compact support in Ω if supp(f) is a compact set in Ω which also implies that supp(f) is a closed set in $\mathbb{R}^n$.*

Definition 2.1.12 ***Cauchy Sequence and Completeness***. *A sequence $\{\mathbf{x}_i\}_{i\in\mathbb{N}}$ in a normed vector space $\mathbf{V}$ is called a Cauchy sequence if for any real number, $\epsilon > 0$, there exists $M \in \mathbb{Z}^+$ such that for every natural number $m, n > M$, we have $\|\mathbf{x}_m - \mathbf{x}_n\| < \epsilon$. A normed vector space is said to be complete if every Cauchy sequence converges.*

Definition 2.1.13 ***Banach Space***. *A Banach space is a complete normed vector space.*

Example 2.1.14 *An example of a Banach space is*

- *$C[a,b]$, the space of continuous, real-valued or complex valued functions with the norm*
$$\langle f,g\rangle = \int_a^b f(t)g(t)\ \mathrm{dt}.$$

Definition 2.1.15 ***Hilbert Space***. *An inner product space which is complete with respect to the norm induced by the inner product is called a Hilbert space.*

Example 2.1.16 *Two relevant examples of Hilbert spaces are the space $\mathbb{R}^n$ together with the Euclidean inner product and the space $L^2(\Omega)$ together with the inner product defined by*
$$\langle f,g\rangle_{L^2(\Omega)} = \int_\Omega f(\mathbf{x})g(\mathbf{x})\,\mathrm{d}\boldsymbol{x}. \tag{2.3}$$

Definition 2.1.17 ***Linear Operator***. *Take two vector spaces $\mathbf{V}$ and $\mathbf{W}$. A mapping $A : \mathbf{V} \to \mathbf{W}$ is called linear if*
$$A(\lambda_1\mathbf{v}_1 + \lambda_2\mathbf{v}_2) = \lambda_1 A(\mathbf{v}_1) + \lambda_2 A(\mathbf{v}_2)$$
for all $\mathbf{v}_1, \mathbf{v}_2 \in \mathbf{V}$ where λ_1, λ_2 are scalars.

Example 2.1.18 *A linear operator mapping $\mathbb{R}^n$ to $\mathbb{R}^m$ is defined by a matrix $\mathbf{A}$ of size $m \times n$, then given $\mathbf{x} \in \mathbb{R}^n, \mathbf{y} = \mathbf{A}\mathbf{x} \in \mathbb{R}^m$.*

Definition 2.1.19 ***Convex Set***. *A set S is convex if, for any $\lambda \in [0,1]$ and $u, v \in S$, $\lambda u + (1-\lambda)v \in S$. That is, S is a convex set if any convex combination of every two elements of S is also in S.*

Definition 2.1.20 ***Convex Function***. *Let S be a convex subset of an n-dimensional vector space $\mathbf{V}$, that is for any $r > 1$ vectors $\mathbf{x}_1, \ldots, \mathbf{x}_r \in S$ and any $\lambda_1, \ldots, \lambda_r \in \mathbb{R}$, $\lambda_k \geq 0, k = 1, \ldots, r$ such that $\lambda_1 + \ldots + \lambda_r = 1$ we have $\lambda_1\mathbf{x}_1 + \ldots + \lambda_r\mathbf{x}_r \in S$. Then a function f defined on S is called convex if for all $\mathbf{x}_i, \mathbf{x}_j \in S$ and $\alpha \in (0,1)$, we have*
$$f(\alpha\mathbf{x}_i + (1-\alpha)\mathbf{x}_j) \leq \alpha f(\mathbf{x}_1) + (1-\alpha)f(\mathbf{x}_j). \tag{2.4}$$
f is called strictly convex if the inequality is strict for $\mathbf{x}_i \neq \mathbf{x}_j$.

Example 2.1.21 *Examples of convex functions are*

- *Affine function, $f(\mathbf{x}) = A\mathbf{x} + b$ is a convex function where $f : \mathbb{R}^2 \to \mathbb{R}$.*
- *The total variation (TV) of a function u defined as*
$$TV(u) = \int_\Omega |\nabla u(\mathbf{x})|\,\mathrm{d}\mathbf{x}$$
where $u : \Omega \subset \mathbb{R}^n \to \mathbb{R}$, is a convex function.

2.2 Calculus of Variations

The calculus of variations is about solving extremal problems for a functional via finding a path such that some integral along the path is extremised. The development of this area started when Johann Bernoulli (1667 – 1748) posed a challenge problem known today as the brachistochrone problem, to his colleagues, including Newton. The problem was to find the shortest path connecting two points in a minimal amount of time. Today, the calculus of variations plays a vital role in many fundamental and modern applications of mathematics, physics, and engineering. In this section, we introduce the tools needed to compute the first variation of a functional using the Gâteaux derivative in order to arrive at the so-called Euler-Lagrange equation which characterises the minimiser of a particular minimisation problem.

Definition 2.2.1 ***Admissible Functions***. *A function $u(\boldsymbol{x})$, that is permissible as input to a functional $\mathcal{J}$ is called an admissible function. The admissible function satisfies function smoothness condition and boundary conditions. The full set of all admissible functions is called the domain of the functional.*

2.2.1 Variation of a Functional

Consider a general functional $\mathcal{J}(u)$ where $\mathcal{J} : \Omega \to \mathbb{R}$ and Ω denotes some normed linear space consisting of admissible functions (for example, $\Omega \subseteq \mathbb{R}^n, n \geq 1$). Let

$$\mathcal{J}(u) = \int_\Omega L(\boldsymbol{x}, u(\boldsymbol{x}), \nabla u(\boldsymbol{x})) \, \mathrm{d}\boldsymbol{x}. \tag{2.5}$$

The functional $\mathcal{J}$ depends upon the independent variable $\boldsymbol{x} = (x_1, x_2, ..., x_n)$, an unknown function $u(\boldsymbol{x})$ of this variable and its gradient $\nabla u(\boldsymbol{x}) = \left(\frac{\partial u(\boldsymbol{x})}{\partial x_1}, \ldots, \frac{\partial u(\boldsymbol{x})}{\partial x_n}\right)$ Here $\mathrm{d}\boldsymbol{x}$ is the n-differential element defined as $\mathrm{d}\boldsymbol{x} = \mathrm{d}x_1 \, \mathrm{d}x_2 ... \, \mathrm{d}x_n$. The calculus of variations deals with the problem of solving the following minimisation problem

$$\min_u \mathcal{J}(u). \tag{2.6}$$

2.2.2 Gâteaux Derivative of a Functional

Definition 2.2.2 ***Gâteux Derivative***. *Let $\mathcal{J}(u)$ be a functional defined on a Banach space B such that $\mathcal{J} : B \to \mathbb{R}$. The Gâteaux derivative of $\mathcal{J}$ is defined as*

$$\delta\mathcal{J}(u(\boldsymbol{x}); v(\boldsymbol{x})) = \lim_{\varepsilon \to 0} \frac{\mathcal{J}(u(\boldsymbol{x}) + \varepsilon v(\boldsymbol{x})) - \mathcal{J}(u(\boldsymbol{x}))}{\varepsilon} = \frac{d}{d\varepsilon} \mathcal{J}(u(\boldsymbol{x}) + \varepsilon v(\boldsymbol{x}))\Big|_{\varepsilon=0}.$$

$\delta\mathcal{J}(u(\boldsymbol{x}); v(\boldsymbol{x}))$ is called the first variation of $\mathcal{J}$ at $u(\boldsymbol{x})$ in the direction of $v(\boldsymbol{x})$ where $v(\boldsymbol{x}) \in C_0^\infty(\Omega)$.

Lemma 2.2.3 ***Local Minimiser***. *A function $u(\boldsymbol{x})^*$ is a local minimiser of the functional $\mathcal{J}$ if there exists a neighbourhood N of $u(\boldsymbol{x})^*$ with $\mathcal{J}(u(\boldsymbol{x})^*) \leq \mathcal{J}(u(\boldsymbol{x}))$ for all $u(\boldsymbol{x}) \in N$.*

Lemma 2.2.4 ***Global Minimiser****. A function $u(\boldsymbol{x})^*$ is a global minimiser of the functional $\mathcal{J}$ if $\mathcal{J}(u(\boldsymbol{x})^*) \leq \mathcal{J}(u(\boldsymbol{x}))$ for all $\boldsymbol{x} \in \mathbb{R}^n$.*

Lemma 2.2.5 ***Stationary Point****. At an extremal point $\boldsymbol{x} \in \mathbb{R}^n$ of a functional $\mathcal{J}$, we have*

$$\delta\mathcal{J}(u(\boldsymbol{x}), v(\boldsymbol{x})) = 0$$

for all $v(\boldsymbol{x})$ in its class of admissible variations. The functional $\mathcal{J}$ is said to be stationary at the extremal point $\boldsymbol{x}$.

Lemma 2.2.6 ***Necessary Condition for a Local Minimiser****. The most important necessary condition to be satisfied by any minimiser of a variational integral $\mathcal{J}(u)$ is the vanishing of its first variation $\delta\mathcal{J}$ defined as*

$$\delta\mathcal{J}(u) = \frac{d}{d\varepsilon}\mathcal{J}(u + \varepsilon v)\Big|_{\varepsilon=0} = 0.$$

2.2.3 Gauss's Theorem

The Gauss or divergence theorem relates the flow of a vector field through a surface to the divergence of the vector inside the surface. Consider a vector field $\boldsymbol{F} = \boldsymbol{F}(\boldsymbol{x})$ which is continuously differentiable on a domain Ω where $\Omega \subset \mathbb{R}^n$ is an open and bounded subset of $\mathbb{R}$ with piece-wise smooth boundary $\partial\Omega$. The theorem states that

$$\int_\Omega (\nabla \cdot \boldsymbol{F}) \ \mathrm{d}\boldsymbol{x} = \int_{\partial\Omega} \boldsymbol{F} \cdot \boldsymbol{n} \ \mathrm{d}s \tag{2.7}$$

where $\nabla \cdot \boldsymbol{F} = \frac{\partial \boldsymbol{F}}{\partial x_1} + \frac{\partial \boldsymbol{F}}{\partial x_2} + ... + \frac{\partial \boldsymbol{F}}{\partial x_n}$, $\mathrm{d}\boldsymbol{x} = \mathrm{d}x_1 \ \mathrm{d}x_2 \ ... \ \mathrm{d}x_n$, $\boldsymbol{n} = (n_1, n_2, ..., n_n)$ is the outward unit normal of $\partial\Omega$ and $\mathrm{d}s$ indicates integration with respect to the surface area on $\partial\Omega$.

2.2.4 Integration by Parts

An immediate consequence of the divergence theorem is the integration by parts formula. Applying (2.7) to the product of a scalar function g and a vector field $\boldsymbol{F}$, we obtain the vectorial representation

$$\int_\Omega (\boldsymbol{F} \cdot \nabla g + g\nabla \cdot \boldsymbol{F}) \ \mathrm{d}\boldsymbol{x} = \int_{\partial\Omega} g\boldsymbol{F} \cdot \boldsymbol{n} \ \mathrm{d}\boldsymbol{s}. \tag{2.8}$$

For the 1-dimensional case, $\boldsymbol{F} = u(x), \ g = v(x)$ and the differentials $\nabla\boldsymbol{F} = u'(x), \nabla g = v'(x)$, the integration by parts formula is in the familiar form

$$\int_\Omega u(x)v'(x) \ \mathrm{d}x = v(x)u(x) - \int_\Omega v(x)u'(x) \ \mathrm{d}x.$$

2.2.5 Fundamental Lemma of the Calculus of Variations

As seen before, the first variation of a functional $\mathcal{J}$ depends on an arbitrary test function $v(\boldsymbol{x})$ and the necessary condition for the minimiser is the vanishing of the first variation of $\mathcal{J}$. Thus to derive the Euler-Lagrange equation, we need the so called fundamental lemma of the calculus of variations (the Du Bois-Reymond lemma) to help us to take the test function $v(\boldsymbol{x})$ out. It is stated as follows.

Lemma 2.2.7 ***The Du Bois-Reymond Lemma.*** *Suppose u is locally integrable function defined on an open set $\Omega \subset \mathbb{R}^n$. If*

$$\int_\Omega u(\boldsymbol{x})v(\boldsymbol{x})\ \mathrm{d}\boldsymbol{x} = 0 \ \textit{ for all } \ v(\boldsymbol{x}) \in C_0^\infty(\Omega) \ \textit{ then } \ u(\boldsymbol{x}) = 0. \tag{2.9}$$

In order to conclude this section, we present an example of how to compute the first variation of a functional of interest to us.

Example 2.2.8 *Consider the problem of finding the first variation of the functional*

$$\mathcal{J}(u) = \int_\Omega |\nabla u|^2\ \mathrm{d}x_1\,\mathrm{d}x_2$$

defined on a domain $\Omega \subset \mathbb{R}^2$. We introduce a small variation εv composed of the parameter ε and a continuously differentiable function v with compact support in Ω. Then we compute,

$$\begin{aligned}
&\frac{d}{d\varepsilon}\mathcal{J}(u+\varepsilon v)\Big|_{\varepsilon=0} = \frac{d}{d\varepsilon}\int_\Omega |\nabla(u+\varepsilon v)|^2\ \mathrm{d}x_1\,\mathrm{d}x_2\Big|_{\varepsilon=0}\\
&= \lim_{\varepsilon\to 0}\frac{1}{\varepsilon}\int_\Omega |\nabla(u+\varepsilon v)|^2 - |\nabla u|^2\ \mathrm{d}x_1\,\mathrm{d}x_2\\
&= \lim_{\varepsilon\to 0}\frac{1}{\varepsilon}\int_\Omega \left(\sqrt{(u_x+\varepsilon v_x)^2+(u_y+\varepsilon v_y)^2}\right)^2 - \left(\sqrt{u_x^2+u_y^2}\right)^2\ \mathrm{d}x_1\,\mathrm{d}x_2\\
&= \lim_{\varepsilon\to 0}\frac{1}{\varepsilon}\int_\Omega u_x^2 + 2\varepsilon u_x v_x + \varepsilon^2 v_x^2 + u_y^2 + 2\varepsilon v_y u_y + \varepsilon^2 v_y^2 - u_x^2 - u_y^2\ \mathrm{d}x_1\,\mathrm{d}x_2\\
&= \lim_{\varepsilon\to 0}\frac{1}{\varepsilon}\int_\Omega 2\varepsilon(u_x v_x + u_u v_y) + \varepsilon^2(v_y^2+v_x^2)\ \mathrm{d}x_1\,\mathrm{d}x_2\\
&= \lim_{\varepsilon\to 0}\int_\Omega 2(u_x v_x + u_y v_y) + O(\varepsilon)\ \mathrm{d}x_1\,\mathrm{d}x_2\\
&= 2\int_\Omega \nabla u\cdot\nabla v\ \mathrm{d}x_1\,\mathrm{d}x_2.
\end{aligned}$$

We need Gauss's theorem (2.8),which gives

$$\int_{\partial\Omega} \phi\boldsymbol{\omega}\cdot\boldsymbol{n}\ \mathrm{d}s = \int_\Omega \phi\nabla\cdot(\boldsymbol{\omega}) + \nabla\phi\cdot\boldsymbol{\omega}\ \mathrm{d}x_1\,\mathrm{d}x_2.$$

Then, using $\phi = v$, and $\boldsymbol{\omega} = \nabla u$, we have

$$\int_\Omega \nabla u\cdot\nabla v\ \mathrm{d}x_1\,\mathrm{d}x_2 = \int_{\partial\Omega} v\nabla u\cdot\boldsymbol{n}\ \mathrm{d}s - \int_\Omega v\nabla\cdot(\nabla u)\ \mathrm{d}x_1\,\mathrm{d}x_2.$$

From the fundamental lemma of calculus of variation (2.9), firstly we have,

$$\int_{\partial\Omega} v\nabla u\cdot\boldsymbol{n}\ \mathrm{d}s = 0, \ \textit{ for all } \ v(\boldsymbol{x}) \in C_0^\infty(\Omega)$$

then, $\nabla u\cdot\boldsymbol{n} = 0$ which represents the boundary condition of the problem. Secondly, we *have*

$$-\int_\Omega v\nabla\cdot(\nabla u)\ \mathrm{d}x_1\,\mathrm{d}x_2 = 0,$$

and using (2.9), we have

$$-\nabla\cdot(\nabla u) = -\Delta u = 0.$$

In summary, we have the following partial differential equation (PDE) known as the Euler-Lagrange equation that must be satisfied

$$-\nabla\cdot(\nabla u) = -\Delta u = 0 \ \ \textit{in}\ \ \Omega,\ \ \nabla u\cdot\boldsymbol{n} = 0 \ \ \textit{on}\ \ \partial\Omega.$$

2.2.6 Functions of Bounded Variation

Let Ω be a bounded open subset of $\mathbb{R}^n$ and let $u \in L^1(\Omega)$. The definition of the total variation of u is

$$TV(u) = \int_\Omega |Du|(\boldsymbol{x})\ \mathrm{d}\boldsymbol{x} = \sup_{\boldsymbol{\varphi}\in V}\Big\{\int_\Omega u(\boldsymbol{x})\ \mathrm{div}\ \boldsymbol{\varphi}\ \mathrm{d}\boldsymbol{x}\Big\}$$

where V is the set of test functions

$$V = \{\boldsymbol{\varphi} = (\varphi_1, \varphi_2, ..., \varphi_n) \in C_0^1(\Omega;\mathbb{R}^n)^n : \|\varphi_i\|_{L^\infty(\Omega)} \leq 1 \ \text{ for } i = 1, ..., n,\}$$

and the divergence is given as

$$\mathrm{div}\ \boldsymbol{\varphi} = \sum_{i=1}^n \frac{\partial\varphi_i}{\partial x_i},$$

$\mathrm{d}\boldsymbol{x}$ is the Lebesgue measure [1] and C_0^1 is the space of continuously differentiable functions with compact support in Ω. Du represents the distributional or weak gradient of u. In [30], letting $u \in C^1(\Omega)$ and using integration by parts we have,

$$\int_\Omega u\ \mathrm{div}\ \boldsymbol{\varphi}\ \mathrm{d}\boldsymbol{x} = -\int_\Omega \sum_{i=1}^n \frac{\partial u}{\partial x_1}\varphi_i\ \mathrm{d}\boldsymbol{x}$$

for every $\boldsymbol{\varphi} \in C_0^1(\Omega;\mathbb{R}^n)^n$, so that

$$\int_\Omega |Du|\,\mathrm{d}\boldsymbol{x} = \int_\Omega |\nabla u|\ \mathrm{d}\boldsymbol{x}$$

where $\nabla u = \left(\frac{\partial u}{\partial x_1}, \frac{\partial u}{\partial x_2}, ..., \frac{\partial u}{\partial x_n}\right)$. Denote by $BV(\Omega)$ the space of all functions in $L^1(\Omega)$ with bounded variation.

[1] In Euclidean spaces, the Lebesgue measure is a standard way to assign a measure (for example: length, area or volume) to any given subset. Hence, sets with finite Lebesgue measure are called Lebesgue measureables. This measure is used to define Lebesgue integration.

Example 2.2.9 *Consider the following three functions:*

$$
\begin{aligned}
&f(x) = (16x-1)(8x-1)(4x-1)(2x-1)(x-1), \;\; x \in \Omega = [0,1] \\
&g(x) = \sin x, \;\; x \in \Omega = [0,1] \\
&h(x) = \begin{cases} 0, & x = 0 \\ x \sin\left(\frac{1}{x}\right), & x \in (0,1]. \end{cases}
\end{aligned}
$$

Here, $f(x)$ and $g(x)$ belong to the space of functions of bounded variation $BV(\Omega)$. The total variations (TV) of $f(x)$ and $g(x)$ are given by

$$
\begin{aligned}
\int_\Omega \left|\nabla f(x)\right| \mathrm{d}x &= \int_0^1 \left|\frac{\mathrm{d}f}{\mathrm{d}x}\right| \mathrm{d}x \\
&= \int_0^1 5120x^4 - 7936x^3 + 3720x^2 - 620x + 31 \,\mathrm{d}x \\
&= 40.0083
\end{aligned}
\tag{2.10}
$$

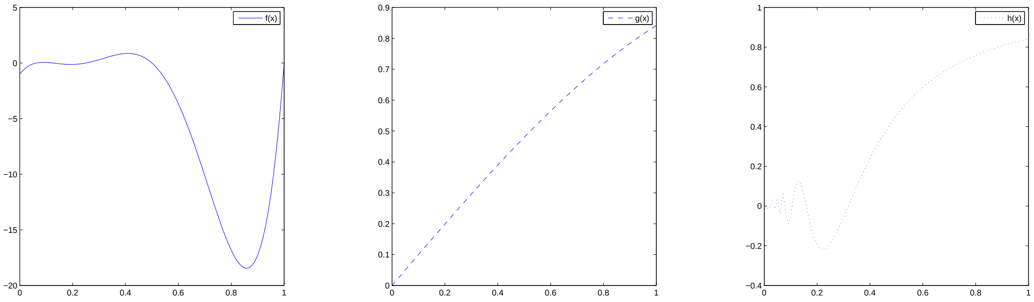
and

$$
\begin{aligned}
\int_\Omega \left|\nabla g(x)\right| \mathrm{d}x &= \int_0^1 \left|\frac{\mathrm{d}g}{\mathrm{d}x}\right| \mathrm{d}x \\
&= \int_0^1 \cos x \,\mathrm{d}x \\
&= 0.8415,
\end{aligned}
\tag{2.11}
$$

respectively. For the function $h(x)$, we have $\Omega = [0,1]$. The function is plotted on Figure 2.1 (c). We see that as $x \to 0$, the frequency of the oscillations of $h(x)$ increases, then the more x approaches zero the more variations need to be added and the value of the integral

$$
\int_\Omega |\nabla h| \;\mathrm{d}x
$$

or total variation of $h(x)$ increases. Therefore, this function has infinite total variation and does not belong to $BV(\Omega)$.



(a) $f(x)$

(b) $g(x)$

(c) $h(x)$

Figure 2.1: From left to right: the graphs of the functions $f(x)$, $g(x)$ and $h(x)$ where $f(x)$ and $g(x)$ are of bounded variation for $\Omega = [0,1]$. The function $h(x)$ has infinite total variation and is the space nor a bounded variation function.

In order to conclude this section, we state the co-area formula which is a powerful tool for the analysis of BV functions.

Definition 2.2.10 ***Lipschitz Continuous Functions***. *A function J is called Lipschitz continuous with Lipschitz constant L_f on $\mathbb{R}^n$ if there is a nonnegative constant L_f such that*

$$\|J(\boldsymbol{y}) - J(\boldsymbol{x})\| \leq L_f \|\boldsymbol{y} - \boldsymbol{x}\| \textit{for all } \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$$

for any given operator norm.

Definition 2.2.11 ***Borel Set***. *Given X, any topological space, we say that $E \subset X$ is a Borel set if E can be obtained by a countable number of operations, starting from open sets, each operation consisting of taking unions, intersections and complements [4, 79].*

Definition 2.2.12 ***Perimeter***. *Let E be a Borel set and Ω an open set in $\mathbb{R}^n$. Define the perimeter of E in Ω as*

$$Per(E, \Omega) = \int_\Omega |D_{\chi_E}| = \sup \left\{ \int_E div\, \boldsymbol{\varphi}\, \mathrm{d}\boldsymbol{x} : \boldsymbol{\varphi} \in C_0^1(\Omega, \mathbb{R}^n) \quad and \quad |\boldsymbol{\varphi}(\boldsymbol{x})| \leq 1 \right\} \tag{2.12}$$

where

$$\chi_E = \begin{cases} 1, & \textit{if } \boldsymbol{x} \in E; \\ 0, & \textit{if } \boldsymbol{x} \in \Omega - E \end{cases} \tag{2.13}$$

is the indicator function of E.

Definition 2.2.13 ***Level Set***. *A level set of a real function value u of n real variables is a set of the form*

$$L_c u = \Big\{ (x_1, \ldots, x_n) | u(x_1, \ldots, x_n) = c \Big\}$$

that is the set where the function takes a given constant c.

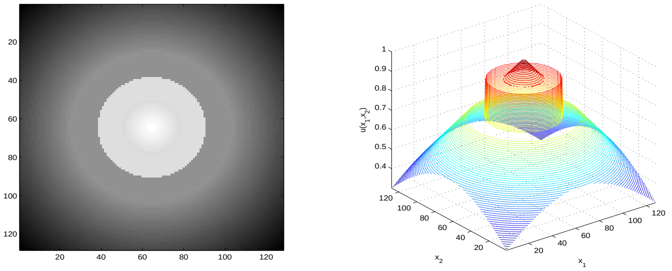
Definition 2.2.14 ***Level Curves***. *Let $u(x_1, x_2)$ be a function in two dimension. The set of pairs $\boldsymbol{x} = (x_1, x_2)$ such that $u(\boldsymbol{x}) = c$ is called the level curves of u for the value c.*

Definition 2.2.15 ***Co-area Formula***. *Let $u = u(\boldsymbol{x})$ and $f = f(\boldsymbol{x})$ be two scalar functions defined on $\mathbb{R}^n$. Assume that u is Lipschitz continuous and that for almost every $\lambda \in \mathbb{R}$, the level set $L_\lambda = \{\boldsymbol{x} \in \mathbb{R}^n : u(x) = \lambda\}$ is a smooth $(n-1)$-dimensional hyper-surface in $\mathbb{R}^n$. Suppose also that f is continuous and integrable. Then*

$$\int_{\mathbb{R}^n} |\nabla u| f \, \mathrm{d}\boldsymbol{x} = \int_{-\infty}^{+\infty} \left(\int_{L_\lambda} f \, \mathrm{d}s \right) \mathrm{d}\lambda. \tag{2.14}$$

For the particular case when $f = 1$ and the region of integration is a subset $\Omega \subset \mathbb{R}^n$, we have

$$\int_\Omega |\nabla u| \, \mathrm{d}\boldsymbol{x} = \int_{-\infty}^{+\infty} \left(\int_{L_\lambda} f \, \mathrm{d}s \right) \mathrm{d}\lambda = \int_{-\infty}^{+\infty} Per(L_\lambda, \Omega) \, \mathrm{d}\lambda. \tag{2.15}$$



(a) $u(\boldsymbol{x})$ (b) λ-level curves

Figure 2.2: On the left is the given grey level image $u(\boldsymbol{x})$ and on the right some of its λ-level curves, these are curves where $u(\boldsymbol{x}) = \lambda$ for some $\lambda = [0, 1]$.

Example 2.2.16 *Given the following function*

$$u(x_1, x_2) = \begin{cases} 1 - \sqrt{(x_1 - 64)^2 + (y - 64)^2)}/128, & (x_1, x_2) \in \Omega \backslash \Omega_1 \cup \Omega_2; \\ 0.9, & (x_1, x_2) \in \Omega_1; \\ 0.7, & (x_1, x_2) \in \Omega_2 \end{cases} \tag{2.16}$$

with $\Omega = [0, 128]^2$, Ω_1 is the ring bounded by two circles $(x_1 - 64)^2 + (x_2 - 64)^2 = 14^2$ and $(x_1 - 64)^2 + (x_2 - 64)^2 = 26^2$ and Ω_2 the ring bounded by the latter circle and $(x_1 - 64)^2 + (x_2 - 64)^2 = 38^2$. Let us select some slice (level set) of u by setting $L_\lambda = \{\boldsymbol{x} \in \mathbb{R}^n : u(x) = \lambda\}$ for $\lambda = [0, 1]$. The 3D plot in Figure 2.2 (b) shows some of its level sets. Therefore according to (2.12), the perimeter of each slice of u is given by

$$Per(L_\lambda, \Omega) = \int_\Omega |D_{\chi_{L_\lambda}}|. \tag{2.17}$$

Using the co-area formula, we obtain

$$\int_\Omega |\nabla u| \, \mathrm{d}\boldsymbol{x} = \int_0^1 Per(L_\lambda, \Omega) \, \mathrm{d}\lambda. \tag{2.18}$$

This result shows that the total variation of a given function u is just the sum of every length of all its λ-level curves. This automatically takes care of all the discontinuities of u and therefore the contribution of edges to the total variation integral is enforced.

2.3 Ill-posed Problems and Regularisation

Mathematical problems represent our attempts to model observations made from particular physical phenomenon. As such, the heat equation comes from modelling an observation made of the temperature of a certain object at a time interval. A problem is said to be well-posed or correctly set if it satisfies three conditions (existence, uniqueness and stability). Problems which violate any of the three conditions are said to be ill-posed. In this section we look at ill-posed problems and regularisation which is a common way of overcoming the ill-posedness of problem, thus allowing for the problem to be solved.

2.3.1 Inverse Problems

Given the problem

$$Au = f \tag{2.19}$$

where $A \in \mathcal{L}(H, F), u \in H, f \in F$ and H, F are Hilbert spaces, we define the idea of well and ill posed problems below.

Definition 2.3.1 ***Forward and Inverse Problems****. A forward problem is the process of calculating the data y from the parameter x using a measurement operator f. The operator f maps the parameter in a function space X typically a Banach or Hilbert space to the space of data Y. We write*

$$y = f(x), \ \ \textit{for} \ x \in X \ \textit{ and } \ y \in Y \tag{2.20}$$

as the correspondence between the parameter x and the data y. An inverse problem is the process to find the parameter $x \in X$ from the knowledge of the data $y \in Y$ such that (2.20) or an approximation of (2.20) holds.

Definition 2.3.2 *Problem (2.19) is well posed in the sense of Hadamard if*

1. *Existence: for all $f \in F$, (2.19) has solution $u^* \in H$.*
2. *Uniqueness: for all $f \in F$, the solution to (2.19) is unique.*
3. *Stability: the solution, $u^* \in H$ depends continuously on the data.*

If any of the three conditions above are not satisfied, then the problem (2.19) is called an ill-posed problem.

Example 2.3.3 *Given*

$$A = \begin{bmatrix} 2 & 3 \\ 4 & 6 \end{bmatrix}, \ y = \begin{bmatrix} 2 \\ 4 \end{bmatrix}$$

and we need to solve the linear system

$$Ax = y \tag{2.21}$$

where $x = [x_1, x_2]^T$. However, the system is an underdetermined system because it can be reduced to only one equation which is

$$2x_1 + 3x_2 = 2.$$

Thus, the system has an infinite number of solutions such as $(x_1, x_2) = (0, \frac{2}{3})$ and $(x_1, x_2) = (1, 0)$. We say that the problem in (2.21) is an ill-posed problem because the solution of the system is not unique.

Many problems in real life applications are inverse problems which exhibit ill-posedness. For example, given two data sets X and Y, we may be asked to calculate the value of $Z = X + Y$ which is an example of a forward problem. The problem becomes an inverse problem if we are given Z and we are asked to calculate the values of X and Y. Ill-posed problems usually result from a lack of precise mathematical formulation and typically violate the stability condition since small changes in the given data lead to large changes in the result. To illustrate this, consider the following example.

Example 2.3.4 *Given*

$$A = \begin{bmatrix} 2 & 3 \\ 2 & 3+\epsilon \end{bmatrix},\ y = \begin{bmatrix} 2 \\ 0 \end{bmatrix}.$$

The forward problem is to compute $T = T(y) = Ay$, for $\epsilon = 0$, which has solution $T = [4, 4]^T$. Meanwhile, for an associated inverse problem, we need to compute y given this T. However for $\epsilon = 0$, A is not invertible, so there is no solution to the problem. If we change $\epsilon = 10^{-6}$, then A becomes invertible and has a unique solution for $y = [2, 0]^T$. Perturbing T slightly, to $\widehat{T} = [4, 4-\epsilon]^T$, the solution to the inverse problem is $\widehat{y} = [3.5, -1]^T$ which is considerably different from y since $|y|_2 = 4$ and $|\widehat{y}|_2 = 13.25$. Observe that one of components of $\widehat{y}$ is negative which results in a huge problem.

2.3.2 Tikhonov Regularisation

Andrey N. Tikhonov introduced the concept of regularisation to solve ill-posed problems. It can be understood as introducing a constraint to the original problem which results in the stability of the solution. The constraint is added to the problem based on prior information about the behaviour of the solution [97].

Example 2.3.5 *Given Hilbert spaces U, Y, elements $u \in U, y \in Y$ and an operator $A : U \to Y$ such that $Au = y$, the problem of finding the solution $u^* = A^{-1}y$ is an ill posed problem if either u^* is not unique, does not exist or is unstable. A standard practice is to look for the least squares solution of the following minimisation problem*

$$\min \|Au - y\|_2^2, \tag{2.22}$$

where the error norm measures how far the solution u is from the true solution u^*. Tikhonov regularisation replaces the minimisation problem (2.22) by the solution of a penalised least squared problem

$$\min \|Au - y\|_2^2 + \gamma \|Lu\|_2^2 \tag{2.23}$$

where L is a regularisation operator and $\gamma > 0$ is the regularisation parameter determines how much weight is given to the regularisation term of the joint functional. We denote the above generalised Tikhonov regularisation model as follows:

$$\min_{u \in U} \mathcal{J}_\gamma(u) = \mathcal{D}(A, u, y) + \gamma \mathcal{S}(u)$$

where $\mathcal{D}$ and $\mathcal{S}$ are the fitting and regularisation functional terms respectively.

2.4 Discretisation of PDEs and Notation

A continuous model and its equation are transferred to a discrete problem through discretisation for subsequent numerical implementation because most equations cannot be solved analytically. Consider a bounded and open domain Ω in $\mathbb{R}^d$ with boundary

$\partial\Omega$. A continuous linear boundary value problem in d dimensions is denoted as

$$\begin{aligned} L^{\Omega}u(x_1,x_2,...,x_d) &= f^{\Omega}(x_1,x_2,...,x_d) \text{ for } (x_1,x_2,...,x_d) \in \Omega \\ L^{\partial\Omega}u(x_1,x_2,...,x_d) &= f^{\partial\Omega}(x_1,x_2,...,x_d) \text{ for } (x_1,x_2,...,x_d) \in \partial\Omega \end{aligned} \tag{2.24}$$

where L represent the linear operator of the problem and $u(x_1, x_2, .., x_d)$ is the required function. For nonlinear boundary value problems we have

$$\begin{aligned} \mathcal{N}^{\Omega}(u(x_1,x_2,...,x_d)) &= f^{\Omega}(x_1,x_2,...,x_d) \text{ for } (x_1,x_2,...,x_d) \in \Omega \\ \mathcal{N}^{\partial\Omega}(u(x_1,x_2,...,x_d)) &= f^{\partial\Omega}(x_1,x_2,...,x_d) \text{ for } (x_1,x_2,...,x_d) \in \partial\Omega. \end{aligned} \tag{2.25}$$

where $\mathcal{N}$ is the nonlinear operator of the problem.

Definition 2.4.1 ***Laplace Operator***. *The Laplace operator or Laplacian is a differential operator which is given by the divergence of the gradient of a function on Euclidean space and it is usually denoted by* $div \cdot \nabla, \nabla \cdot \nabla, \nabla^2$, *or* Δ. *The Laplacian of a scalar function* $f(x, y, z)$ *is defined by*

$$\Delta f(x,y,z) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} + \frac{\partial^2 f}{\partial z^2}.$$

Example 2.4.2 *Consider the following 2D Poisson equation:*

$$\Delta u(x_1,x_2) = f(x_1,x_2) \tag{2.26}$$

with Neumann boundary condition

$$\nabla u \cdot \boldsymbol{n} = 0 \quad on \quad \partial\Omega. \tag{2.27}$$

In our work, the domain $\Omega \in \mathbb{R}^n$ is usually rectangular and the values of f known at uniformly distributed points in the domain. Therefore the most natural discretisation method to use is the finite difference method. Assuming that $\Omega = (a, b) \times (c, d)$ is rectangular we impose a cartesian grid with grid spacing

$$h_1 = \frac{b-a}{n_1} \tag{2.28}$$

in the x_1 direction and

$$h_2 = \frac{d-c}{n_2} \tag{2.29}$$

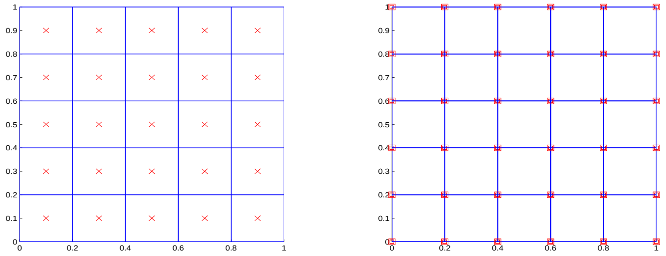
in the x_2 direction. In a vertex-centred discretisation, grid points are placed at the vertices of the mesh so that, there are $(n_1 + 1) \times (n_2 + 1)$ grid points including points on the boundary. The grid point (i, j) are located at

$$(x_{1,i}, x_{2,j}) = (ih_1, jh_2) \text{ for } 0 \le i \le n_1 \text{ and } 0 \le j \le n_2. \tag{2.30}$$

In a cell-centred discretisation, grid points are located at the centre of the cells so that there are $n_1 \times n_2$ grid points and the point (i, j) is located at

$$(x_{1,i}, x_{2,j}) = \Big(a + \frac{2i-1}{2}h_1, c + \frac{2j-1}{2}h_2\Big) \text{ for } 1 \le i \le n_1 \text{ and } 1 \le j \le n_2. \tag{2.31}$$

The interior of the discrete grid is denoted by Ω^h and the boundary by $\partial\Omega^h$. Figure 2.3 shows examples of vertex and cell-centred discretisation of a square domain. The



(a) Cell-centred.

(b) Vertex-centred.

Figure 2.3: Illustration of (a) cell-centred discretisation and (b) vertex-centred discretisation on a square mesh. Red crosses show the cell-centred points and the red boxes show the vertex grid points.

PDE is approximated locally using the Taylor series expansion

$$u(x_1 + h_1, x_2) = u(x_1, x_2) + h_1 \frac{\partial u(x_1, x_2)}{\partial x_1} + \frac{h_1}{2} \frac{\partial^2 u(x_1, x_2)}{\partial x_1^2} + O(h_1^3)$$

and

$$u(x_1 - h_1, x_2) = u(x_1, x_2) - h_1 \frac{\partial u(x_1, x_2)}{\partial x_1} + \frac{h_1}{2} \frac{\partial^2 u(x_1, x_2)}{\partial x_1^2} + O(h_1^3)$$

where $O(h_1^3)$ denotes terms containing third and higher powers of h_1. The operator $\frac{\partial u(x_1,x_2)}{\partial x_1}$ at the grid point i, j can be approximated as follows

1. First order forward :

$$\delta_{x_1}^{+} u_{i,j} = \frac{u_{i+1,j} - u_{i,j}}{h_1} \approx \left(\frac{\partial u}{\partial x_1} \right)_{i,j} + O(h_1),$$

2. First order backward :

$$\delta_{x_1}^{-} u_{i,j} = \frac{u_{i,j} - u_{i-1,j}}{h_1} \approx \left(\frac{\partial u}{\partial x_1} \right)_{i,j} + O(h_1),$$

3. Second order central :

$$\delta_{x_1}^{c} u_{i,j} = \frac{u_{i+1,j} - u_{i-1,j}}{2h_1} \approx \left(\frac{\partial u}{\partial x_1} \right)_{i,j} + O(h_1^2),$$

where $u_{i,j} = u(x_{1,i}, x_{2,j})$. Approximations of higher order derivatives can be constructed in the similar way. For example, the centred second order difference

$$\delta_{x_1 x_1}^{2} u_{i,j} = \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h_1^2} \approx \left(\frac{\partial^2 u}{\partial x_1^2} \right)_{i,j} + O(h_1^2). \tag{2.32}$$

For the discrete version of equations (2.24) and (2.25), we denote

$$L_h u_h = f_h, \ \text{ and } \ N_h(u_h) = f_h \ \text{ respectively.}$$

2.4.1 Stencil Notation

We shall use the difference formulations, especially the centred second order difference given in (2.32) to approximate the Laplace operator in model problem (2.26) at the interior points of the domain Ω^h,

$$\begin{aligned}(\Delta_h u)_{i,j} &= \delta^2_{x_1 x_1} u_{i,j} + \delta^2_{x_2 x_2} u_{i,j} \\ &= \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h_1^2} + \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{h_2^2}.\end{aligned} \tag{2.33}$$

It is called a five point stencil since only five points are involved. We finally obtain

$$(\Delta_h u)_{i,j} = \frac{u_{i+1,j} - 4u_{i,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1}}{h_1 h_2} \tag{2.34}$$

which can be denoted by the following stencil

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}. \tag{2.35}$$

For the right hand side of model problem (2.26), we simply take the value of f at the points, i.e. $f_{i,j} = f(x_{1,i}, x_{2,j})$. The grid function u_h is stacked along rows of the grid starting at the bottom left point and ending at the top right to produce a vector $\boldsymbol{u}_h$. This type of ordering is called lexicographical ordering. The right hand side vector is stacked in a similar manner into a vector $\boldsymbol{f}_h$. Thus, the discrete linear equation can be written as $A_h \boldsymbol{u}_h = \boldsymbol{f}_h$. For the nonlinear equation, the discrete version using matrix notation is $A(\boldsymbol{u}_h) = \boldsymbol{f}_h$. For more details on model problem (2.26), refer to [50, 87].

2.4.2 Boundary Conditions

Definition 2.4.3 ***Ghost Points***. *In finite difference methods, the points outside the discretisation domain Ω are called as the ghost points.*

There are two types of boundary condition that usually occur in PDEs, namely Dirichlet and Neumann boundary conditions. Dirichlet boundary conditions specify the value of the function which needs to be satisfied at the boundary. However, Neumann boundary conditions specify the normal derivative of the function on a surface. For the model problem (2.26), we have Neumann boundary condition $\nabla u \cdot \boldsymbol{n} = 0$ where involves ghost points outside of the domain after discretisation of the problem,

$$\frac{u_{n_1+1,j} - u_{n_1-1,j}}{2h_1} = 0.$$

as shown in Figure 2.4 using the white circles.

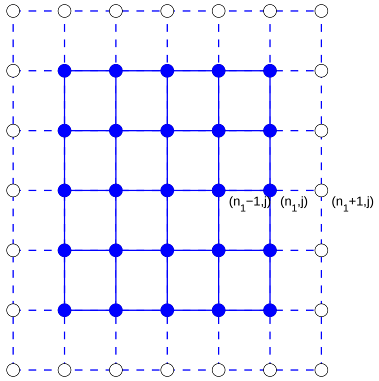


Figure 2.4: Illustration of the ghost points outside the domain using vertex-centred discretisation. The grid points are represented by the blue circles and the white circles are the ghost points.

Using the stencil notation, we have

$$(\Delta_h(u_h))_{i,j} = \frac{1}{h_1 h_2}\begin{bmatrix} 0 & 1 & 0 \\ 1 & -3 & 0 \\ 0 & 1 & 0 \end{bmatrix} u_{i,j} = f_{,j}$$

for the right boundary on the square domain Ω^h.

2.5 Iterative Methods

Now we briefly review iterative methods for a linear system $A\boldsymbol{x} = \boldsymbol{b}$. Iterative methods are used to compute a sequence of progressively accurate iterates to approximate the solution of $A\boldsymbol{x} = \boldsymbol{b}$ where $\boldsymbol{x}$ and $\boldsymbol{b}$ are of dimension n and A is an $n \times n$ matrix. The process starts with an initial approximation $\boldsymbol{x}^{(0)}$ and generates a sequence $\{\boldsymbol{x}^{(k)}\}_{k=1}^{\infty}$ using the relation

$$\boldsymbol{x}^{(k)} = T\boldsymbol{x}^{(k-1)} + \boldsymbol{c}$$

where the matrix T and the vector $\boldsymbol{c}$ are derived from the matrix A as follows

$$\begin{aligned}
&(M - N)\boldsymbol{x} = A\boldsymbol{x} = \boldsymbol{b} \\
&\Leftrightarrow M\boldsymbol{x} = N\boldsymbol{x} + \boldsymbol{b} \\
&\Leftrightarrow \boldsymbol{x} = M^{-1}N\boldsymbol{x} + M^{-1}\boldsymbol{b} \\
&\Leftrightarrow \boldsymbol{x} = T\boldsymbol{x} + \boldsymbol{c}
\end{aligned}$$

where M is a nonsingular matrix and $M = \text{diag}(A)$. Iterative methods are needed to solve large systems where the direct method is too expensive to solve.

2.5.1 Jacobi Method

The Jacobi method solves the ith equation of $A\boldsymbol{x} = \boldsymbol{b}$ for x_i using

$$x_i = \sum_{j=1,j\neq i}^{n} \left(\frac{-a_{ij}x_j}{a_{ii}}\right) + \frac{b_i}{a_{ii}}, \text{for } i = 1, \dots, n.$$

For $\boldsymbol{x}^{(k-1)}, k \geq 1$, we have

$$x_i^{(k)} = \sum_{j=1,j\neq i}^{N} \left(\frac{-a_{ij}x_j^{(k-1)}}{a_{ii}}\right) + \frac{b_i}{a_{ii}}$$

where $a_{ii} \neq 0$ for $i = 1, \dots, N$. If one or more $a_{ii} = 0$ and the system is nonsingular then we can reorder so that no a_{ii} is equal to zero. For the Jacobi method, we have

$$\begin{aligned}
&(D - L - U)\boldsymbol{x} = \boldsymbol{b} \\
&\Leftrightarrow D\boldsymbol{x} = (L + U)\boldsymbol{x} + \boldsymbol{b} \\
&\Leftrightarrow \boldsymbol{x} = D^{-1}(L + U)\boldsymbol{x} + D^{-1}\boldsymbol{b} \\
&\Leftrightarrow \boldsymbol{x} = T\boldsymbol{x} + \boldsymbol{c}
\end{aligned}$$

where D is a diagonal matrix, $-L$ is the strictly lower triangular matrix and $-U$ is the strictly upper triangular matrix of the matrix A. The matrix form of the Jacobi method is given by

$$\boldsymbol{x}^{(k)} = T_J\boldsymbol{x}^{(k-1)} + \boldsymbol{c}_J$$

where $T_J = D^{-1}(L + U)$ and $\boldsymbol{c}_J = D^{-1}\boldsymbol{b}$. The algorithm for solving $A\boldsymbol{x} = \boldsymbol{b}$ using the Jacobi method is given by Algorithm 1. In the weighted Jacobi method, the

Algorithm 1 Jacobi Method
$(\boldsymbol{x}) \leftarrow \text{Jacobi}(A, \boldsymbol{b}, \boldsymbol{x}^{(0)}, IMAX, TOL)$

```
1. Let $k = 1$, $N =$ length $\boldsymbol{b}$.
2. For $k = 1, ..., IMAX$,
    (a) For $i = 1, ..., N$,
        i. Set
            $$x_i^{(k)} = \sum_{j=1,j\neq i}^{N} \left(\frac{-a_{ij}x_j^{(k-1)}}{a_{ii}}\right) + \frac{b_i}{a_{ii}}. \qquad (2.36)$$
    (b) End for.
    (c) If $\|\boldsymbol{b} - A\boldsymbol{x}^{(k)}\|_2 < TOL$ or $\|\boldsymbol{x}^{(k)} - \boldsymbol{x}^{(k-1)}\|_2 < TOL$ , exit else continue.
3. End for.
```

intermediate values $\widehat{\boldsymbol{x}}$ are computed using

$$\widehat{x}_i = \sum_{j=1, j\neq i}^{N} \left(\frac{-a_{ij} x_j^{(k-1)}}{a_{ii}} \right) + \frac{b_i}{a_{ii}} \tag{2.37}$$

and the new approximation for $\boldsymbol{x}^{(k)}$ is given by

$$\boldsymbol{x}_i^{(k)} = (1-\omega) x_i^{(k-1)} + \omega \widehat{x}_i$$

where ω is a weighting factor. In matrix form, the weighted Jacobi method is

$$\begin{aligned} \boldsymbol{x}^{(k)} &= \left((1-\omega)I + \omega T_J\right) \boldsymbol{x}^{(k-1)} + \omega D^{-1}\boldsymbol{b} \\ \boldsymbol{x}^{(k)} &= T_\omega \boldsymbol{x}^{(k-1)} + \boldsymbol{c}_\omega. \end{aligned}$$

2.5.2 Gauss Seidel Method

A better approximation to the Jacobi method is the Gauss Seidel method where the update value for $x_i^{(k)}$ is calculated using the recent values of $x_1^{(k)}, ..., x_{i-1}^{(k)}$ as follows:

$$x_i^{(k)} = \frac{-\sum_{j=1}^{i-1} a_{ij} x_j^{(k)} - \sum_{j=i+1}^{N} a_{ij} x_j^{(k-1)} + b_i}{a_{ii}} \quad \text{for } i = 1, ..., N. \tag{2.38}$$

Rewriting the above equation as

$$a_{ii} x_i^{(k)} + \sum_{j=1}^{i-1} a_{ij} x_j^{(k)} = -\sum_{j=i+1}^{N} a_{ij} x_J^{(k-1)} + b_i,$$

we can see that the matrix form of the Gauss Seidel method is given by $M = D - L$,

$$\begin{aligned} (D-L)\boldsymbol{x}^{(k)} &= U\boldsymbol{x}^{(k-1)} + \boldsymbol{b} \\ \boldsymbol{x}^{(k)} &= (D-L)^{-1}U\boldsymbol{x}^{(k-1)} + (D-L)^{-1}\boldsymbol{b} \\ \boldsymbol{x}^{(k)} &= T_{GS}\boldsymbol{x}^{(k-1)} + \boldsymbol{c}_{GS}. \end{aligned}$$

The algorithm for the Gauss Seidel method is the same as the algorithm for the Jacobi method except that we replace equation (2.36) in Algorithm 1 with equation (2.38).

2.5.3 SOR Method

In the Successive Over Relaxation (SOR) method, the intermediate values $\widehat{x}_i$ are computed using the Gauss Seidel method and

$$\boldsymbol{x}^{(k)} = (1-\omega) x_i^{(k-1)} + \omega \widehat{\boldsymbol{x}}$$

where ω is positive constant. If $\omega < 1$, this is called under relaxation and is used to obtain convergence when the Gauss Seidel method does not converge. If $\omega > 1$ it is called over relaxation and it is used to accelerate convergence of the system when the

Gauss Seidel method does converge. The SOR method is based on the matrix splitting

$$\omega A = (D - \omega L) - (\omega U + (1-\omega)D)$$

and can be defined by the recurrence relation

$$\begin{aligned}
&\boldsymbol{x}^{(k)} = \Big((D - \omega L) - (\omega U + (1-\omega)D)\Big)\boldsymbol{x}^{(k-1)} + \omega(D-\omega L)^{-1}\boldsymbol{b} \\
&\boldsymbol{x}^{(k)} = T_{SOR}\boldsymbol{x}^{(k-1)} + \boldsymbol{c}_{SOR}.
\end{aligned}$$

2.5.4 Block Methods

In these methods the vector $\boldsymbol{x}$ and $\boldsymbol{b}$ are partitioned into several disjoint sub vectors

$$\boldsymbol{x} = (\boldsymbol{x}_1^T, \boldsymbol{x}_2^T, \ldots, \boldsymbol{x}_s^T),\ \boldsymbol{b} = (\boldsymbol{b}_1^T, \boldsymbol{b}_2^T, \ldots, \boldsymbol{b}_s^T).$$

Then the system $A\boldsymbol{x} = \boldsymbol{b}$ can be written in the block form

$$\begin{bmatrix} A_{11} & A_{12} & \cdot & A_{1s} \\ A_{21} & A_{22} & \cdot & A_{2s} \\ \cdot & \cdot & \cdot & \cdot \\ A_{s1} & A_{s2} & \cdot & A_{ss} \end{bmatrix} \begin{pmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \cdot \\ \boldsymbol{x}_s \end{pmatrix} = \begin{pmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \\ \cdot \\ \boldsymbol{b}_s \end{pmatrix}$$

where the block A_{pq} is of size $n_p \times n_q$ where n_p and n_q is the size of $\boldsymbol{x}_p$ and $\boldsymbol{b}_q$ respectively. Assuming that the diagonal blocks are nonsingular, the Jacobi and Gauss Seidel methods can be easily extended to the block level. In the Block Jacobi method for $i = 1, ..., s$, $\boldsymbol{x}_i$ is updated as follows:

$$\boldsymbol{x}_i^{(k)} = A_{ii}^{-1}\left(\sum_{j=1, j\neq i}^{s} -A_{ij}\boldsymbol{x}_j^{(k-1)} + \boldsymbol{b}_i\right).$$

For the Block Gauss Seidel method

$$\boldsymbol{x}_i^{(k)} = A_{ii}^{-1}\left(\sum_{j=1, j\neq i}^{i-1} -A_{ij}\boldsymbol{x}_j^{(k)} + \sum_{j=i+1}^{s} -A_{ij}\boldsymbol{x}_j^{(k-1)} + \boldsymbol{b}_i\right).$$

We have to invert the matrix A_{ii} in order to update $\boldsymbol{x}_i$ and if $\boldsymbol{x}_i$ is large then the step is more expensive. Using matrix notation, we can write

$$\boldsymbol{x}^{(k)} = D_B^{-1}(U_B + L_B)\boldsymbol{x}^{(k-1)} + D_B^{-1}\boldsymbol{b}$$

for the Block Jacobi method and

$$\boldsymbol{x}^{(k)} = (D_B - L_B)^{-1}(U_B)\boldsymbol{x}^{(k-1)} + (D_B - L_B)^{-1}\boldsymbol{b}$$

for the Block Gauss Seidel method.

2.5.5 Convergence

Definition 2.5.1 ***Spectral Radius***. *The spectral radius of a matrix M is defined as*

$$\rho(M) := \max |\lambda|$$

where λ is the eigenvalues of M.

In this section will show that a sequence $\{\boldsymbol{x}^{(k)}\}_{k=0}^{\infty}$ converges to the true solution $\boldsymbol{x}$ of the original system $A\boldsymbol{x} = \boldsymbol{b}$ where $\boldsymbol{x}^{(k)} = T\boldsymbol{x}^{(k-1)} + \boldsymbol{c}$ if and only if the spectral radius of T is less than 1 ($\rho(T) < 1$).

Definition 2.5.2 ***Convergent Matrix***. *A square matrix A is said to be convergent if $\lim_{k\to\infty} A^k = 0$.*

Theorem 2.5.3 ***Convergence of a Matrix***. *A matrix A is convergent if and only if $\rho(A) < 1$.*

Proof The proof can be found in [84].

Lemma 2.5.4 *If the spectral radius $\rho(T) < 1$ then $(I-T)^{-1}$ exists and $(I-T)^{-1} = \sum_{j=0}^{\infty} T^j$.*

Proof If λ is an eigenvalue of T then $(1-\lambda)$ is an eigenvalue of $(I-T)$. Since $\rho(T) < 1$, 1 is not an eigenvalue of T. Hence 0 is not an eigenvalue for $(I-T)$ and $(I-T)$ is not singular. Let

$$S_m = I + T + T^2 + \ldots + T^m$$

then

$$(I-T)S_m = (I + T + T^2 + \ldots + T^m) - (T + T^2 + \ldots + T^{m+1}) = I - T^{m+1}.$$

Using Theorem 2.5.3, $\rho(T) < 1$ implies that T is convergent and

$$\lim_{m\to\infty}(I-T)S_m = \lim_{m\to\infty}(I - T^{m+1}) = I.$$

Thus

$$(I-T)^{-1} = \lim_{m\to\infty} S_m = \sum_{j=0}^{\infty} T^j$$

and the proof is complete.

Theorem 2.5.5 ***Convergence of a Sequence***. *For any $\boldsymbol{x}^{(0)} \in \mathbb{R}^n$, the sequence $\{\boldsymbol{x}^{(k)}\}_{k=0}^{\infty}$ defined by $\boldsymbol{x}^{(k)} = T\boldsymbol{x}^{(k-1)} + \boldsymbol{c}$ (for each $k > 1$) converges to the unique solution of $\boldsymbol{x} = T\boldsymbol{x} + \boldsymbol{c}$ if and only if $\rho(T) < 1$.*

Proof Assume that $\rho(T) < 1$, we have

$$\begin{aligned}
\boldsymbol{x}^{(k)} &= T\boldsymbol{x}^{(k-1)} + \boldsymbol{c} \\
\boldsymbol{x}^{(k)} &= T(T\boldsymbol{x}^{(k-2)} + \boldsymbol{c}) + \boldsymbol{c} \\
\boldsymbol{x}^{(k)} &= T^2\boldsymbol{x}^{(k-2)} + (T + I)\boldsymbol{c} \qquad (2.39) \\
&\vdots \\
\boldsymbol{x}^{(k)} &= T\boldsymbol{x}^{(0)} + (T^{k-1} + \ldots + T^2 + T + I)\boldsymbol{c}.
\end{aligned}$$

Using (2.39), Theorem (2.5.3) and $\rho(T) < 1$, we have

$$\lim_{k\to\infty} \boldsymbol{x}^{(k)} = \lim_{k\to\infty} \left(\sum_{j=0}^{k-1} T^j \boldsymbol{c} \right)$$

which by Lemma (2.5.4) is equal to $(I - T)^{-1}\boldsymbol{c}$. The sequence $\{\boldsymbol{x}^{(k)}\}_{k=0}^{\infty}$ therefore converges to the unique solution of $\boldsymbol{x} = (I - T)^{(-1)}\boldsymbol{c}$ or $\boldsymbol{x} = T\boldsymbol{x} + \boldsymbol{c}$. Conversely, assume that $\boldsymbol{x}^*$ is the unique solution of $\boldsymbol{x} = T\boldsymbol{x} + \boldsymbol{c}$. If $\boldsymbol{c} = 0$, then $\boldsymbol{x}^*$ is the unique solution of $\boldsymbol{x} = T\boldsymbol{x}$, now let $\boldsymbol{y} \in \mathbb{R}^n$ be an arbitrary vector and take initial guess $\boldsymbol{x}^{(0)} = \boldsymbol{x}^* - \boldsymbol{y}$, we have

$$\begin{aligned}
\lim_{k\to\infty} T^k\boldsymbol{y} &= \lim_{k\to\infty} T^k(\boldsymbol{x}^* - \boldsymbol{x}^{(0)}) \\
&= \lim_{k\to\infty} T^{k-1}(\boldsymbol{x}^* - \boldsymbol{x}^{(1)}) \\
&= \lim_{k\to\infty} T^{k-2}(\boldsymbol{x}^* - T\boldsymbol{x}^{(2)}) \\
&\vdots \\
&= \lim_{k\to\infty} (\boldsymbol{x}^* - \boldsymbol{x}^{(k)}) = 0.
\end{aligned}$$

Since $\boldsymbol{y} \in \mathbb{R}^n$ was arbitrary, the matrix T must be convergent. Theorem (2.5.3) implies that $\rho(T) < 1$.

The general convergence rate for an iterative method is defined as

$$\rho = \lim_{k\to\infty} \left(\sup_{\boldsymbol{e}^{(0)}\in\mathbb{R}} \frac{\|\boldsymbol{e}^{(k)}\|}{\|\boldsymbol{e}^{(0)}\|} \right)^{\frac{1}{k}},$$

where $\boldsymbol{e}^{(k)}$ is the error in the approximation $\boldsymbol{x}^{(k)}$ to the system $A\boldsymbol{x} = \boldsymbol{b}$ given by

$$\boldsymbol{e}^{(k)} = \boldsymbol{x}^* - \boldsymbol{x}^{(k)}$$

where $\boldsymbol{x}^*$ is the true solution. We have

$$\begin{aligned}
\boldsymbol{e}^{(k)} &= \boldsymbol{x}^* - \boldsymbol{x}^{(k)} \\
&= T\boldsymbol{x}^* + \boldsymbol{c} - (T\boldsymbol{x}^{(k)} + \boldsymbol{c}) \\
&= T(\boldsymbol{x}^* - \boldsymbol{x}^{(k)}) \\
&= T\boldsymbol{e}^{(k-1)} \\
&\vdots \\
&= T\boldsymbol{e}^{(0)}.
\end{aligned}$$

This is equivalent to

$$\begin{aligned}
\rho &= \lim_{k\to\infty}\left(\sup_{\boldsymbol{e}^{(0)}\in\mathbb{R}} \frac{\|\boldsymbol{e}^{(k)}\|}{\|\boldsymbol{e}^{(0)}\|}\right)^{\frac{1}{k}} \\
&= \lim_{k\to\infty} (\|T\|)^{\frac{1}{k}} = \rho(T)
\end{aligned}$$

using the fact that $\lim_{k\to\infty}(\|A\|)^{\frac{1}{k}} = \rho(A)$ for any matrix norm. Therefore the optimal iterative method is the one whose iteration matrix T has minimal spectral radius.

Theorem 2.5.6 ***Spectral Radius for the Jacobi and Gauss Seidel Methods****. If a matrix A has positive diagonal entries and all other entries are negative or zero then only one of the following statements holds*

- $0 < \rho(T_{GS}) < \rho(T_J) < 1$
- $1 < \rho(T_J) < \rho(T_{GS})$
- $\rho(T_J) = \rho(T_{GS}) = 0$
- $\rho(T_J) = \rho(T_{GS}) = 1$

where T_{GS} and T_J are the iteration matrices for Gauss Seidel and Jacobi respectively.

For Theorem (2.5.6), if one of the Jacobi or Gauss Seidel methods converges then so does the other and if one of them is divergent then so is the other. The Gauss Seidel method converges faster than the Jacobi method.

Definition 2.5.7 ***Regular Splitting****. $A = M - N$ is called a regular splitting of A if M is nonsingular and M^{-1} and N are nonnegative.*

The following theorems regarding convergence [51] are stated without proof.

Theorem 2.5.8 *If M and N are a regular splitting of A and $T = M^{-1}N$ then $\rho(T) < 1$ if and only if A is nonsingular and A^{-1} is nonnegative.*

Theorem 2.5.9 *If all of the diagonal elements of A are non-zero then $\rho(T_{SOR}) \geq |\omega - 1|$ and hence SOR converges only when $0 < \omega < 2$.*

Theorem 2.5.10 *If A is positive definite i.e $\boldsymbol{x}^T A\boldsymbol{x} > 0$ for any $\boldsymbol{x}$ and $0 < \omega < 2$ then the SOR method converges for any initial guess $\boldsymbol{x}^{(0)}$.*

Theorem 2.5.11 *If A is positive definite and tridiagonal then $\rho(T_{GS}) = \rho(T_J)^2$ then the optimal value for ω for SOR is*

$$\omega = \frac{2}{1+\sqrt{1-\rho(T_J)^2}}$$

for which $\rho(T_{SOR}) = \omega - 1$.

Practically there are many problems which do not satisfy these sufficient conditions for Jacobi and Gauss Seidel convergence. Then we have to consider Krylov type methods [102] or multigrid methods (in Section 2.7).

2.6 Iterative Solutions of Nonlinear Equations

Let us now say that we want to solve the following nonlinear system

$$\begin{cases} F_1(x_1, x_2, \ldots, x_n) = 0 \\ F_2(x_1, x_2, \ldots, x_n) = 0 \\ \vdots \\ F_n(x_1, x_2, \ldots, x_n) = 0, \end{cases} \tag{2.40}$$

which may be obtained from discretisation of a nonlinear PDE or onlinear optimisation

$$\min \mathcal{J}(x_1, \ldots, x_n) \tag{2.41}$$

such as (2.6). We can represent the system as $\boldsymbol{F}(\boldsymbol{x}) = \boldsymbol{0}$ where

$$\boldsymbol{F} = (F_1, F_2, \ldots, F_n)^T, \quad \boldsymbol{x} = (x_1, x_2, \ldots, x_n)^T, \tag{2.42}$$

and $F_i : D \subset \mathbb{R}^n \to \mathbb{R}$, $i = 1, ..., n$ are nonlinear operators which are continuously differentiable on $\mathbb{R}^n$. We want to find $\boldsymbol{x}^* \in \mathbb{R}^n$, a solution to the equation (2.40). In this section, we will start with the Newton method, before introducing the gradient descent method and the Quasi Newton method. We will then present the line search method.

2.6.1 Newton Method

Let J denote the Jacobian matrix of $\boldsymbol{F}$

$$J_{ij} = \frac{\partial F_i(\boldsymbol{x})}{\partial x_j}$$

and assume that J is Lipschitz continuous with constant L_f in $\mathbb{R}^n$. Newton's method attempts to evaluate $\boldsymbol{F}(\boldsymbol{x}) = \boldsymbol{0}$ using the following recurrence relation

$$\boldsymbol{x}^{(k)} = \boldsymbol{x}^{(k-1)} - \left(J(\boldsymbol{x}^{(k-1)})\right)^{-1} \boldsymbol{F}(\boldsymbol{x}^{(k-1)}).$$

Let $\boldsymbol{d}^{(k-1)}$ denote the descent direction. Then we can write the Newton recurrence relation as

$$\begin{aligned}&\text{Solve } \boldsymbol{d}^{(k-1)} = -\Big(J(\boldsymbol{x}^{(k-1)})\Big)^{-1}\Big(\boldsymbol{F}(\boldsymbol{x}^{(k-1)})\Big),\\ &\text{Update } \boldsymbol{x}^{(k)} = \boldsymbol{x}^{(k-1)} + \boldsymbol{d}^{(k-1)}.\end{aligned}$$

2.6.2 Gradient Descent Method

Consider a general nonlinear functional $F : \mathbb{R}^n \to \mathbb{R}$ and suppose that we want to solve for $\boldsymbol{x}$, where

$$F(\boldsymbol{x}) = 0 \tag{2.43}$$

and $\boldsymbol{x} \in \mathbb{R}^n$. The gradient descent method, also known as the steepest descent method, generates a sequence of $\boldsymbol{x}^{(k)}, k \geq 1$ through the recurrence relation

$$\boldsymbol{x}^{(k)} = \boldsymbol{x}^{(k-1)} + \alpha^{(k-1)}\boldsymbol{d}^{(k-1)}, \quad \boldsymbol{d}^{(k-1)} = -\nabla F(\boldsymbol{x}^{(k-1)})$$

where the positive scalar $\alpha^{(k-1)}$ is called the step length and $\boldsymbol{d}^{(k-1)}$ is the taken search direction. The main characteristic of the gradient descent method is that the iterates decrease the function value at each step

$$F(\boldsymbol{x}^{(k)}) \leq F(\boldsymbol{x}^{(k-1)}). \tag{2.44}$$

If the step length $\alpha^{(k-1)}$ is fixed to the time step Δt of a newly introduced time variable t, the descent method is known as the time marching method. Note this $\leq$ is not $<$ in (2.44), i.e. stagnation could happen.

2.6.3 Quasi Newton Method

Suppose that we are given an optimisation problem (as will be seen many times later)

$$\min_x f(x)$$

where f is convex, twice differentiable and $f : \mathbb{R}^n \to \mathbb{R}$. Note that convexity is a strong requirement in this work. We are given an initial point $x^{(0)}$ and we will use an iterative method to generate a sequence of the solution $x^{(k)}$ that converges to the minimum point x^*. Denote the gradient of f at $x^{(k)}$ by $\nabla f(x^{(k)})$ and the Hessian matrix (the matrix of the second derivative) by $H^{(k)} = \nabla^2 f(x^{(k)})$. The second order Taylor expansion around $x^{(k)}$ is given by

$$\widehat{f}(p) = f(x^{(k)}) + (p)^T\nabla f(x^{(k)}) + \frac{1}{2}(p)^T H^{(k)}(p)$$

where $p = x - x^{(k)}$. $\widehat{f}(p)$ defines a quadratic model of the function near the point $x^{(k)}$. The gradient of $\widehat{f}(p)$ with respect to x is given by

$$\nabla\widehat{f}(p) = \nabla f(x^{(k)}) + H^{(k)}(p).$$

The minimal values for $\nabla\widehat{f}(p)$ occur when $p^{(k)} = -(H^{(k)})^{-1}\nabla f(x^{(k)})$. Then, we have the recurrence relation for the Newton method

$$x^{(k+1)} = x^{(k)} + \alpha^{(k)}p^{(k)}.$$

To find the value of $p^{(k)}$, one has to solve the linear system $H^{(k)}p^{(k)} = -\nabla f(x^{(k)})$ or find the inverse of the Hessian matrix. Thus, the process of calculating $p^{(k)}$ can be too expensive for large scale problems. The quasi Newton method attempts to avoid the computation of the Hessian matrix using an approximation of the Hessian matrix. The first quasi Newton method is DFP (Davidson, Fletcher and Powell) who discovered it in 1959. Instead of computing the true Hessian, we will use an approximation which is based on the change in the gradient between iterations. Another variant of the quasi Newton method is the BFGS (Broyden, Fletcher, Goldfarb and Shanno). It is actually the same as the DFP with a single modification whereby instead of approximating the Hessian $H^{(k)}$, we approximate its inverse $(H^{(k)})^{-1}$. This is considered to be the most effective quasi Newton method.

2.6.4 Line Search Method

We will now consider the step length parameter $\alpha^{(k)}$ in the recurrence relation for the iterative method. It most cases, the initial step size $\alpha^{(0)} = 1$ will be used. However, the success of a line search method depends on the effective choice of both the direction $p^{(k)}$ and the step length $\alpha^{(k)}$. A simple condition which may be imposed on $\alpha^{(k)}$ is to require a reduction in f, that is

$$f(x^{(k)} + \alpha^{(k)}p^{(k)}) \leq f(x^{(k)}).$$

This requirement is not enough to produce convergence to x^* [72]. A popular inexact line search condition stipulates that $\alpha^{(k)}$ should first of all give sufficient decreases in the objective function as measured by the following inequality:

$$f(x^{(k)} + \alpha^{(k)}p^{(k)}) \leq f(x^{(k)}) + c_1\alpha^{(k)}\nabla f(x^{(k)})^T p^{(k)} \tag{2.45}$$

for some constant $c_1 \in (0, 1)$. This inequality is called the Armijo condition. The second useful condition is called the curvature condition which requires $\alpha^{(k)}$ to satisfy

$$\nabla f(x^{(k)} + \alpha^{(k)}p^{(k)}) \geq c_2\nabla f(x^{(k)})^T p^{(k)}$$

for some $c_2 \in (c_1, 1)$ where c_1 is the constant in equation (2.45). The Armijo and curvature conditions are known collectively as the Wolfe conditions.

2.7 Multigrid Methods

Multigrid methods (MG), also known as multilevel methods, have been shown to be an efficient solver for linear and nonlinear elliptic PDEs using a hierarchy of discretisation where there is a pyramid of grids. For example, in Figure 2.5, we have a pyramid of grids with four levels. The top of the pyramid is denoted as level 1 with 4 cell-centred points and the bottom of the pyramid is denoted as level 4 with 16 cell-centred discretisation points. In this section, we will give a brief discussion of the multigrid methods and we will only consider geometric multigrid and not the algebraic multigrid. In the examples of the grid transfer operator, we assume a vertex discretisation as defined in the previous section. For a more comprehensive introduction to the multigrid method see [90, 5] for more details.

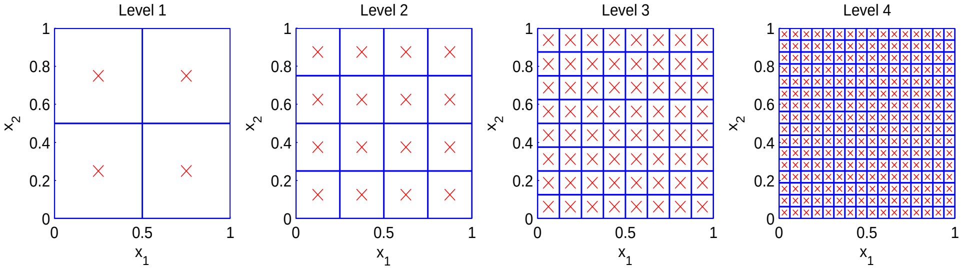


Figure 2.5: Illustration of a pyramid of grid with four levels. Red crosses are the cell-centred discretisation points.

Definition 2.7.1 ***Fourier Mode***. *Given* $\boldsymbol{v}^{(0)} = (v_1^0, \dots, v_n^0)$. *The fourier mode of* v_i^0 *is given by*

$$v_i^0 = \sin\left(\frac{ik\pi}{n}\right)$$

where $0 \leq i \leq n$ *and* $1 \leq k \leq n-1$. *The integer* k *represents the wave number of frequency of* $\boldsymbol{v}^{(0)}$. *We can observe that a small* k *yields a vector* $\boldsymbol{v}^{(0)}$ *with few oscillations along one dimensional grid while a large* k *yields a highly oscillatory* $\boldsymbol{v}^{(0)}$.

2.7.1 The Basic Principles of Multigrid

These methods are based on two principles, the smoothing principle and the coarsening principle. The relaxation or iterative methods discussed in the previous section have a strong smoothing effect on the error. The schemes are effective at removing the high order oscillatory Fourier modes of the error. That does not mean the error becomes small, it just become smooth. The coarsening principle states that smooth error terms have a good approximation on a coarse grid. In the coarse grid, the computation is far less expensive than a fine grid computation. The Nyquist-Shannon sampling theorem stated that only low frequency components of fine grid errors are represented properly on a coarser grid. Consider a linear system

$$A\boldsymbol{u} = \boldsymbol{f}.$$

Let $\boldsymbol{v}$ be an approximation to the solution $\boldsymbol{u}$, then the error is defined as

$$\boldsymbol{e} = \boldsymbol{u} - \boldsymbol{v}$$

and the residual is given by

$$\boldsymbol{r} = \boldsymbol{f} - A\boldsymbol{v} = A\boldsymbol{e}.$$

After a relaxation or smoothing step on the fine grid, the error become smooths and the residual equation will be approximated on the coarse grid. The matrix A is replaced by a simpler approximation $\widehat{A}$ on the coarse grid. So, the basic idea is to smooth the error on the fine grid, transfer the remaining error to the coarser grid, smooth the error there and repeat the process again. The components for multigrid methods are

- Smoother or relaxation strategy:
  If A is replaced by $D - L - U$, we have the Jacobi method and if we use $\boldsymbol{v}^{(k)} = (D-L)^{-1}U\boldsymbol{v}^{(k-1)} + (D-L)^{-1}\boldsymbol{f}$, we have the Gauss Seidel method.

- Coarsening strategy:
  Consider Ω^h with grid spacing (h, k) as the fine grid. We will construct a coarse grid Ω^H with grid spacing (H, K) for the grid Ω^h. A typical standard coarsening is to double the spacing i.e $H = 2h, K = 2k$. If Ω^h has $(n+1) \times (m+1)$ grid points including boundary points then $\Omega^{H=2h}$ will have $(\frac{n}{2}+1) \times (\frac{m}{2}+1)$ grid points including the boundary points. The coarse grid will be a subset of the fine grid. Figure 2.6 illustrates standard coarsening for vertex-centred discretisation points.

- Coarse grid operator:
  Let A_h be the matrix operator on Ω^h. If A_H is just the original operator A discretised on Ω^H, then the method is called the discretisation coarse grid approximation (DCA). An alternative would be the Galerkin operator $I_h^H A_h I_H^h$ where I_h^H is the restriction operator and I_H^h is the interpolation operator.

- Restriction operator (transfer operator from the fine to coarse grid):

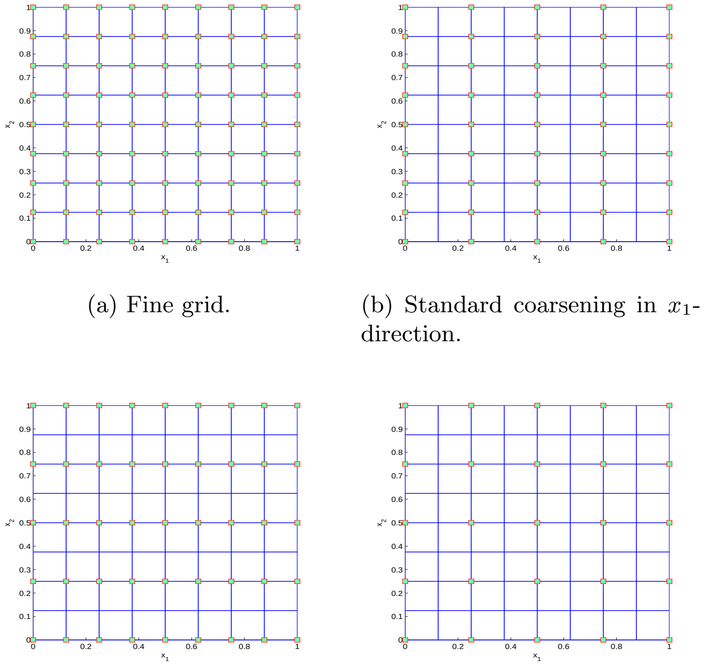
  $$v_{H=2h} = I_h^{H=2h} v_h.$$

  There are three standard restriction operators: injection, half weighting and full weighting. For the injection operator, we have

  $$v_{i,j}^{H=2h} = v_{2i,2j}^h.$$

  Meanwhile, we have

  $$v_{i,j}^{H=2h} = \frac{1}{8}\left[v_{2i,2j-1}^h + v_{2i,2j+1}^h + v_{2i-1,2j}^h + v_{2i+1,2j}^h + v_{2i,2j}^h\right]$$



(a) Fine grid.

(b) Standard coarsening in x_1-direction.

(c) Standard coarsening in x_2-direction.

(d) Coarse grid.

Figure 2.6: Illustration of the standard coarsening strategy. The fine grid in (a) has 9 × 9 discretisation points. An example of semi-coarsening where the coarse grid in (b) is obtained by doubling the mesh size in the x_1-direction. In (c), we obtained the coarse grid by doubling the mesh size in the x_2-direction. Finally, the coarse grid (d) is constructed using these standard procedures.

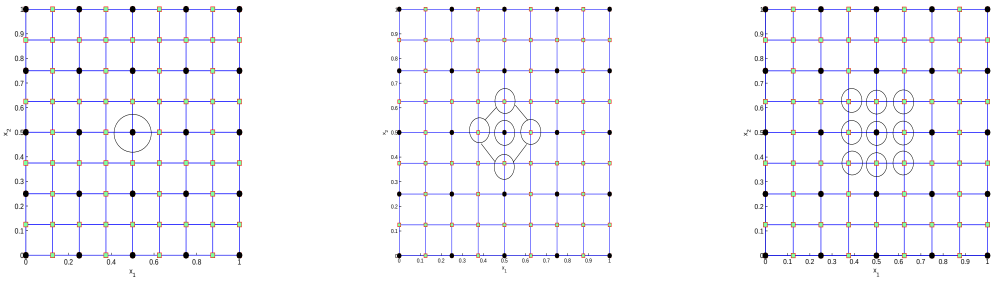
for the half weighting operator and

$$v_{i,j}^{H=2h} = \frac{1}{16}\Big[v_{2i-1,2j-1}^{h} + v_{2i-1,2j+1}^{h} + v_{2i+1,2j-1}^{h} + v_{2i+1,2j+1}^{h} + \\ 2\left(v_{2i,2j-1}^{h} + v_{2i,2j+1}^{h} + v_{2i-1,2j}^{h} + v_{2i+1,2j}^{h}\right) + 4v_{2i,2j}^{h}\Big]$$

for the full weighting operator.

- Interpolation operator (transfer operator from the coarse grid to the fine grid): In multigrid theory, there is a rule that the sum of the orders of the restriction and interpolation operator should be larger than the order of the differential operator [90]. An interpolation operator has interpolation order b if it preserves a polynomial of order $b-1$. The most commonly used interpolation or prolongation operator is the bilinear operator

$$v_h = I_{H=2h}^{h} v_{H=2h}$$



(a) Injection operator. (b) Half weighting operator. (c) Full weighting operator.

Figure 2.7: Illustration of the restriction operators. (a) is the injection operator, (b) is the half weighting operator and (c) is the full weighting operator for vertex-centred discretisation. The points in circles are the active points used to obtain the coarse points in black circles for each operator.

where

$$\begin{aligned} v^h_{2i,2j} &= v^{2h}_{i,j} \\ v^h_{2i+1,2j} &= \frac{1}{2}(v^{2h}_{i,j} + v^{2h}_{i+1,j}) \\ v^h_{2i,2j+1} &= \frac{1}{2}(v^{2h}_{i,j} + v^{2h}_{i,j+1}) \\ v^h_{2i+1,2j+1} &= \frac{1}{4}\left(v^{2h}_{i,j} + v^{2h}_{i+1,j} + v^{2h}_{i,j+1} + v^{2h}_{i+1,j+1}\right). \end{aligned}$$

The coarse grid point that coincides with the fine grid point is left unchanged, and the surrounding fine grid points receive a contribution depending on the neighbourhood relation. This operator is the adjoint operator to the full weighting operator. We can also represent the bilinear operator using stencil notation as follows:

$$I^h_{H=2h} = \left] \begin{matrix} \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ \frac{1}{2} & 1 & \frac{1}{2} \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \end{matrix} \right[^{h}_{2h=H}.$$

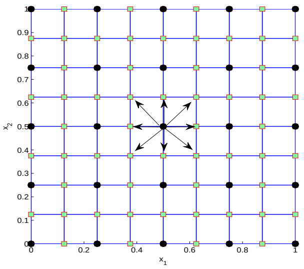


Figure 2.8: Illustration of bilinear operator from the coarse grid to the fine grid. The coarse point in black circles are used to obtain the nine fine points surrounding it.

- Cycle types:

  The common multigrid cycles are V-cycle and W-cycle shown in Figure 2.9.

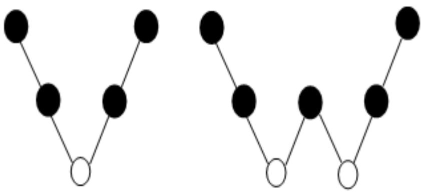


Figure 2.9: Illustration of multigrid cycles with three levels of grid. Left is the V-cycle and on the right is the W-cycle. The white circles denote the coarsest grid, \ and / denote the restriction and interpolation steps, respectively.

The components for multigrid are chosen based on the problem to be solved. Each component can be tailored to the specific need to ensure the convergence of the method.

2.7.2 Two Grid Cycle

Let $u_h^{(k)}$ be an approximation to the solution u_h of the discrete elliptic boundary problem $A_h u_h = f_h$. The error is

$$e_h^{(k)} = u_h - u_h^{(k)}$$

and the defect or residual is given by

$$r_h^{(k)} = f_h - A_h u_h^{(k)}.$$

The defect equation

$$A_h e_h^{(k)} = r_h^{(k)} \tag{2.46}$$

is equivalent to the original problem. Define an iteration

$$u_h^{(k+1)} = u_h^{(k)} + \widehat{e}_h^{(k)}$$

by replacing A_h in equation (2.46) by $\widehat{A}_h$ and where $\widehat{e}_h^{(k)}$ is the solution for $\widehat{A}_h \widehat{e}_h^{(k)} = \widehat{r}_h$. At the coarse grid, A_h is approximated by A_H where $H > h$. Then the defect becomes

$$\widehat{A}_H \widehat{e}_H^{(k)} = r_H^{(k)}.$$

As mentioned before, we need the restriction and interpolation operator to transfer the residual using

$$r_H^{(k)} = I_h^H r_h^{(k)}, \text{ and } \widehat{r}_h^{(k)} = I_H^h \widehat{r}_H^{(k)}.$$

Algorithm 2 outlines all steps in two grid cycles. The multigrid cycle is an application of the two grid cycle recursively.

2.7.3 Multilevel Framework

One of the variants of the multigrid methods is the multilevel framework where the procedure starts with restricting the problem to the several levels of the grid until the

Algorithm 2 Two Grid Cycle
$u_h^{(k+1)} \leftarrow \text{TwoGridCycle}(A_h, f_h, u_h^{(k)})$

```
1. Compute the defect $r_h^{(k)} \leftarrow f_h - A_h u_h^{(k)}$.
2. $H \leftarrow 2h$.
3. Restrict the defect $d_H^{(k)} \leftarrow I_h^H r_h^{(k)}$.
4. Solve on $\Omega_H$, $\widehat{e}_H^{(k)} \leftarrow A_H^{-1} r_H^{(k)}$.
5. Interpolate the coarse grid correction $\widehat{e}_h^{(k)} \leftarrow I_H^h \widehat{e}_H^{(k)}$.
6. Update the solution $u_h^{(k+1)} = u_h^{(k)} + \widehat{e}_h^{(k)}$.
```

minimum level or the coarsest level is reached. At the coarsest level, the solution is found using iterative or direct methods. Then the solution is prolonged to the next fine level until the finest level. The procedure is illustrated in Algorithm 3. The multilevel approach here is comparable to the full multigrid method (FMG) known from multigrid applications. See [13, 90] and the references therein for more details.

Algorithm 3 Multilevel Framework
$u_h^* \leftarrow \text{MultilevelFramework}(A_h, f_h, u_h^{(0)}, l)$

```
1. If $l = minlevel$ then
   solve on the coarsest level $u_{h_l}^* \leftarrow A_h^{-1} f_h$.
2. Else
   (a) $h \leftarrow 2^l h$.
   (b) MultilevelFramework$(A_h, f_h, u_{h_{l-1}}, l)$
   (c) $u_{h_l} \leftarrow$ Interpolate$(u_{h_{l-1}})$
3. End if.
```

Chapter 3

Mathematical Models for Image Registration and Segmentation

In this chapter, a review of mathematical models for image registration will be presented. We begin with the general idea of image registration models and two similarity measures for quantifying the differences between images. We then review several models for parametric and non-parametric image registration. We also introduce two solution schemes to solve the minimisation problems arising from image registration. Interpolation methods also play an important role in image registration. Thus, we discuss several techniques for interpolation. We briefly introduce image segmentation models which are useful to know for Chapter 7.

3.1 Introduction

Image registration is one of the fundamental tasks of medical imaging. After twenty years of development in medical image analysis, image registration is the key sanctioning technology in this particular area [43]. The broad range of imaging modalities such as Magnetic Resonance Imaging (MRI), Computer Tomography (CT), and Positron Emission Tomography (PET) is the main reason for the rapid improvement in image registration allowing for information from different modalities to be compared, analysed and combined [36].

Image registration is the process of establishing correspondences of the features between images acquired from the same patient known as intra-subject registration or from different patients known as inter-subject registration. In the first case, the same patient will undergo the scanning process before treatment (pre-treatment) and after treatment (post-treatment) using the same imaging modality (mono-modality) or different imaging modalities (multi-modality) [81]. Such correspondences can be used to transform one particular image so that its appearance becomes similar to its pair. Given this transformation, this pair of images can then be used for further medical tasks such as automatic segmentation, treatment planning and radiation therapy.

For the second case, several individuals from the same or different populations will undergo scanning using a particular imaging machine. The data set obtained will be used to study anatomical variability across populations where the images of the different subjects are used to characterise the differences between individuals in a certain population or the differences between an individual and a reference image [81]. The most natural application of registration is to correct the patient's movements during scanning. For example, in MR mammography for the detection of breast cancer, patients will undergo scanning before and after injection of a contrast agent that is used to distinguish between normal and abnormal tissues. However, as mentioned before, it is not only limited to correcting the patient's motion.

3.1.1 Image Registration Model

Image registration is the process of finding a geometric transformation between two images known as the reference R and template T. Two main ingredients are combined in order to compute a transformation which matches the given images. First, there must be a measure of the image similarity to calculate how much these two images are equal. Second, there must be a measure of the regularity of the transformation. The second term also known as the regularisation or penalty, is used to include prior knowledge of the transformation and mathematically transforms the registration problem to a well-posed one. Typically, the problem is modelled as a minimisation problem of an energy functional which consists of a weighted sum of these two measures. We can write the problem as

$$\min_{\boldsymbol{\varphi}} \Big\{ \mathcal{J}(T, R, \boldsymbol{\varphi}) = \mathcal{D}(T, R, \boldsymbol{\varphi}) + \gamma \mathcal{S}(\boldsymbol{\varphi}) \Big\} \tag{3.1}$$

where T, R are the given images, $\mathcal{D}$ is the similarity measure, $\mathcal{S}$ is the smoothness term and $\boldsymbol{\varphi}$ is the transformation which we are aiming to find. $\gamma > 0$ is the regularisation parameter which measures the trade-off between the similarity and smoothness terms. A common similarity measure in mono-modal image registration, where the intensity values of the images T and R are comparable, is the sum of the squared difference (SSD). This particular distance measure assumes that images are identical at registration except for noise [20]. It is given by the L_2 norm of the difference between T and R as follows

$$\mathcal{D}^{\mathrm{SSD}}(T, R, \boldsymbol{\varphi}(\boldsymbol{x})) = \frac{1}{2} \int_\Omega (T(\boldsymbol{\varphi}(\boldsymbol{x})) - R(\boldsymbol{x}))^2 \,\mathrm{d}\Omega. \tag{3.2}$$

The functional $\mathcal{D}$ is defined on a function space χ where χ is usually a Hilbert space. χ is equipped with the scalar product

$$< \boldsymbol{\varphi}, \boldsymbol{\psi} >_\chi = \int_\Omega \boldsymbol{\varphi} \cdot \boldsymbol{\psi} \,\mathrm{d}\Omega = \int_\Omega < \boldsymbol{\varphi}, \boldsymbol{\psi} >_{\mathbb{R}^d} \mathrm{d}\Omega$$

where $< \cdot, \cdot >_{\mathbb{R}^d}$ is the Euclidean scalar product.

For multi-modal image registration, where images are acquired from different imaging machines, a well known similarity measure is the mutual information (MI) measure [96] which assumes only a probabilistic relationship between pixels or voxels in the images. The term is given by

$$\mathcal{D}^{\mathrm{MI}}(T(\boldsymbol{\varphi}(\boldsymbol{x})), R) = H(T(\boldsymbol{\varphi}(\boldsymbol{x}))) + H(R) - H(T(\boldsymbol{\varphi}(\boldsymbol{x})), R) \tag{3.3}$$

where $H(T(\boldsymbol{\varphi}(\boldsymbol{x})))$ and $H(R)$ are defined as the individual entropies for T and R respectively and $H(T(\boldsymbol{\varphi}(\boldsymbol{x})), R)$ is the joint entropy of T and R.

The transformation model in image registration refers to the way in which the images are transformed. There exist many image registration methods and different ways of classifying them. In [65], the authors have suggested a nine dimensional scheme to categorise them. Following the classification of deformation models given by [45] for geometric transformation in terms of parameters, registration methods can be categorised as either parametric or non-parametric [67]. The first type of model consists of only a few parameters. For example, in rigid registration there are three global parameters for 2D images which describe translation and rotation. For the second category, which originate from physical models such as diffusion [88], elastic [6] and curvature [24] registration, the search space is much larger where the process of finding the transformation involves every pixel or voxel which is very expensive in terms of the computational cost. For example, there are $2 \times 512^2 = 524288$ unknowns to determine for a typical CT slice of size 512×512 pixels which correspond to an element of about $0.5 \times 0.5mm^2$ in area and nowadays many images are acquired directly as 3D volumes [36]. Even though there is still a finite number of parameters, this particular category aims to find a smooth transformation rather than a set of parameters.

The optimisation procedure is used to find the best possible transformation and should be quick and reliable. There exist some standard methods of finding the parameters in order to have an optimal objective function, for instance the gradient descent and the conjugate gradient methods. For non-parametric image registration, the minimiser is often described in terms of partial differential equations (PDEs). It should be noted that choosing an optimisation scheme is problem-dependent.

3.1.2 Mathematical Setting

An image I is considered as a compactly supported function which maps a domain Ω into a set of real numbers $V \subset \mathbb{R}_0^+$. The domain Ω is a subset of $\mathbb{R}^d$ where d is the dimensionality of the data with smooth boundary $\partial\Omega$. Practically, in many real applications, mainly medical imaging, we have $d = 3$. However, throughout this chapter, $d = 2$ but with some extra work it can be extended to $d = 3$. Usually, I is a set of measurements obtained by the integration of some density field over a finite area Ω. In this thesis, images are limited to scalar or grey intensity images. Digital images are the discrete setting for the continuous image I. The domain Ω is subdivided into cells with certain height (h_2) and width (h_1) which are called pixels or picture elements. For each pixel, an intensity value or grey level is assigned. We use $\Omega = (0, 1)^2$ or $\Omega = (0, N_1) \times (0, N_2)$ to denote the spatial domain where N_1 and N_2

are coordinates on the x_1 and x_2 axes respectively. We use the left handed coordinate system where $\boldsymbol{x} = (x_1, x_2)^T$ and rearrange the 2-dimensional array as long vectors using lexicographical ordering. In summary,

$$\begin{aligned} &I : \mathbb{R}^2 \to V, \Omega = (0, N_1) \times (0, N_2), \boldsymbol{x} = (x_1, x_2)^T, \\ &I(\boldsymbol{x}) = \mathbb{R}_0^+ \ \ \forall \boldsymbol{x} \in \Omega, \ \text{ if } \ \boldsymbol{x} \notin \Omega, \ \ I(\boldsymbol{x}) = 0. \end{aligned}$$

In registration, we are given two images denoted as the reference R and template T. The image R is kept unchanged in the process and we will transform the image T so that it appears similar to R. The transformation is denoted by

$$\boldsymbol{\varphi} = \boldsymbol{\varphi}(\boldsymbol{x}) : \Omega \to \mathbb{R}^d.$$

The transformation $\boldsymbol{\varphi}$ can be expressed as

$$\boldsymbol{\varphi}(\boldsymbol{x}) = \boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x}) \tag{3.4}$$

where $\boldsymbol{u}(\boldsymbol{x})$ denotes the displacement field. The transformed template image is denoted by

$$T \circ \boldsymbol{\varphi}(\boldsymbol{x}) = T(\boldsymbol{\varphi}(\boldsymbol{x})) = T(\boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x})).$$

After the corresponding location $\boldsymbol{\varphi}(\boldsymbol{x})$ is calculated for each spatial location $\boldsymbol{x} \in \Omega$, an interpolation step is required to assign the intensity values for the transformed template $T(\boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x}))$ at non-grid locations.

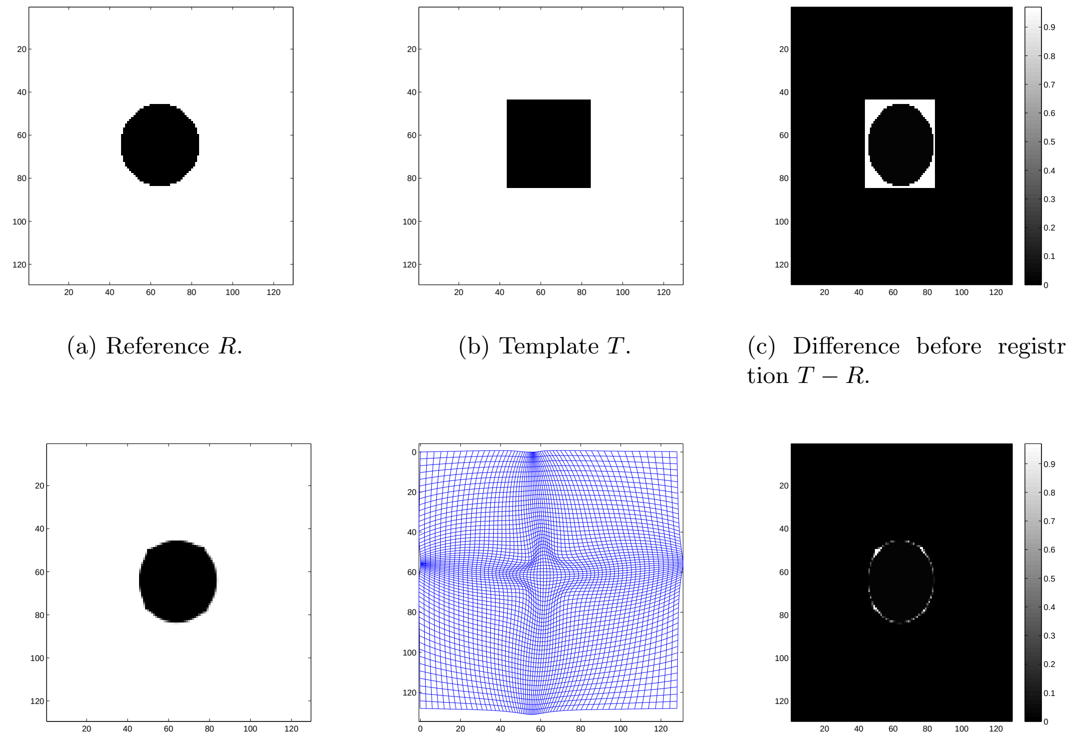
We exploit the Eulerian frame for the transformation instead of the Lagrange frame as is commonly done in image registration [68, 67]. We make the assumption that the transformation is invertible in the Eulerian frame. Assume the template image T is the deformed version of the reference image R through the transformation $\boldsymbol{\phi}$

$$R \xrightarrow{\boldsymbol{\phi}} T.$$

Then, for any $\widetilde{\boldsymbol{x}} \in R$ and $\widehat{\boldsymbol{x}} \in T$, $\widehat{\boldsymbol{x}} = \boldsymbol{\phi}(\widetilde{\boldsymbol{x}})$, or equivalently we can write $\widetilde{\boldsymbol{x}} = \boldsymbol{\phi}^{-1}(\widehat{\boldsymbol{x}})$. Letting the transformation $\boldsymbol{\varphi}(\boldsymbol{x})$ be given as $\boldsymbol{\varphi}(\boldsymbol{x}) = \boldsymbol{\phi}^{-1}(\boldsymbol{x})$, we can write $\boldsymbol{\varphi}(\boldsymbol{x})$ as $\boldsymbol{\varphi}(\boldsymbol{x}) = \boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x})$ as before and write

$$\boldsymbol{\varphi}^{-1}(\boldsymbol{x}) = \boldsymbol{x} - \boldsymbol{u}(\boldsymbol{x}) = \boldsymbol{\phi}(\boldsymbol{x}).$$

where $\boldsymbol{\varphi}^{-1}(\boldsymbol{x})$ is the inverse transformation. In the literature, we can see that the definition of $\boldsymbol{\varphi}(\boldsymbol{x})$ is interchangeably used with $\boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x})$ and $\boldsymbol{x} - \boldsymbol{u}(\boldsymbol{x})$. An illustration of the image registration problem is given in Figure 3.1 where we want to register a square to a circle. The resulting transformation $\boldsymbol{\varphi}$ and the transformed template image $T(\boldsymbol{\varphi}(\boldsymbol{x}))$ are the output for the given images R and T.



(a) Reference R.

(b) Template T.

(c) Difference before registration $T - R$.

(d) Transformed template $T(\boldsymbol{\varphi}(\boldsymbol{x}))$.

(e) Transformation $\boldsymbol{\varphi}(\boldsymbol{x})$ is applied to a regular grid.

(f) Difference after registration $T(\boldsymbol{\varphi}) - R$.

Figure 3.1: Illustration of an image registration problem. Reference and template images are given in (a) and (b) respectively. The difference before registration is given in (c) and (d) is the transformed template image using the transformation in (e). (f) is the difference image after registration and we can observe that the difference image is reduced after registration. Notice that the transformed template image looks similar to the reference image after registration.

3.1.3 Variational Formulation of Image Registration

There exist many approaches to tackling issues in image registration, such as statistical methods, but our concern is with the variational approach which has been shown to provide good results as well as interesting mathematical problems. Referring to equation (3.4), the transformation is expressed in terms of the displacement field $\boldsymbol{u}(\boldsymbol{x})$ where finding $\boldsymbol{\varphi}(\boldsymbol{x})$ is equivalent to finding $\boldsymbol{u}(\boldsymbol{x})$. $\boldsymbol{u}(\boldsymbol{x})$ is sought over an admissible Hilbert space χ. The joint functional can be written as in (3.1) or equivalently as

$$\min_{\boldsymbol{u}(\boldsymbol{x})\in\chi} \Big\{ \mathcal{J}(T, R, \boldsymbol{u}(\boldsymbol{x})) = \mathcal{D}(T, R, \boldsymbol{u}(\boldsymbol{x})) + \gamma \mathcal{S}(\boldsymbol{u}(\boldsymbol{x})) \Big\}. \tag{3.5}$$

A necessary condition for a minimiser $\boldsymbol{u}(\boldsymbol{x})$ of $\mathcal{J}$ is the vanishing of the Gâteaux derivative of $\mathcal{J}$ for all variational directions $\boldsymbol{v}(\boldsymbol{x}) \in \chi$, i.e

$$\delta\mathcal{J}(\boldsymbol{u}; \boldsymbol{v}) = \lim_{\varepsilon\to 0} \frac{\mathcal{J}(\boldsymbol{u} + \varepsilon\boldsymbol{v}) - \mathcal{J}(\boldsymbol{u})}{\varepsilon} = \frac{d}{d\varepsilon}\mathcal{J}(\boldsymbol{u} + \varepsilon\boldsymbol{v})\Big|_{\varepsilon=0} = 0.$$

Since,

$$\delta \mathcal{J}(\boldsymbol{u};\boldsymbol{v}) =< \nabla_{\boldsymbol{u}}\mathcal{J}, \boldsymbol{v} >_{\chi}$$

where $\nabla_{\boldsymbol{u}}\mathcal{J}$ is the gradient of the functional $\mathcal{J}$. The necessary condition is given by

$$\nabla_{\boldsymbol{u}}\mathcal{J} = 0, \tag{3.6}$$

also known as the Euler-Lagrange equation for the minimisation problem in (3.5), which can be computed easily when both terms $\mathcal{D}$ and $\mathcal{S}$ are known.

For two given mono-modal images T and R, the Gâteaux derivative for the term $\mathcal{D}^{\text{SSD}}(T, R, \boldsymbol{u}(\boldsymbol{x}))$ is given by:

$$\boldsymbol{f}(\boldsymbol{u}) = (f_1(\boldsymbol{u}(\boldsymbol{x})), f_2(\boldsymbol{u}(\boldsymbol{x})))^T = (T(\boldsymbol{x}+\boldsymbol{u}(\boldsymbol{x})) - R(\boldsymbol{x}))\nabla_{\boldsymbol{u}}T(\boldsymbol{x}+\boldsymbol{u}(\boldsymbol{x})) \tag{3.7}$$

where

$$\nabla_{\boldsymbol{u}}T(\boldsymbol{x}+\boldsymbol{u}(\boldsymbol{x})) = \left[\partial_{\boldsymbol{u}_1}T(\boldsymbol{x}+\boldsymbol{u}(\boldsymbol{x})), \partial_{\boldsymbol{u}_2}T(\boldsymbol{x}+\boldsymbol{u}(\boldsymbol{x}))\right]^T.$$

$\boldsymbol{f}$ is called the force term throughout this thesis. For the general smoothness term $\mathcal{S}$, which will be discussed later, we can write the Gâteaux derivative of $\mathcal{S}$

$$\delta\mathcal{S}(\boldsymbol{u};\boldsymbol{v}) = \int_{\Omega} < \mathcal{A}(\boldsymbol{u}), \boldsymbol{v} >_{\mathbb{R}^d} \mathrm{d}\Omega$$

where $\mathcal{A}$ is a partial differential operator associated with the smoothness term $\mathcal{S}$. Using Gauss' theorem and the fundamental lemma of the calculus of variations, we have

$$\gamma\mathcal{A}(\boldsymbol{u}) + \boldsymbol{f} = \boldsymbol{0},\ \boldsymbol{x} \in \Omega. \tag{3.8}$$

From the derivation of the Gâteaux derivative of $\mathcal{S}$, the boundary conditions for equation (3.8) come from the vanishing of the line integral on the boundary $\partial\Omega$ which are depending on the regularisation term $\mathcal{S}$. We can see that the tasks of finding a minimiser $\boldsymbol{u}$ of $\mathcal{J}$ and solving the Euler-Lagrange equation in (3.8) are equivalent.

3.2 Similarity Measures

Given images T and R of size $N_1 \times N_2$, let

$$T = \{t_i, i = 1, ..., N_1N_2\} \text{ and } R = \{r_i, i = 1, ..., N_1N_2\}$$

denote the intensity values for R and T. The similarity or dissimilarity term is a measure that quantifies the dependency or interdependency between the two sequences [32]. We will discuss two measures for mono-modality image registration.

3.2.1 Sum of the Squared Difference (SSD)

The SSD is the simplest distance measure for images coming from the same imaging machine and is given in equation (3.2). It is the optimum measure when two images only differ by Gaussian noise [96]. The SSD measure is widely used in MRI and is very sensitive to a small number of pixels which have very large intensity differences for images T and R. For example, when patients are scanned before and after injection of the contrast agent, the intensity values for the same object in R and T are no longer equal. One of the remedies is to use the sum of the absolute difference (SAD) defined as follows:

$$\mathcal{D}^{\text{SAD}}(T, R, \boldsymbol{\varphi}) = \int_\Omega |T(\boldsymbol{\varphi}(\boldsymbol{x})) - R(\boldsymbol{x})| \, \mathrm{d}\Omega.$$

3.2.2 Cross Correlation (CC)

If we assume a linear relationship between the intensity values of R and T such that

$$\lambda T(\boldsymbol{\varphi}(\boldsymbol{x})) = \mu R(\boldsymbol{x})$$

where λ and μ are two scalars, the optimum similarity measure is the normalised cross correlation (NCC)

$$\mathcal{D}^{\text{NCC}}(T, R, \boldsymbol{\varphi}) = \frac{< T(\boldsymbol{\varphi}(\boldsymbol{x})), R(\boldsymbol{x}) >}{\|T(\boldsymbol{\varphi}(\boldsymbol{x}))\| \|R(\boldsymbol{x})\|}$$

where

$$< T(\boldsymbol{\varphi}(\boldsymbol{x})), R(\boldsymbol{x}) >= \int_\Omega T(\boldsymbol{\varphi}(\boldsymbol{x}))R(\boldsymbol{x}) \, \mathrm{d}\Omega,$$

$$\|T(\boldsymbol{\varphi}(\boldsymbol{x}))\| = \sqrt{< T(\boldsymbol{\varphi}(\boldsymbol{x})), T(\boldsymbol{\varphi}(\boldsymbol{x})) >}$$

and similarly for $\|R(\boldsymbol{x})\|$.

3.3 Parametric Image Registration

There are two ways to introduce regularisation into image registration. First, using a small number of parameters where the parameters follow a specified model. Second, using a smoothness or regularisation term $\mathcal{S}(\boldsymbol{\varphi}(\boldsymbol{x}))$. We note that, in the first method, we still can have a smoothness measure for the parameters. In this section, we will review three basic parametric models which are the so-called rigid, affine and projective models. In medical imaging, bones usually undergo rigid deformation and soft tissues are deformed non-rigidly during the scanning process.

3.3.1 Rigid Transformation

Rigid transformation is the simplest model which allows only rotation and translation. It can be expressed as:

$$\boldsymbol{\varphi}(\boldsymbol{x}) = \begin{bmatrix} \varphi_1(\boldsymbol{x}) \\ \varphi_2(\boldsymbol{x}) \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ \sin\theta & -\cos\theta \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b} \qquad (3.9)$$

where θ is the rotation angle and $\boldsymbol{b}$ is the translation vector.

3.3.2 Affine Transformation

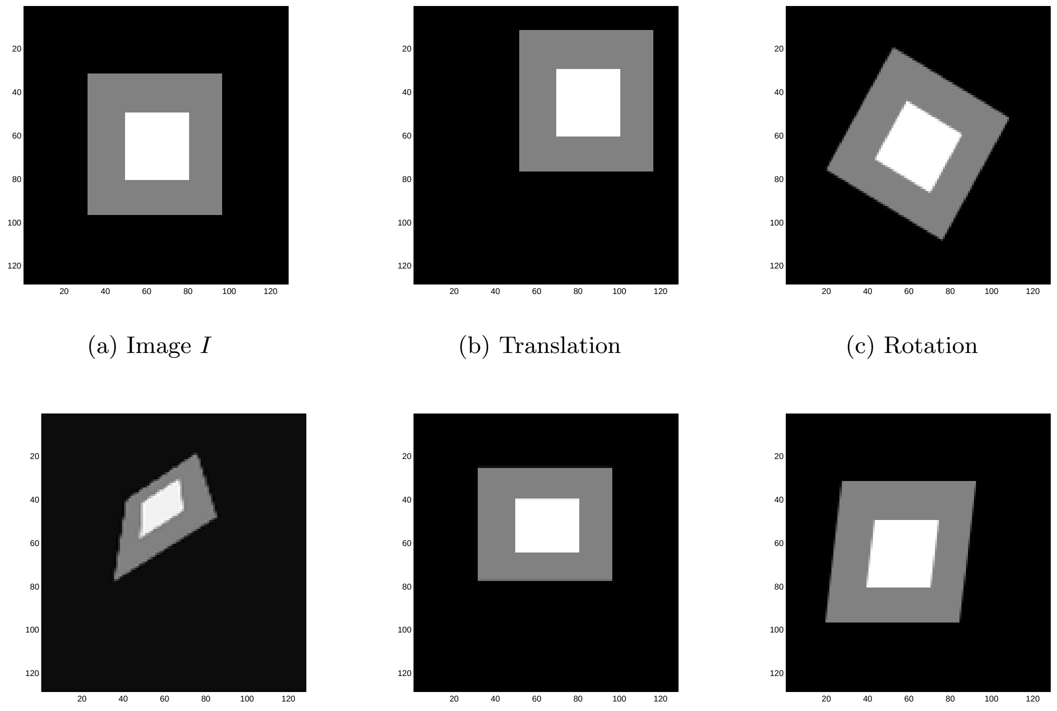
The transformation is given by:

$$\boldsymbol{\varphi}(\boldsymbol{x}) = \begin{bmatrix} \varphi_1(\boldsymbol{x}) \\ \varphi_2(\boldsymbol{x}) \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}. \tag{3.10}$$

The model allows for rotation, scaling, shearing and translation. Since there are only 6 parameters for 2D images, the model is much faster than nonlinear variational models. It is the most popular method for pre-registration in clinical applications. The coefficient matrix $\boldsymbol{A}$ represents a combination of rotation, scaling and shearing via

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ \sin\theta & -\cos\theta \end{bmatrix} \begin{bmatrix} c_{x_1} & 0 \\ 0 & c_{x_2} \end{bmatrix} \begin{bmatrix} s_{x_1} & 1 \\ 1 & s_{x_2} \end{bmatrix}$$

where $\theta, c_{x_1}, c_{x_2}, s_{x_1}$, and s_{x_2} are the parameters for rotation, scaling and shearing in x_1 and x_2 direction respectively. The vector $\boldsymbol{b}$ represents translation in the x_1 and x_2 directions. See Figure 3.2 for an example of the affine transformation of a given image I.



(a) Image I

(b) Translation

(c) Rotation

(d) Projective

(e) Scaling

(f) Shearing

Figure 3.2: Illustration of translation, rotation, scaling, shearing and projective transformation for the image I.

3.3.3 Projective Transformation

Projective transformation maps lines onto lines and does not preserved parallelism between lines. The transformation consists of eight parameters and is defined by

$$\begin{bmatrix} \varphi_1(\boldsymbol{x}) \\ \varphi_2(\boldsymbol{x}) \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ 1 \end{bmatrix}.$$

3.4 Non-parametric Image Registration

We now introduce the non-parametric image registration models based on variational or energy minimisation techniques. The non-parametric image registration method takes the form

$$\min_{\boldsymbol{u}(\boldsymbol{x})} \Big\{ \mathcal{J}_\gamma(\boldsymbol{u}(\boldsymbol{x})) = \mathcal{D}(T, R, \boldsymbol{u}(\boldsymbol{x})) + \gamma \mathcal{S}(\boldsymbol{u}(\boldsymbol{x})) \Big\}, \tag{3.11}$$

where the choice of regulariser $\mathcal{S}(\boldsymbol{u})$ differentiates different models. $\gamma \in \mathbb{R}_{>0}$ is known as the regularisation parameter and measures the trade-off between the fitting term $\mathcal{D}$ and regularisation term $\mathcal{S}$. There exists a large amount of literature regarding non-parametric models such as in elastic [6, 2, 29, 67], fluid flow [15, 14], and diffusion [23, 88, 74]. To complete the variational formulation (3.11), we now specify the two terms.

Firstly, given two mono-modal images R, T, the SSD fitting term $\mathcal{D}$ is given by equation (3.2). The gradient of (3.2) with respect to the deformation $\boldsymbol{u}$ is given by equation (3.7). The force term $\boldsymbol{f}(\boldsymbol{u})$ is nonlinear. Secondly, adopting different regularisers for the second term $\mathcal{S}(\boldsymbol{u}(\boldsymbol{x}))$ will lead to different non-parametric image registration models.

In this section, variational models with several regularisation terms are briefly reviewed.

3.4.1 Linear Elastic Image Registration

Linear elastic image registration is based on the linearised elastic potential $\boldsymbol{u}$. It is the most popular choice due to the physical property of the model. The regularisation term for linear elastic [6, 2, 29, 67] is

$$\mathcal{S}^{\text{LE}}(\boldsymbol{u}) = \int_\Omega \frac{\mu}{4} \sum_{l,m=1}^{2} (\partial_{x_l} u_m + \partial_{x_m} u_l)^2 + \frac{\lambda}{2} (\text{div } \boldsymbol{u})^2 \ \mathrm{d}\Omega$$

where μ and λ are the Lame constants, μ is the shear modulus that refers to the rigidity that estimates the stiffness of the material and λ is related to the bulk modulus. The Euler-Lagrange equation for (3.11) with $\mathcal{S}^{\text{LE}}$ as the regularisation term is the Navier Lame equation:

$$-\gamma(\mu \nabla^2 \boldsymbol{u} + (\mu + \lambda) \nabla \text{div } \boldsymbol{u}) + \boldsymbol{f}(\boldsymbol{u}) = \boldsymbol{0}$$

which is a second order nonlinear PDE with boundary conditions as follows:

$$\nabla \cdot \boldsymbol{u} = 0, \ (\nabla u_l + \partial_{x_l} \boldsymbol{u}) \cdot \boldsymbol{n} = 0, \ l = 1, 2$$

where $\boldsymbol{n}$ denotes the outward unit normal vector on the boundary $\partial\Omega$. The boundary conditions are replaced with Dirichlet or Neumann boundary conditions because images have a uniform background and objects inside the images are far from the boundary [67]. This variational model allows small deformation and penalises affine linear transformation.

3.4.2 Nonlinear Elastic Image Registration

The regularisation term for nonlinear elastic image registration [105, 106, 62] is

$$\begin{aligned}\mathcal{S}^{\text{NLE}}(\boldsymbol{u}) = \int_\Omega & \frac{\lambda}{8}\left(2(\text{div}\boldsymbol{u}) + \sum_{k=1}^{2} |\nabla u_k|^2\right)^2 \\ & + \frac{\mu}{4}\left(\sum_{i=1}^{2}\left[2\frac{\partial u_i}{\partial x_i} + \sum_{k=1}^{2}\left(\frac{\partial u_k}{\partial x_i}\right)^2\right]^2 + \sum_{i,j=1,i\neq j}^{2}\left[\frac{\partial u_j}{\partial x_i} + \frac{\partial u_i}{\partial x_j} + \sum_{k=1}^{2}\frac{\partial u_k}{\partial x_i}\frac{\partial u_k}{\partial x_j}\right]^2\right) \mathrm{d}\Omega\end{aligned}$$

where λ and μ are Lame constants. The model is known to recover large deformation provided that the re-gridding step in [15] is incorporated.

3.4.3 Hyperelastic Energy for Image Registration

The term is given by

$$\mathcal{S}^{\text{Hyper}}(\boldsymbol{u}) = \int_\Omega \alpha_1 \text{length}(\boldsymbol{u}) + \alpha_2 \text{surface}(\boldsymbol{u}) + \alpha_3 \text{volume}(\boldsymbol{u}) \, \mathrm{d}\Omega$$

where $\alpha_i > 0, i = 1, 2, 3$ are some parameters and

$$\begin{aligned}\text{length}(\boldsymbol{u}) &= \left(\left(\frac{\partial u_1}{\partial x_1} - 1\right)^2 + \left(\frac{\partial u_1}{\partial x_2}\right)^2 + \left(\frac{\partial u_2}{\partial x_1}\right)^2 + \left(\frac{\partial u_2}{\partial x_2} - 1\right)^2\right), \\ \text{surface}(\boldsymbol{u}) &= \left(\max\left\{\left(\frac{\partial u_2}{\partial x_2}\right)^2 + \left(\frac{\partial u_2}{\partial x_1}\right)^2 + \left(\frac{\partial u_1}{\partial x_1}\right)^2 + \left(\frac{\partial u_1}{\partial x_2}\right)^2 - 3, 0\right\}^2 - 3\right)^2, \\ \text{volume}(\boldsymbol{u}) &= \left(\frac{\left(\frac{\partial u_1}{\partial x_1}\frac{\partial u_2}{\partial x_2} - \frac{\partial u_1}{\partial x_2}\frac{\partial u_2}{\partial x_1} - 1\right)^2}{\frac{\partial u_1}{\partial x_1}\frac{\partial u_2}{\partial x_2} - \frac{\partial u_1}{\partial x_2}\frac{\partial u_2}{\partial x_1}}\right)^2.\end{aligned}$$

See [7] for more details.

3.4.4 Fluid Registration

The class of deformation models based on motion can be classified into fluid flow and optical flow. Fluid flow registration is based on the spatial smoothing of the velocity $\boldsymbol{v}$ where

$$\boldsymbol{v}(\boldsymbol{x},t) = \frac{\partial \boldsymbol{u}(\boldsymbol{x},t)}{\partial t}.$$

Fluid flow registration is related to linear elastic registration via

$$\mathcal{S}^{\text{fluid}}(\boldsymbol{v}(\boldsymbol{x},t)) = \mathcal{S}^{\text{LE}}(\boldsymbol{v}(\boldsymbol{x},t)).$$

Thus, the Euler-Lagrange equation for (3.11) with $\mathcal{S}^{\text{fluid}}$ as the regularisation term is given by:

$$-\gamma\left[\mu\Delta\boldsymbol{v} + (\mu+\lambda)\nabla \text{ div } \boldsymbol{v}\right] + \boldsymbol{f}(\boldsymbol{u}) = \boldsymbol{0}.$$

From the material derivative, we have

$$\boldsymbol{v}(\boldsymbol{x},t) = \frac{d}{dt}\boldsymbol{u}(\boldsymbol{x},t)$$

where $\frac{d}{dt}$ denotes the total derivative. Using the chain rule formula, we have

$$\boldsymbol{v}(\boldsymbol{x},t) = \nabla\boldsymbol{u}(\boldsymbol{x},t)\frac{\partial \boldsymbol{u}}{\partial t} + \frac{\partial \boldsymbol{u}(\boldsymbol{x},t)}{\partial t}.$$

Thus,

$$\boldsymbol{v}(\boldsymbol{x},t) = \nabla\boldsymbol{u}(\boldsymbol{x},t)\boldsymbol{v}(\boldsymbol{x},t) + \frac{\partial \boldsymbol{u}(\boldsymbol{x},t)}{\partial t}. \tag{3.12}$$

In comparison with linear elastic registration, we have an additional Euler step for the fluid model where we have to solve for $\boldsymbol{u}$ using equation (3.12). The main advantage of fluid flow registration is its ability to recover large deformation. Other regularisation terms such as diffusion and linear curvature can exploit this advantage to recover large deformation. In [67], the authors showed that diffusion registration to a fluid type formulation is able to improve the original diffusion model.

3.4.5 Demon Registration

Thirion [88], introduced the so-called demon registration where every pixel in the image acts as the image entities or demons that exert a pulling or pushing action in a similar way to that which Maxwell used for solving the Gibbs paradox in thermodynamics. Demon registration is based on the pixel velocities caused by edge based forces which are inspired from the optical flow equations. However, during this time, the theoretical basis underlying the demon algorithm is not fully understood. Thus, the method is somehow based on an ad hoc idea in image registration. Later, several authors [74, 53, 93, 95, 11, 67], provided a strong justification and a better understanding of the demon algorithm. The displacement field $\boldsymbol{u}(\boldsymbol{x})$ for demon registration can be written as

$$\boldsymbol{u}(\boldsymbol{x}) = \frac{T(\boldsymbol{x}+\boldsymbol{u}(\boldsymbol{x})) - R(\boldsymbol{x})}{\|\nabla R(\boldsymbol{x})\|_{\mathbb{R}^2}^2 + (T(\boldsymbol{x}+\boldsymbol{u}(\boldsymbol{x})) - R(\boldsymbol{x}))^2}\nabla R(\boldsymbol{x}). \tag{3.13}$$

In [67], the author provided the variational formulation for the demon algorithm. To adjust the force strength, the normalisation factor γ is proposed in [9] as follows:

$$\boldsymbol{u}(\boldsymbol{x}) = \frac{T(\boldsymbol{x}+\boldsymbol{u}(\boldsymbol{x})) - R(\boldsymbol{x})}{\|\nabla R(\boldsymbol{x})\|_{\mathbb{R}^2}^2 + \gamma(T(\boldsymbol{x}+\boldsymbol{u}(\boldsymbol{x})) - R(\boldsymbol{x}))^2}\nabla R(\boldsymbol{x}).$$

As we can see from equation (3.13), the internal forces come from the edges of the reference image R. To increase the speed of convergence, the image forces from the transformed template $T(\boldsymbol{x}+\boldsymbol{u}(\boldsymbol{x}))$ are added as proposed in [100] as follows:

$$\begin{aligned}\boldsymbol{u}(\boldsymbol{x}) = & \frac{T(\boldsymbol{x}+\boldsymbol{u}(\boldsymbol{x})) - R(\boldsymbol{x})}{\|\nabla R(\boldsymbol{x})\|_{\mathbb{R}^2}^2 + \gamma(T(\boldsymbol{x}+\boldsymbol{u}(\boldsymbol{x})) - R(\boldsymbol{x}))^2}\nabla T(\boldsymbol{x}+\boldsymbol{u}(\boldsymbol{x})) \\ & + \frac{T(\boldsymbol{x}+\boldsymbol{u}(\boldsymbol{x})) - R(\boldsymbol{x})}{\|\nabla R(\boldsymbol{x})\|_{\mathbb{R}^2}^2 + \gamma(T(\boldsymbol{x}+\boldsymbol{u}(\boldsymbol{x})) - R(\boldsymbol{x}))^2}\nabla R(\boldsymbol{x}).\end{aligned}$$

3.4.6 Diffusion Image Registration

Diffusion image registration is the simplest choice of $\mathcal{S}$ which is based on the L^2 norm of the gradient of the displacement field $\boldsymbol{u}(\boldsymbol{x})$. The regularisation term for diffusion image registration is given by [23, 88, 74]:

$$\mathcal{S}^{\text{diff}}(\boldsymbol{u}) = \frac{1}{2}\int_\Omega |\nabla u_1|^2 + |\nabla u_2|^2 \ \mathrm{d}\Omega.$$

The Euler-Lagrange equation for (3.11) with S^{diff} as the regularisation term is given by:

$$-\gamma\Delta\boldsymbol{u} + \boldsymbol{f}(\boldsymbol{u}) = \boldsymbol{0}$$

with Neumann boundary conditions $\nabla u_l \cdot \boldsymbol{n} = 0,\ l = 1, 2$ where $\boldsymbol{n}$ denotes the outward unit normal vector on the boundary $\partial\Omega$. The model can be solved efficiently using the additive operator splitting (AOS) method [67]. The model can be seen as a special case of the linear elastic model when $\mu = 1$ and $\lambda = -1$. It is well known as the classical method of Horn and Schunck [48] for optical flow computation.

3.4.7 Total Variation Image Registration

Total variation image registration is based on the TV semi-norm of $\nabla u_l, l = 1, 2$. The model [47, 78, 28] is given by

$$\mathcal{S}^{\text{TV}}(\boldsymbol{u}) = \sum_{l=1}^{2}\int_\Omega \sqrt{u_{l_{x_1}}^2 + u_{l_{x_2}}^2 + \beta}\ \mathrm{d}\Omega.$$

The Euler-Lagrange equation for (3.11) with $\mathcal{S}^{\mathrm{TV}}$ as the regularisation term is:

$$-\gamma \nabla \cdot \frac{\nabla u_l}{|\nabla u_l|_\beta} + f_l(\boldsymbol{u}) = 0,\ l = 1, 2,$$

with Neumann boundary conditions $\nabla u_l \cdot \boldsymbol{n} = 0,\ l = 1, 2$ and where β is a real number used to avoid division by zero. The model is useful to preserve discontinuities of the deformation field via preserving the piecewise constant smoothness of the deformation field. However, the model is not suitable for registration problems where smoothness is the main concern.

Elastic, diffusion and total variation regularisation are first order based models which penalise rigid displacement. Therefore, affine or rigid pre-registration needs to be included in the pre-registration step before applying the respective image registration models. Fischer and Modersitzki in [25, 26, 67, 24] proposed a second order regularisation term to overcome this problem. It was later refined by Henn and Witsch in [38, 39, 40] and Chumchob, Chen and Brito in [19, 18].

3.4.8 Fischer and Modersitzki's Linear Curvature

The first second order regularisation term [25, 26, 67, 24] for image registration is

$$\mathcal{S}^{\mathrm{FMC}}(\boldsymbol{u}) = \int_\Omega \Big[(\Delta u_1)^2 + (\Delta u_2)^2 \Big] \,\mathrm{d}\Omega. \tag{3.14}$$

This term is an approximation of the surface curvature $\kappa(u_l)$ where

$$\kappa(u_l) = \nabla \cdot \frac{\nabla u_l}{\sqrt{|\nabla u_l|^2 + 1}}.$$

When $|\nabla u_l| \approx 0$, we have $\kappa(u_l) = \Delta u_l$. The Euler-Lagrange equation for (3.11) with $\mathcal{S}^{\mathrm{FMC}}$ as the regularisation term is given by:

$$\gamma \Delta^2 \boldsymbol{u} + \boldsymbol{f}(\boldsymbol{u}) = \boldsymbol{0} \tag{3.15}$$

with boundary conditions $\Delta u_l = 0, \nabla \Delta u_l \cdot \boldsymbol{n} = 0,\ l = 1, 2$. The boundary conditions are replaced by the boundary conditions $\nabla u_l \cdot \boldsymbol{n} = \nabla \Delta u_l \cdot \boldsymbol{n} = 0,\ l = 1, 2$ for ease of implementation. Since the model requires smoothness in terms of second order derivatives, the model leads to smoother deformation compared to the models of first order. In addition, it does not require affine linear pre-registration since the affine kernel is included in the model.

3.4.9 Henn and Witsch's Curvature

The $\mathcal{S}^{\mathrm{FMC}}$ model was later refined by Henn and Witsch [38, 39, 40] where they proposed:

$$\mathcal{S}^{\mathrm{HWC}}(\boldsymbol{u}) = \frac{1}{2} \sum_{l=1}^{2} \int_\Omega \Big[(\Delta u_l)^2 - 2(u_{l_{x_1 x_1}} u_{l_{x_2 x_2}} - u^2_{l_{x_1 x_2}}) \Big] \,\mathrm{d}\Omega. \tag{3.16}$$

The minimisation problem (3.11) with $\mathcal{S}^{\text{HWC}}$ as the regularisation term leads to the same Euler-Lagrange equation (6.14) subject to the boundary conditions

$$B_1(u_l) = \frac{\partial}{\partial \boldsymbol{n}} \Delta u_l + \frac{\partial}{\partial \boldsymbol{s}} K(u_l) = 0, \qquad B_2(u_l) = \frac{\partial^2 u_l}{\partial \boldsymbol{n}^2} = 0$$

where $K(u_l) = \frac{\partial^2 u_l}{\partial x_1 \partial x_2}(n_1^2 - n_2^2)$, and $\boldsymbol{s}$ is the tangential direction of the unit vector of the outward normal vector $\boldsymbol{n} = (n_1, n_2)$. The model is based on an approximation to the sum of the squares of principle curvatures $\kappa_1(u_l)$ and $\kappa_2(u_l)$. We have,

$$\begin{aligned}\kappa_1^2(u_l) + \kappa_2^2(u_l) &= (\kappa_1(u_l) + \kappa_2(u_l))^2 - 2\kappa_1^2(u_l)\kappa_2^2(u_l) \\ &= \kappa_{\text{MC}}^2(u_l) - 2\kappa_{\text{GC}}(u_l) \\ &= \left(\nabla \cdot \left(\frac{\nabla u_l}{\sqrt{1 + |\nabla u_l|^2}}\right)\right)^2 - 2\left(\frac{u_{l_{x_1x_1}} u_{l_{x_2x_2}} - u_{l_{x_1x_2}}^2}{(1 + |\nabla u_l|^2)^2}\right)\end{aligned} \tag{3.17}$$

where κ_{MC} and κ_{GC} are the mean and Gaussian curvatures for surface $u_l(\boldsymbol{x})$. When $\nabla u_l \approx 0$, equation (3.17) can be written as

$$\kappa_1^2(u_l) + \kappa_2^2(u_l) \approx (\Delta u_l)^2 - 2\left(u_{l_{x_1x_1}} u_{l_{x_2x_2}} - u_{l_{x_1x_2}}^2\right).$$

3.4.10 Mean Curvature

The previous two second order regularisation terms use an approximation to the surface curvature. Chumchob, Chen and Brito in [19, 18] proposed a full curvature model using:

$$\mathcal{S}^{\text{MC}}(\boldsymbol{u}) = \int_\Omega \Big[\mathbb{k}(\kappa(u_1)) + \mathbb{k}(\kappa(u_2))\Big] \,\mathrm{d}\Omega$$

where $\mathbb{k}(s) = \frac{1}{2}s^2$ and

$$\kappa(u_l) = \nabla \cdot \frac{\nabla u_l}{|\nabla u_l|_\beta} = \nabla \cdot \frac{\nabla u_l}{\sqrt{|\nabla u_l|^2 + \beta}}.$$

The Euler-Lagrange equation for (3.11) with $\mathcal{S}^{\text{MC}}$ as the regularisation term is given by:

$$\gamma \nabla \cdot \Big(\frac{1}{|\nabla u_l|_\beta} \nabla \mathbb{k}'(\kappa(u_l)) - \frac{\nabla u_l \cdot \nabla \mathbb{k}'(\kappa(u_l))}{(|\nabla u_l|_\beta)^3} \nabla u_l\Big) + f_l(\boldsymbol{u}) = 0,\ l = 1, 2$$

with boundary conditions $\nabla u_l \cdot \boldsymbol{n} = \nabla \kappa(u_l) \cdot \boldsymbol{n} = 0,\ l = 1, 2.$

3.5 General Solution Schemes

There are two ways to solve minimisation problem (3.1) or (3.5). Either we use the so-called discretise then optimise approach or optimise then discretise. For the first method, the minimisation problem is discretised in the discretisation domain and then the problem is solved using any optimisation technique such as steepest descent or the Newton method. For the second method, the minimisation problem is solved using the

Euler-Lagrange equations which are computed in the continuous domain. The discrete version of the Euler-Lagrange equations are solved using a method of our choice such as parabolic or elliptic schemes.

Image registration problem in (3.1) is solve numerically using finite difference method with a uniform mesh. However, it is possible to use finite element or finite volume to solve the problem with a non-uniform mesh. The resulting transformation is highly dependent on the discretisation of the minimisation problem in (3.1) or the resulting Euler-Lagrange equations. For a uniform mesh, the grid spacing in x_1 direction, h_1 is equal to the grid spacing in x_2, h_2; i.e. $h_1 = h_2$ and will be denoted as h. Larger h, for example $h = 4$ will produce a rough alignment between the reference and template images because only a few points involves. For a smaller h, for example $h = 1$, we get a better alignment because we are able to capture locally fine details. However, for a very small h, for example $h = 0.5$, the computational cost increases, making the model less robust. In this thesis, we take $h = 1$ that balances between robustness and accuracy of the problem in (3.1).

3.5.1 Discretise then Optimise

Consider the discrete version of equation (3.1)

$$\min_{\boldsymbol{\varphi}} \mathcal{J}^h(T, R, \boldsymbol{\varphi}) = \mathcal{D}^h(T, R, \boldsymbol{\varphi}) + \gamma \mathcal{S}^h(\boldsymbol{\varphi}).$$

The Taylor expansion of $\mathcal{J}^h$ around the current approximation $\boldsymbol{\varphi}^{(k)}, k \in \mathbb{N}$ is given by

$$\begin{aligned}\mathcal{J}^h(T, R, \boldsymbol{\varphi}^{(k)} + \delta\boldsymbol{\varphi}^{(k)}) = \mathcal{J}^h(T, R, \boldsymbol{\varphi}^{(k)}) + \boldsymbol{J}(\boldsymbol{\varphi}^{(k)})\delta\boldsymbol{\varphi}^{(k)} \\ + \frac{1}{2}(\delta\boldsymbol{\varphi}^{(k)})^T \boldsymbol{H}(\boldsymbol{\varphi})\delta\boldsymbol{\varphi}^{(k)}\end{aligned}$$

where $\boldsymbol{J}$ and $\boldsymbol{H}$ are the Jacobian and Hessian of $\mathcal{J}^h(T, R, \boldsymbol{\varphi})$ at $\boldsymbol{\varphi}^{(k)}$ respectively. The updated value for $\boldsymbol{\varphi}^{(k+1)}$ is given by

$$\boldsymbol{\varphi}^{(k+1)} = \boldsymbol{\varphi}^{(k)} + \alpha^{(k)}\delta\boldsymbol{\varphi}^{(k)}$$

where $\alpha^{(k)}$ is the line search parameter used to guarantee the reduction of $\mathcal{J}$. For the Newton type method, the perturbation $\delta\boldsymbol{\varphi}^{(k)}$ is determined by solving the following normal equation

$$\boldsymbol{H}(\boldsymbol{\varphi}^{(k)})\delta\boldsymbol{\varphi}^{(k)} = -\boldsymbol{J}(\boldsymbol{\varphi}^{(k)}).$$

3.5.2 Optimise then Discretise

The main idea is to solve the Euler-Lagrange equation

$$\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\varphi}) + \gamma \boldsymbol{\mathcal{A}}(\boldsymbol{\varphi})(\boldsymbol{x}) = \boldsymbol{0} \text{ for all } \boldsymbol{x} \in \Omega \tag{3.18}$$

with the appropriate boundary conditions. As mentioned before, the Euler-Lagrange equation is the necessary condition for the minimiser of the joint functional $\mathcal{J}$ in equation (3.1). The force term $\boldsymbol{f}$ is the Gateaux derivative for the similarity measure $\mathcal{D}(T, R, \boldsymbol{\varphi})$ and $\boldsymbol{\mathcal{A}}(\boldsymbol{\varphi})$ is related to the regularisation term $\mathcal{S}(\boldsymbol{\varphi})$. One of the methods of solving equation (3.18) is to exploit a fixed point iteration or elliptic scheme. Starting with an initial guess $\boldsymbol{\varphi}^{(0)}$, we define $\boldsymbol{\varphi}^{(k+1)}$ implicitly by

$$\boldsymbol{\mathcal{A}}(\boldsymbol{\varphi}^{(k+1)})(\boldsymbol{x}) = -\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\varphi}^{(k)}),$$

where $k \in \mathbb{N}$. A parabolic approach for solving equation (3.1) is done by introducing an artificial time variable t and determining the steady state solution of the following scheme

$$\frac{\boldsymbol{\varphi}^{(k+1)} - \boldsymbol{\varphi}^{(k)}}{\tau} + \boldsymbol{\mathcal{A}}(\boldsymbol{\varphi}^{(k+1)})(\boldsymbol{x}) = -\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\varphi}^{(k)}). \tag{3.19}$$

One of the drawbacks of using the time variable t is that the time step τ should be small enough to guarantee the convergence of the iterative scheme. Based on the Courant-Friedrichs-Lewy (CFL) condition, the time step should satisfy

$$\tau < \mathcal{O}\left(\left(\frac{1}{h}\right)^4\right)$$

where h is the step size in the discretisation of the differential operator using the finite difference method. An additive operator splitting scheme (AOS) [67, 101] is faster and more efficient than the implicit scheme (3.19). The basic idea is to replace the inverse of the sum by a sum of inverses. We replace equation (3.19) by

$$\boldsymbol{\varphi}^{(k+1)} = \frac{1}{2}\sum_{l=1}^{2} \left(\boldsymbol{I} - 2\tau\boldsymbol{\mathcal{A}}_l\right)^{-1} \left(\boldsymbol{\varphi}^{(k)} - \tau\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\varphi}^{(k)})\right)$$

where $\boldsymbol{\mathcal{A}}_l$ denotes the corresponding coefficient matrix for $\boldsymbol{\mathcal{A}}$ in the x_l direction.

3.6 Interpolation Methods

Image registration aims to find an optimal transformation so that the template image becomes similar to the given reference image. The reference image is defined before discretisation of the problem. During the transformation of the template image $T(\boldsymbol{\varphi})$, points may fall outside of the discretisation points. Thus, interpolation methods are required in order to assign the intensity values. Nearest neighbour is the simplest method of interpolation followed by linear interpolation. Bilinear interpolation gives a jagged effect to the image. On the other hand, cubic B-splines give better results.

3.6.1 Nearest Neighbour Interpolation

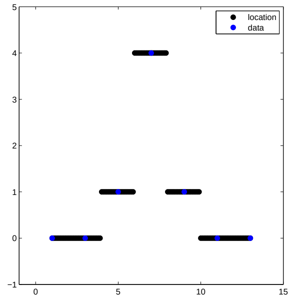


Figure 3.3: Illustration of nearest neighbour interpolation in a 1D problem.

Referring to Figure 3.3, we are given the values of the data points in blue where $x = \{1, 3, 5, 7, 9, 11, 13\}$ and $y = \{0, 0, 1, 4, 1, 0, 0\}$ respectively. Based on nearest neighbour interpolation, the values for y at non-integer points x in $[1, 13]$ are the values of the nearest integer x.

3.6.2 Linear Interpolation

Interpolation problems arise whenever one tries to construct a function $p(x)$ from a given function $f(x)$ which approximates $f(x)$. One of the solutions is to use polynomial interpolation. A polynomial of order n is a function of the form

$$p(x) = a_1 + a_2 x + \ldots + a_n x^{n-1} = \sum_{j=1}^{n} a_j x^{j-1}.$$

For linear interpolation, we have $n = 1$ and

$$p(x) = a_1 + a_2 x.$$

For any two given points (x_0, y_0) and (x_1, y_1) we have

$$a_1 + a_2 x = y_0 + (y_1 - y_0)\frac{x - x_0}{x_1 - x_0}$$

which approximates the values of $p(x) = y$ for any x between x_0 and x_1 as illustrated in Figure 3.4.

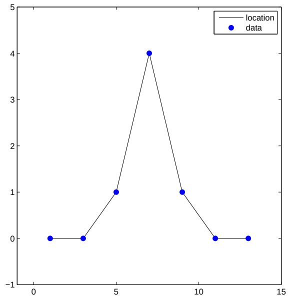


Figure 3.4: Illustration of linear interpolation in a 1D problem.

3.6.3 Polynomial Interpolation

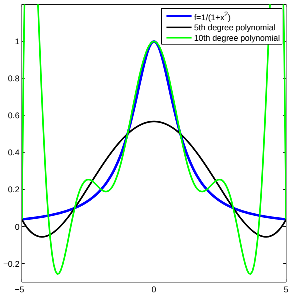


Figure 3.5: Illustration of Runge phenomenon using higher order polynomial interpolation.

Higher order polynomial interpolation can give us a smooth and higher order continuous function at the edges of the interval, but it will suffer from Runge's phenomenon where oscillation occurs at the edges. This problem highlights that going to a higher degree of polynomial interpolation does not always improve accuracy. To illustrate Runge's phenomenon [83, 89], consider $f(x) = \frac{1}{1+x^2}$ in the interval $[-5, 5]$ and interpolate using polynomials of degrees five and ten as shown in Figure 3.5. To ease the computation of polynomial interpolation, one goes for a low order piecewise polynomial interpolation such as cubic. We say $p(x)$ is a piecewise polynomial of order n if the degree of $p(x)$ is less than n on each subintervals. For a given knot sequence, a n-th degree spline function is a piecewise polynomial of degree n that is $(n-1)$ continuous at each knot.

3.6.4 Spline Interpolation

To illustrate how spline interpolation works in function approximation, consider that we want to construct the piecewise cubic polynomial interpolation as follows:

$$S(x) = \begin{cases} s_1(x), & \text{if } x_1 \leq x < x_2; \\ s_2(x), & \text{if } x_2 \leq x < x_3; \\ \vdots & \vdots \\ s_{n-1}(x), & \text{if } x_{n-1} \leq x < x_n. \end{cases} \tag{3.20}$$

We have to find

$$s_i(x) = a_i(x - x_i)^3 + b_i(x - x_i)^2 + c_i(x - x_i) + d_i, \quad i = 1, 2, \ldots, n - 1$$

using the following properties:

1. $S(x)$ will be continuous on the interval $[x_1, x_n]$.
2. $S'(x)$ and $S''(x)$ will be continuous on the interval $[x_1, x_n]$.

A similar idea can be extended to 2D using the tensor product but it is easier if we use basis functions or B-splines. B-splines are built recursively from constant B-splines. Though we are interpolating at $n + 1$ knots $t_0, \ldots, t_n$, in order to derive B-splines we need extra nodes outside $[t_0, t_n]$ to build the basis. Thus we add extra knots on either end as follows

$$\ldots < t_{-2} < t_{-1} < t_0, \ldots < t_n < t_{n+1} < \ldots. \tag{3.21}$$

Given these knots, we can define the constant B-spline

$$B_{j,1} = \begin{cases} 1, & \text{if } t_j \leq x < t_{j+1}; \\ 0, & \text{otherwise.} \end{cases}$$

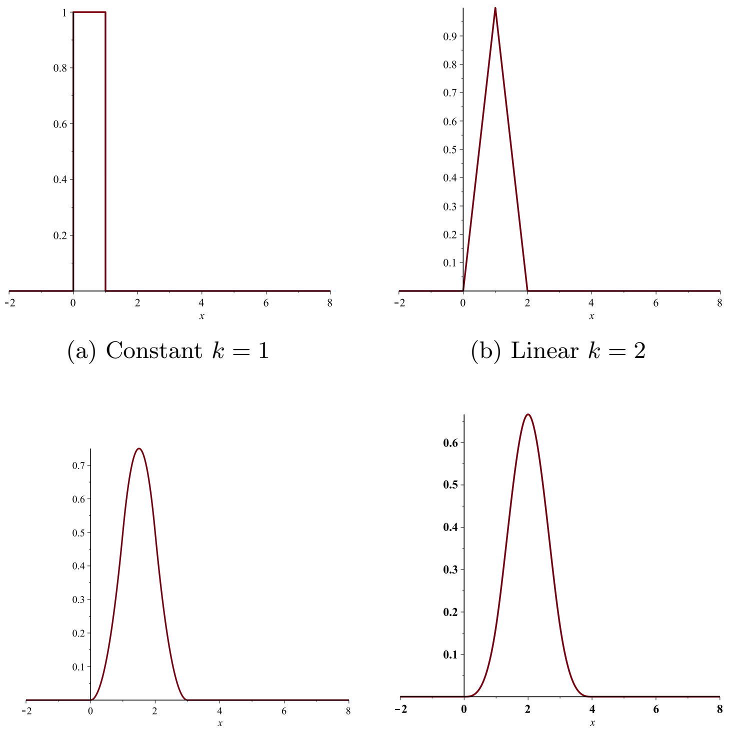
For $k > 1$,

$$B_{j,k} = \omega_{j,k} B_{j,k-1} + (1 - \omega_{j+1,k}) B_{j+1,k-1}$$

where

$$\omega_{j,k}(x) = \frac{x - t_j}{t_{j+k+1} - t_j}.$$

The basis functions for $k = 1, 2, 3, 4$ are given in Figure 3.6.



(a) Constant $k = 1$

(b) Linear $k = 2$

(c) Quadratic $k = 3$

(d) Cubic $k = 4$

Figure 3.6: Illustration of basis functions for $k = 1, 2, 3, 4$.

The interpolation conditions give $n+1$ equations that constrain the unknown coefficient $c_{j,k}$ in the expansion of S_k where

$$S_k(t_j) = \sum_{j=-\infty}^{j=\infty} c_{j,k} B_{j,k}(t_j).$$

The compact support of the B-splines immediately suggests that we set most of the $c_{j,k}$ coefficients to zero, giving S_k as a finite sum. For $k > 1$, we are left with $n + k - 1$ nontrivial $c_{j,k}$ variables to be determined from $n + 1$ interpolation conditions. The coefficients $c_{j,1}$ in the expansion of

$$S_1(x) = \sum_{j=-\infty}^{j=\infty} c_{j,1} B_{j,1}(x)$$

are completely determined by the interpolation requirement

$$S_1(t_j) = f(t_j), \text{ for } \ j = 0, 1, \ldots, n.$$

Therefore we have

$$S_1(x) = \sum_{j=0}^{n} f_j B_{j,1}(x).$$

The basis function can be used easily in higher order interpolation using the tensor product.

Definition 3.6.1 ***Direct Matrix Product (Kronecker Product).*** *Given two matrices $A \in \mathbb{R}^{n\times m}$ and $B \in \mathbb{R}^{p\times q}$, the elements of the direct matrix product $C = A\otimes B \in \mathbb{R}^{mp\times nq}$ are defined by*

$$c_{\alpha,\beta} = a_{i,j}b_{k,l} \quad \textit{where} \quad \alpha = p(i-1)+k, \beta = q(j-1)+1.$$

Alternatively, we can write

$$A\otimes B = \begin{pmatrix} a_{1,1}B & a_{1,2}B & \dots & a_{1,n}B \\ a_{2,1}B & a_{2,2}B & \dots & a_{2,n}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1}B & a_{m,2}B & \dots & a_{m,n}B \end{pmatrix}.$$

For example, if we wish to evaluate the spline w at the point (a,b) using

$$B_{i,h,s}(x)B_{j,k,t}(y),$$

this can be accomplished by factoring out appropriately,

$$w(a,b) = \sum_i \Big(\sum_j c_{i,j}B_{j,k,t}(b)\Big)B_{i,h,s}(a) = \sum_j \Big(\sum_i c_{i,j}B_{i,h,s}(a)\Big)B_{j,k,t}(b).$$

For a given knot sequence $(\sigma_i,\tau_j), i = 1,\dots,m, j = 1,\dots,n$, there is exactly one spline function which agrees with $w(\sigma_i,\tau_j)$ of the given rectangular mesh. This interpolation can be written in the following form

$$w(x,y) = \sum_{i,j} c_{i,j}B_{i,h,s}(x)\otimes B_{j,k,t}(y)$$

where

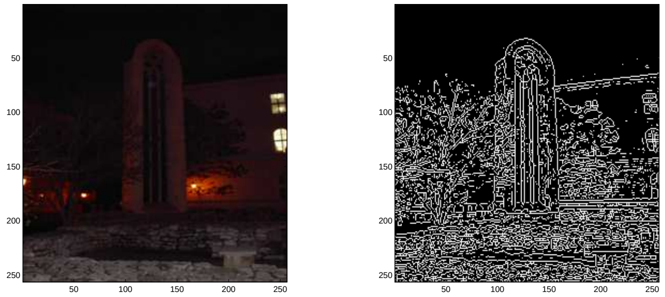
$$c_{i,j} = \Big(B_{i,h,s}(\tau_j)\Big)^{-1}((w(\sigma_i,\tau_j)))\Big(B_{j,k,t}(\tau_j)\Big)^{-1}.$$

3.7 Image Segmentation

Image segmentation is the process of dividing or grouping pixels of an image into regions or categories. Each of the regions belongs to a different object or part of the image which has the same characteristic based on intensity or texture. Every pixel in the image is allocated to one of these categories for further processing and analysis. For example, in a two phase image segmentation model, images are separated into foreground and background to distinguish homogeneous objects. Image segmentation has broad applications in medical image processing. As such, radiologists use image segmentation to quantify size, location and region of the cancerous tissues.

In this section, a brief review of the Mumford-Shah [70] and Chan-Vese [12] models for image segmentation will be given. These models will be useful to know for Chapter

7 where we combine the task of registration and segmentation into one framework. For an illustration of image segmentation we refer to Figure 3.7.



(a) Given Image (b) Binary Representation

Figure 3.7: Illustration of an image segmentation problem. (a) is the image to be segmented because the image appears dark and the boundaries of the objects are not clearly visible. (b) shows the binary representation of the image in (a) where white pixels represent the edges of the object in (a).

3.7.1 Mumford-Shah Segmentation

Image segmentation is an ill-posed problem in the sense of Hadamard. In large number of images, a unique solution to the segmentation problem does not exist. According to [3], the solution of image segmentation problem is not unique because it is possible to find different region or partition of an image at different level of analysis or detail. Hence, the segmentation results depend on the image itself and on the particular application.

There are two main approaches in image segmentation: the non-variational approach and the variational (energy based) approach. We are interested in the variational approach where the problem is formulated as an energy minimisation problem. This approach is further classified into edge and region based models. In the edge based model, the edge information is used to guide the active contour towards the object boundary. Meanwhile, the region based models use image intensities to guide the motion of the active contours.

Let Ω be a bounded domain in $\mathbb{R}^n$ and $z(\boldsymbol{x})$ be a bounded measurable function defined on Ω where z represents the given image. The Mumford and Shah [70] functional is described by a pair (z, Γ) where Γ is a closed subset and z is a function belonging to $\mathcal{C}^1(\Omega \backslash \Gamma)$. The authors of [70] tried to approximate the given image z, by a piecewise smooth function where they represent the problem as

$$\begin{aligned}\min_{\Gamma, c_1, c_2} \Big\{ \mathcal{J}^{\mathrm{MS}}(\Gamma(\boldsymbol{x}), c_1, c_2) = \mu \mathrm{length}(\Gamma(\boldsymbol{x})) + \lambda_1 \int_{\mathrm{inside}(\Gamma)} |z(\boldsymbol{x}) - c_1|^2 \, \mathrm{d}\boldsymbol{x} + \\ \lambda_2 \int_{\mathrm{outside}(\Gamma)} |z(\boldsymbol{x}) - c_2|^2 \, \mathrm{d}\boldsymbol{x} \Big\}\end{aligned} \tag{3.22}$$

where Γ is the edge set, c_1 is the average of the intensities of the pixels inside the shape defined by the contour Γ and c_2 is the average of the intensities of the pixels outside the shape defined by Γ. $\mu, \lambda_1, \lambda_2$ are positive parameters where the first term in (3.22) aims to minimise the length of the contour Γ so that the boundaries are as short and smooth as possible. The second and third terms in (3.22) are the fitting terms to derive c_1 and c_2 close to the average intensities of the inside and outside respectively of the shape defined by Γ. Practically, the functional in (3.22) is not convex and difficult to minimise. Chan and Vese [12] solved the functional in (3.22) using level set representation where the image is modelled as a piecewise constant function.

3.7.2 Chan-Vese Segmentation Model

Chan and Vese [12] used the level set method [73] to solve the image segmentation problem. The given image z is defined on a two dimensional domain $\Omega \in \mathbb{R}^n$. The unknown curve Γ can be represented by the zero level set of a Lipschitz function [73] $\phi : \mathbb{R}^2 \to \mathbb{R}$, such that

$$\begin{cases} \Gamma = \partial\Omega_1 = \{(x, y) \in \Omega | \phi(x, y) = 0\}, \\ \text{inside}(\Gamma) = \Omega_1 = \{(x, y) \in \Omega | \phi(x, y) > 0\}, \\ \text{outside}(\Gamma) = \Omega_2 = \{(x, y) \in \Omega | \phi(x, y) < 0\}. \end{cases}$$

Using the zero level set to represent Γ and introducing the Heaviside step function, $H(x)$, the problem in (3.22) is reformulated as

$$\min_{\phi, c_1, c_2} \Big\{ \mathcal{J}^{\text{CV}}(\phi(\boldsymbol{x}, c_1, c_2)) = \mu \int_\Omega \delta(\phi(\boldsymbol{x})) |\nabla H(\phi(\boldsymbol{x}))| \, \mathrm{d}\boldsymbol{x} + \lambda_1 \int_\Omega |z(\boldsymbol{x}) - c_1|^2 H(\phi(\boldsymbol{x})) \, \mathrm{d}\boldsymbol{x} \\ + \lambda_2 \int_\Omega |z(\boldsymbol{x}) - c_2|^2 (1 - H(\phi(\boldsymbol{x}))) \, \mathrm{d}\boldsymbol{x} \Big\}. \tag{3.23}$$

Once the level set function ϕ is obtained, the segmented image is given by

$$u = c_1 H(\phi(\boldsymbol{x})) + c_2(1 - H(\phi(\boldsymbol{x}))).$$

The proof for the existence of a minimiser of (3.23) can be found in [12]. Note that the contour Γ is given by the non-zero elements of the delta function $\delta(\phi)$ evaluated over ϕ which is assumed to be positive inside Γ (corresponding to $H(\phi) = 1$) and negative outside Γ (corresponding to $H(\phi) = 0$).

The minimiser of (3.23) with respect to c_1 and c_2 where $\phi(x, y)$ is fixed is given by

$$c_1(\phi(\boldsymbol{x})) = \frac{\int_\Omega z(\boldsymbol{x}) H(\phi(\boldsymbol{x})) \, \mathrm{d}\boldsymbol{x}}{\int_\Omega H(\phi(\boldsymbol{x})) \, \mathrm{d}\boldsymbol{x}}$$

and

$$c_2(\phi(\boldsymbol{x})) = \frac{\int_\Omega z(\boldsymbol{x})(1 - H(\phi(\boldsymbol{x}))) \, \mathrm{d}\boldsymbol{x}}{\int_\Omega 1 - H(\phi(\boldsymbol{x})) \, \mathrm{d}\boldsymbol{x}}.$$

To compute the Euler-Lagrange equation for the unknown function ϕ, they replace

$H(x)$ and $\delta(x)$ with analytic approximations $H_\varepsilon(x)$ and $\frac{dH_\varepsilon(x)}{dx} = \delta_\varepsilon(x)$ where

$$H_\varepsilon(x) = \frac{1}{2}\Big(1 + \frac{2}{\pi}\arctan\left(\frac{x}{\varepsilon}\right)\Big), \quad \delta_\varepsilon(x) = \frac{\varepsilon}{\pi(\varepsilon^2 + x^2)}.$$

The Euler-Lagrange equation for ϕ is given by

$$\begin{cases} \delta_\varepsilon(\phi)\left[\mu\nabla\cdot\left(\frac{\nabla\phi}{|\nabla\phi|} - \lambda_1(z-c_1)^2 + \lambda_2(z-c_2)^2\right)\right] = 0, \text{ in } \Omega \\ \frac{\delta_\varepsilon(\phi)}{|\nabla\phi|}\frac{\partial\phi}{\partial\boldsymbol{n}} = 0, \text{ or } \frac{\partial\phi}{\partial\boldsymbol{n}} = 0 \text{ on } \partial\Omega. \end{cases} \tag{3.24}$$

The solution of equation (3.24) can be found via introducing an artificial time step t as considered by the authors of [12].

Chapter 4

A Decomposition Model Combining Parametric and Non-parametric Deformation

We have considered some relevant and useful tools for mathematical preliminaries in Chapter 2. At the same time, we have reviewed mathematical models for image registration and image segmentation in Chapter 3. In this chapter, we propose a decomposition model combining both parametric and non-parametric deformations that have been discussed in the previous chapter. The proposed decomposition model possesses advantages of the two categories of models. Parametric models are relatively faster than non-parametric models, meanwhile non-parametric models are well known to effectively match local differences. We propose a decomposition model where the overall deformation consists of both global and pixel level displacement for effective image registration. The resulting model is robust and fast in comparison with individual models. We present in Section 4.2 a parametric model based on cubic B-spline. We next present the linear curvature model in Section 4.3 and the solution scheme for the model. We present our proposed decomposition model in Section 4.4 and discuss how to choose the regularisation parameters in Section 4.5. Finally we present the numerical results that show the advantages of the proposed model in Section 4.6 before we conclude the chapter in Section 4.7.

4.1 Introduction

Image registration models can be categorised into either parametric or non-parametric categories. The first category involves only a few parameters; for example, in rigid registration there are only three parameters that describe global translation and rotation for 2D images. However, non-rigid registration models are expected to give better results when input images undergo different scaling and consist of local deformations. Even today, in clinical applications, one of the most commonly applied models is still affine (which has six parameters [16]) because the automated solution has been proven to be accurate and has reached a degree of maturity in comparison with non-rigid image registration [81]. Here ‘affine’ means a linear transform. For more sophisticated parametric transforms such as cubic splines [82] to be reviewed shortly, the number of parameters can be larger e.g. 64 parameters. Nevertheless a nonlinear least squares system with fewer than 1000 unknowns is not a computational challenge; consequently all parametric registration methods are efficient, though registration accuracy depends on the input images. Landmark based registration methods also belong to the first category since parametric transforms are used to match the landmarks [42].

The second category of methods originates from modelling the transformation as a physical displacement so ideas of physical processes such as diffusion [88], elastic deformation [6] and curvature motion [24] are used for registration. Although treating a transformation as a completely unknown function provides flexibility because the search space is much larger, the process of finding a displacement for every pixel (subject to an overall smoothness control i.e. regularisation) is very expensive in terms of computational cost. Hence a fast algorithms are a major issue in the second category of methods.

Mathematically, both categories of methods involve nonlinear problems to solve. As far as the numerical solution is concerned, getting a good initial guess for a nonlinear system is of importance. Practically, there are several methods [16] to initialise an affine transform in the first category and a method from the second category often requires assistance of another method (often an affine method) from the first category by way of providing an initial solution.

We propose a decomposition model combining both the parametric and non-parametric transformations that possesses advantages of the two categories. In terms of effectiveness in accurate registration, first, alignment is carried out by the parametric part of the transformation and, second, alignment or deformation is modelled by the non-parametric part of the transformation. In terms of efficiency, the new model benefits from the fast implementation of the parametric part of the transformation and also from a good initial guess to accelerate the solution of the non-parametric part of the transformation. We shall choose one cubic spline based model for the parametric transformation and a linear curvature model for non-parametric transformation as the latter offers the advantage of allowing affine linear transformation. It is a general framework, other combinations are possible and can be studied later. We present an alternating minimisation method for solving the coupled transformation. Numerical experiments are shown to illustrate the superiority of the decomposition model over individual models.

4.2 Parametric Image Registration: Cubic B-spline

We briefly introduce the free form deformation (FFD) based cubic B-spline model [82], to be denoted by M1, because it has been shown to be capable of successfully registering cardiac [69], breast [82] and brain [55] images. The FFD model was investigated in [52] and ranked as one of the best methods among fourteen non-rigid methods. It is a parametric model and, though more complex than an affine model, still efficient. Other examples of parametric models using basis functions such as wavelet and radial spline basis functions can be developed [54]. Theoretically any transformation based on a set of parameters which are fewer than the number of pixels can be potentially explored to lead to an efficient method.

In FFD, the variational optimisation problem is

$$\begin{aligned}\min_{\varphi_1,\varphi_2}\Big\{\mathcal{J}^{\mathrm{M1}}(\varphi_1,\varphi_2) =\ & \mathcal{D}(\boldsymbol{\varphi}(\boldsymbol{x}))+\gamma\mathcal{S}(\boldsymbol{\varphi}(\boldsymbol{x}))\\ =\ & \mathcal{D}(\boldsymbol{\varphi}(\boldsymbol{x}))+\gamma\sum_{\ell=1}^{2}\mathcal{S}^{TP}(\varphi_\ell(\boldsymbol{x})) \qquad (4.1)\\ =\frac{1}{2}\int_\Omega\Big[T(\boldsymbol{\varphi}(\boldsymbol{x}))-R\Big]^2\mathrm{d}\Omega+\gamma\sum_{\ell=1}^{2}\int_\Omega\Big[&\left(\frac{\partial^2\varphi_\ell}{\partial x_1^2}\right)^2+2\left(\frac{\partial^2\varphi_\ell}{\partial x_1\partial x_2}\right)^2+\left(\frac{\partial^2\varphi_\ell}{\partial x_2^2}\right)^2\Big]\mathrm{d}\Omega\Big\}.\end{aligned}$$

This particular regularisation term is proposed in [98] to obtain a smooth B-spline interpolation function. It is zero for any affine transformations and penalises only non affine transformations [98, 82]. It remains to define the parametric representation for $\boldsymbol{\varphi}$.

First, a rectangular grid of $n_{x_1}\times n_{x_2}$ uniformly distributed grid points, called the control point grid, is set up covering the image domain Ω. We shall construct B-splines on this control point grid. Denote by $(\delta_{x_1},\delta_{x_2})$ the spacings between the nodes in the control point grid and by $(\alpha_{1,i,j},\alpha_{2,i,j})$ the coefficient centre at the $(i,j)^{\mathrm{th}}$ control point. The transformation at every position $\boldsymbol{x}$ is given as $\boldsymbol{\varphi}(\boldsymbol{x})=\boldsymbol{\varphi}(\boldsymbol{x},\boldsymbol{\alpha})=(\varphi_1(\boldsymbol{x},\alpha_1),\varphi_2(\boldsymbol{x},\alpha_2))$ where $\boldsymbol{\alpha}=(\alpha_1,\alpha_2)$ denotes the collection of all $2n_{x_1}n_{x_2}$ B-spline coefficients:

$$\alpha_1=\begin{bmatrix}\alpha_{1,1,1} & \alpha_{1,1,2} & \cdots & \alpha_{1,1,n_{x_2}}\\ \alpha_{1,2,1} & \alpha_{1,2,2} & \cdots & \alpha_{1,2,n_{x_2}}\\ \cdots & \cdots & \ddots & \cdots\\ \alpha_{1,n_{x_1},1} & \alpha_{1,n_{x_1},2} & \cdots & \alpha_{1,n_{x_1},n_{x_2}}\end{bmatrix},\quad \alpha_2=\begin{bmatrix}\alpha_{2,1,1} & \alpha_{2,1,2} & \cdots & \alpha_{2,1,n_{x_2}}\\ \alpha_{2,2,1} & \alpha_{2,2,2} & \cdots & \alpha_{2,2,n_{x_2}}\\ \cdots & \cdots & \ddots & \cdots\\ \alpha_{2,n_{x_1},1} & \alpha_{2,n_{x_1},2} & \cdots & \alpha_{2,n_{x_1},n_{x_2}}\end{bmatrix}.$$

Second, the cubic B-splines are written in the form of global basis functions $B_{i,j}$ as follows:

$$\varphi_1(\boldsymbol{x},\alpha_1)=\sum_{i=-1}^{n_{x_1}-2}\sum_{j=-1}^{n_{x_2}-2}B_{i,j}(\boldsymbol{x})\alpha_{1,i,j},\quad \varphi_2(\boldsymbol{x},\alpha_2)=\sum_{i=-1}^{n_{x_1}-2}\sum_{j=-1}^{n_{x_2}-2}B_{i,j}(\boldsymbol{x})\alpha_{2,i,j},$$

where

$$B_{i,j}(\boldsymbol{x})=\begin{cases}B_l(\mu)B_m(\nu), & i=\widetilde{i}+l, j=\widetilde{j}+m,\ \text{for } l,m=0,1,2,3;\\ 0, & \text{elsewhere,}\end{cases}$$

and $\widetilde{i} = \lfloor \frac{x_1}{\delta_{x_1}} \rfloor - 1, \widetilde{j} = \lfloor \frac{x_2}{\delta_{x_2}} \rfloor - 1, \mu = \frac{x_1}{\delta_{x_1}} - \lfloor \frac{x_1}{\delta_{x_1}} \rfloor, \nu = \frac{x_2}{\delta_{x_2}} - \lfloor \frac{x_2}{\delta_{x_2}} \rfloor$. Equivalently, the transformation is given by:

$$\varphi_1(\boldsymbol{x}, \alpha_1) = \sum_{l=0}^{3} \sum_{m=0}^{3} B_l(\mu) B_m(\nu) \alpha_{1,i+l,j+m}, \quad \varphi_2(\boldsymbol{x}, \alpha_2) = \sum_{l=0}^{3} \sum_{m=0}^{3} B_l(\mu) B_m(\nu) \alpha_{2,i+l,j+m} \tag{4.2}$$

where $i = \lfloor \frac{x_1}{\delta_{x_1}} \rfloor - 1, j = \lfloor \frac{x_2}{\delta_{x_2}} \rfloor - 1, \mu = \frac{x_1}{\delta_{x_1}} - \lfloor \frac{x_1}{\delta_{x_1}} \rfloor, \nu = \frac{x_2}{\delta_{x_2}} - \lfloor \frac{x_2}{\delta_{x_2}} \rfloor$, and $B_l(\mu)$ represents the lth basis function of the cubic B-spline where

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

Third, before we discuss how to solve (4.1) using (4.2), we assume that the reference image R and the template image T are discrete images of size $N_1 \times N_2$ (without changing the notation) on a grid of pixels $\Omega^h \subset \Omega$. Through Ω^h, we split the domain Ω into $N_1 \times N_2$ cells of size $h_1 \times h_2$. The grid points can be located at the vertex of the cell as follows:

$$\Omega^h = \{\boldsymbol{x}_{i,j} = (x_{1,i}, x_{2,j}) = (ih_1, jh_2) \mid 0 \le i \le N_1 - 1, 0 \le j \le N_2 - 1\}.$$

For example, in Test 1 later with $N_1 = N_2 = 129, (\delta_{x_1}, \delta_{x_2}) = (4, 4)$, we have $n_{x_1} \times n_{x_2} = 36 \times 36$ and there are $2n_{x_1}n_{x_2} = 2592$ parameters to determine. Since $B_{i,j}$ has a local compact support (i.e zero beyond 4×4 control points), equation (4.2) reduces to a simpler form.

Next, the discretised formulation for (4.1) is given by:

$$\begin{aligned} \min_{\alpha_1,\alpha_2} \mathscr{J}^{\mathrm{M1}}(\alpha_1, \alpha_2) = \frac{1}{2} \sum_{c=0}^{N_1-1} \sum_{d=0}^{N_2-1} (T(\boldsymbol{\varphi}(\boldsymbol{x}_{c,d}, \boldsymbol{\alpha})) - R(\boldsymbol{x}_{c,d}))^2 + \\ \gamma \sum_{\ell=1}^{2} \sum_{c=0}^{N_1-1} \sum_{d=0}^{N_2-1} (\varphi_{\ell_{x_1x_1}}(\boldsymbol{x}_{c,d}, \alpha_l))^2 + 2(\varphi_{\ell_{x_1x_2}}(\boldsymbol{x}_{c,d}, \alpha_\ell))^2 + (\varphi_{\ell_{x_2x_2}}(\boldsymbol{x}_{c,d}, \alpha_\ell))^2 \end{aligned} \tag{4.3}$$

where control points are labeled as i, j and the pixels are indexed as c, d. Further the fitting part in (4.3) can be written as

$$\begin{aligned} &[T(\boldsymbol{\varphi}(\boldsymbol{x}_{c,d}, \boldsymbol{\alpha})) - R(\boldsymbol{x}_{c,d})]^2 = \\ &\Big[T\Big(\sum_{i=-1}^{n_{x_1}-2} \sum_{j=-1}^{n_{x_2}-2} B_{i,j}(\boldsymbol{x}_{c,d}) \alpha_{1,i,j}, \sum_{i=-1}^{n_{x_1}-2} \sum_{j=-1}^{n_{x_2}-2} B_{i,j}(\boldsymbol{x}_{c,d}) \alpha_{2,i,j} \Big) - R(\boldsymbol{x}_{c,d})\Big]^2 \end{aligned}$$

where we see that the unknowns are $\alpha_{\ell,i,j}$. Here $\varphi_{\ell_{x_1x_1}} = \frac{\partial^2 \varphi_\ell}{\partial x_1^2}$ and similarly for $\varphi_{\ell_{x_1x_2}}$ and $\varphi_{\ell_{x_2x_2}}$.

The regularisation part in (4.3) can also be written as:

$$\mathcal{S}^{TP}(\varphi_\ell(\boldsymbol{x}_{c,d}, \alpha_\ell)) = \sum_{i=-1}^{n_{x_1}-2} \sum_{j=-1}^{n_{x_2}-2} \left[\frac{1}{\delta_{x_1}^4} C_{i,j}^2(\boldsymbol{x}_{c,d}) + \frac{1}{\delta_{x_1}^2 \delta_{x_2}^2} 2D_{i,j}^2(\boldsymbol{x}_{c,d}) + \frac{1}{\delta_{x_2}^4} E_{i,j}^2(\boldsymbol{x}_{c,d}) \right] (\alpha_{\ell,i,j})^2$$

where

$$C_{i,j}(\boldsymbol{x}_{c,d}) = \begin{cases} B_l''(\mu) B_m(\nu), & i = \widetilde{i} + l, j = \widetilde{j} + m, \text{ for } l, m = 0, 1, 2, 3; \\ 0, & \text{elsewhere,} \end{cases}$$

$$D_{i,j}(\boldsymbol{x}_{c,d}) = \begin{cases} B_l'(\mu) B_m'(\nu), & i = \widetilde{i} + l, j = \widetilde{j} + m, \text{ for } l, m = 0, 1, 2, 3; \\ 0, & \text{elsewhere,} \end{cases}$$

$$E_{i,j}(\boldsymbol{x}_{c,d}) = \begin{cases} B_l(\mu) B_m''(\nu), & i = \widetilde{i} + l, j = \widetilde{j} + m, \text{ for } l, m = 0, 1, 2, 3; \\ 0, & \text{elsewhere,} \end{cases}$$

where $\widetilde{i} = \lfloor \frac{x_{1,c}}{\delta_{x_1}} \rfloor - 1, \widetilde{j} = \lfloor \frac{x_{2,d}}{\delta_{x_2}} \rfloor - 1, \mu = \frac{x_{1,c}}{\delta_{x_1}} - \lfloor \frac{x_{1,c}}{\delta_{x_1}} \rfloor, \nu = \frac{x_{2,d}}{\delta_{x_2}} - \lfloor \frac{x_{2,d}}{\delta_{x_2}} \rfloor$, $B_l''(\mu) = \frac{\partial^2 B_l}{\partial x_1^2}$ and $B_l'(\mu) = \frac{\partial B_l}{\partial x_1}$.

To ease the formulation, we take $N = N_1 = N_2, \delta_x = \delta_{x_1} = \delta_{x_2}$ and the vector $\boldsymbol{\alpha}$ is given by $2n_x n_x \times 1$ where $n_x = \frac{N}{\delta_x} + 4$. The first order optimality condition is

$$\boldsymbol{G} = \nabla_{\boldsymbol{\alpha}} \mathcal{J}^{M1}(\alpha_1, \alpha_2) = \boldsymbol{0} \tag{4.4}$$

where $\boldsymbol{G} = [g_1, g_2]^T = [g_{1,1,1},\ g_{1,2,1},\ g_{1,3,1}, \cdots, g_{1,n_{x_1},n_{x_2}},\ g_{2,1,1},\ g_{2,2,1},\ g_{2,3,1}, \cdots, g_{2,n_{x_1},n_{x_2}}]^T$. Noting $\mathcal{D}(\boldsymbol{\varphi}(\boldsymbol{x}, \boldsymbol{\alpha})) = T(\boldsymbol{\varphi}(\boldsymbol{x}, \boldsymbol{\alpha})) - R(\boldsymbol{x})$,

$$\begin{aligned} g_{1,i,j} = \frac{\partial \mathcal{J}^{M1}}{\partial \alpha_{1,i,j}} &= \frac{1}{2} \frac{\partial(\mathcal{D}^2)}{\partial \alpha_{1,i,j}} + \gamma \frac{\partial \mathcal{S}_1^{TP}}{\partial \alpha_{1,i,j}} = \mathcal{D}(\boldsymbol{\varphi}(\boldsymbol{x}, \boldsymbol{\alpha})) \nabla_{\varphi_1} T(\boldsymbol{\varphi}(\boldsymbol{x}, \boldsymbol{\alpha})) B_{i,j}(\boldsymbol{x}) + \gamma \frac{\partial \mathcal{S}_1^{TP}}{\partial \alpha_{1,i,j}}, \\ g_{2,i,j} = \frac{\partial \mathcal{J}^{M1}}{\partial \alpha_{2,i,j}} &= \frac{1}{2} \frac{\partial(\mathcal{D}^2)}{\partial \alpha_{2,i,j}} + \gamma \frac{\partial \mathcal{S}_2^{TP}}{\partial \alpha_{2,i,j}} = \mathcal{D}(\boldsymbol{\varphi}(\boldsymbol{x}, \boldsymbol{\alpha})) \nabla_{\varphi_2} T(\boldsymbol{\varphi}(\boldsymbol{x}, \boldsymbol{\alpha})) B_{i,j}(\boldsymbol{x}) + \gamma \frac{\partial \mathcal{S}_2^{TP}}{\partial \alpha_{2,i,j}}. \end{aligned}$$

Finally, we apply a quasi-Newton method to (4.4) that requires second order information from the gradient of the previous step and converges faster than the steepest descent method. Detailed implementation of the quasi-Newton method [51, 102] is given in Algorithm 4.

Note that all vectors here are of size $2N_1N_2 \times 1$ or $2N^2 \times 1$. We usually take $m = 10$ while the choice of γ is to be discussed shortly.

Initial guess $\boldsymbol{\alpha}^{(0)}$. The well known multilevel method [35] can be adapted to image registration for providing a good initial solution. On the finest level (level 1), we set up $L - 1$ coarse levels using standard coarsening. The idea is to solve the minimisation problem (4.1) starting at the coarsest level L. Each coarser solution is propagated towards the finest level $L = 1$. With standard coarsening, B-spline functions from adjacent levels are related [59]. Consider line by line propagation of spline coefficients. Following [104, 59], starting from a coarse level k's coefficients $\alpha_{i,j},\ \alpha_{i,j+1},\ \alpha_{i,j+2},\ \alpha_{i,j+3}$ over one line of a control box, the finer level $k-1$ coefficients

Algorithm 4 Application of LBFGS [51, 102] for solving registration model.
$(\boldsymbol{\varphi}(\boldsymbol{x},\boldsymbol{\alpha}),\boldsymbol{\alpha}) \leftarrow \text{LBFGS}(\gamma,\mathcal{J},\boldsymbol{\alpha}^{(0)},m,\boldsymbol{maxit},\boldsymbol{TOL})$

```
1. Compute $g^{(0)} = \nabla\mathcal{J}(\boldsymbol{\alpha}^{(0)}),\ d^{(0)} = -g^{(0)}$.
2. For $k_1 = 0, 1, \ldots, \boldsymbol{maxit}$:
    (a) Compute $\boldsymbol{\alpha}^{(k_1+1)} = \boldsymbol{\alpha}^{(k_1)} + \kappa d^{(k_1)}$
            where $\kappa$ is the line search parameter, and $g^{(k_1+1)} = \nabla\mathcal{J}(\boldsymbol{\alpha}^{(k_1+1)})$.
    (b) Let $S^{(k_1)} = \boldsymbol{\alpha}^{(k_1+1)} - \boldsymbol{\alpha}^{(k_1)},\ y^{(k_1)} = g^{(k_1+1)} - g^{(k_1)},\ q = g^{(k_1)}$.
    (c) For $i = k_1-1, k_1-2, .., k_1-m$,   $\rho^{(i)} = \frac{1}{(y^{(i)})^T S^{(i)}},\ a^{(i)} = \rho^{(i)} q^T S^{(i)},\ q = q - a^{(i)} y^{(i)}$.
    (d) Set $p^{(k_1)} = \frac{(y^{(k_1)})^T S^{(k_1)}}{\|y^{(k_1)}\|_2^2}$ and $r = p^{(k_1)} q$.
    (e) For $i = k_1-m, k_1-m+1, .., k_1-1$,   $\beta = \rho^{(i)}(y^{(i)})^T r,\ r = r + (a^{(i)} - \beta) S^{(i)}$.
    (f) The new descent direction is given by $d^{(k_1)} = -r$.
    (g) Check the convergence criterion $\mathcal{J}(\boldsymbol{\alpha}^{(k_1+1)}) - \mathcal{J}(\boldsymbol{\alpha}^{(k_1)}) < \boldsymbol{TOL}$; if satisfied,
        exit else continue.
3. End for. Finish the algorithm with $\boldsymbol{\alpha} = \boldsymbol{\alpha}^{(k_1)}$ and compute $\boldsymbol{\varphi}(\boldsymbol{x},\boldsymbol{\alpha})$.
```

at positions $(i, j+\frac{1}{2}),\ (i, j+1),\ (i, j+\frac{3}{2}),\ (i, j+2)$ are given approximately by

$$\begin{bmatrix} \alpha_{i,j+\frac{1}{2}} \\ \alpha_{i,j+1} \\ \alpha_{i,j+\frac{3}{2}} \\ \alpha_{i,j+2} \end{bmatrix}^{(k-1)} = \frac{1}{8}\begin{bmatrix} 4 & 4 & 0 & 0 \\ 1 & 6 & 1 & 0 \\ 0 & 4 & 4 & 0 \\ 0 & 1 & 6 & 1 \end{bmatrix}\begin{bmatrix} \alpha_{i,j} \\ \alpha_{i,j+1} \\ \alpha_{i,j+2} \\ \alpha_{i,j+3} \end{bmatrix}^{(k)} \tag{4.5}$$

and at positions $(i, j+1),\ (i, j+\frac{3}{2}),\ (i, j+2),\ (i, j+\frac{5}{2})$ by

$$\begin{bmatrix} \alpha_{i,j+1} \\ \alpha_{i,j+\frac{3}{2}} \\ \alpha_{i,j+2} \\ \alpha_{i,j+\frac{5}{2}} \end{bmatrix}^{(k-1)} = \frac{1}{8}\begin{bmatrix} 1 & 6 & 1 & 0 \\ 0 & 4 & 4 & 0 \\ 0 & 1 & 6 & 1 \\ 0 & 0 & 4 & 4 \end{bmatrix}\begin{bmatrix} \alpha_{i,j} \\ \alpha_{i,j+1} \\ \alpha_{i,j+2} \\ \alpha_{i,j+3} \end{bmatrix}^{(k)}. \tag{4.6}$$

Recursively, $\boldsymbol{\alpha}^{(0)}$ on the finest level 1 is obtained by this procedure once $\boldsymbol{\alpha}^{(0)}$ on the coarsest level is computed first.

Choice of coupling parameter γ. We shall discuss how to choose γ using the above set up of L levels. In [82], the authors used a fixed parameter $\gamma = 0.01$ for (4.1) which is sufficient for a large class of problems. However a fixed γ can produce undesired effects for some images (usually involving global rotation). Here we adapt the parameter continuation approach used in [17] to find a suitable γ for (4.1). The idea has two steps.

First, on the coarsest level L, we use a bisection like method to find the best γ (and the corresponding $\boldsymbol{\alpha}$). We start with a large value of $\gamma = \gamma^{(0)} = \gamma_1 > 0$ and reduce γ systematically for subsequent iterations in order to decrease $\mathcal{J}^{\mathrm{M1}}$, or precisely use Algorithm 5 for optimal selection of the regularisation parameter γ for the parametric model (4.3):

$$(\gamma^*, \boldsymbol{\alpha}^*) \leftarrow \mathrm{Continuation}(\mathcal{J}^{\mathrm{M1}}, \gamma_0, \gamma_1, \tau, \boldsymbol{\alpha}^{(0)}, m, c, \boldsymbol{maxit}, \boldsymbol{TOL})$$

where $c = 1$. Here we use the notation $\gamma^{(m)}$ and $\boldsymbol{\alpha}^{(m)}$ for a solution pair at iteration m. With $\gamma = \gamma^{(m)}$, once we find the solution $\boldsymbol{\alpha}^{(m)}$ to (4.3), we set the new γ as $\gamma^{(m+1)} = \tau\gamma^{(m)}$ with the ratio τ depending on the consistency of this current solution:

$$\mathcal{J}^{\mathrm{M1}}_{\gamma^{(m)}}(\boldsymbol{\alpha}^{(m)}) < \mathcal{J}^{\mathrm{M1}}_{\gamma^{(m)}}(\boldsymbol{\alpha}^{(m-1)}); \tag{4.7}$$

if condition (4.7) is satisfied, we take $\tau = 0.5$ and continue, or if not, we restart the previous step with $\gamma^{(m)} = \tau\gamma^{(m-1)}$ and $\tau = 0.9$.

Algorithm 5 Direct search for the parameter γ on a single level.
$(\gamma^*, \boldsymbol{\alpha}^*) \leftarrow \mathrm{Continuation}(\mathcal{J}, \gamma_0, \gamma_1, \tau, \boldsymbol{\alpha}^{(0)}, m, c, \boldsymbol{maxit}, \boldsymbol{TOL})$

```
1. Initialize $\gamma^{(0)} = \gamma_1$ and $g^{(0)} = -\nabla\mathcal{J}(\boldsymbol{\alpha}^{(0)}),\ d^{(0)} = -g^{(0)}$.
2. For $k_2 = 0, 1, \dots, \boldsymbol{maxit}$:
   (a) If $c = 1, \gamma^{(k_2+1)} = \tau\gamma^{(k_2)}$ else $c = 2, \gamma^{(k_2+1)} = \gamma^{(k_2)} - \tau$.
   (b) Solve the registration problem to find $\boldsymbol{\alpha}^{(k_2+1)}$ c.f. Algorithm 4 where only
       perform step 2(a) until step 2(f).
   (c) If $\mathcal{J}_{\gamma^{(k_2+1)}}(\boldsymbol{\alpha}^{(k_2+1)}) < \mathcal{J}_{\gamma^{(k_2+1)}}(\boldsymbol{\alpha}^{(k_2)})$, then
       If $c = 1$, set $\tau = 0.5$, else $c = 2$, set $\tau = 0.01$ end if. $k_2 = k_2 + 1$ and go to
       step 2(e).
   (d) Else set $\tau = 0.9$ for $c = 1$ or $\tau = 0.001$ for $c = 2$ and go to step 2(e).
   (e) If $\gamma^{(k_2+1)} \leq \gamma_0$, exit
       else go to step 2(a).
3. End for. Finish the algorithm with $\gamma^* = \gamma^{(k_2)},\ \boldsymbol{\alpha}^* = \boldsymbol{\alpha}^{(k_2)}$.
```

Second, using the ideas of (4.5)-(4.6), we prolongate $\boldsymbol{\alpha}$ to the next finer level until $k = 1$. This multilevel image registration procedure for (4.3) is given in Algorithm 6:

$$(\boldsymbol{\varphi}(\boldsymbol{x}, \boldsymbol{\alpha}), \gamma) \leftarrow \mathrm{MLIR}(\boldsymbol{\alpha}^{(0,L)}, T, R, L, \mathcal{J}^{\mathrm{M1}}, \gamma_0, \gamma_1, \tau, m, c, \boldsymbol{maxit}, \boldsymbol{TOL})$$

where $c = 1, m = 10$.

Algorithm 6 Multilevel Solution for Image Registration.
$(\boldsymbol{\varphi}(\boldsymbol{x},\boldsymbol{\alpha}),\gamma) \leftarrow \mathrm{MLIR}(\boldsymbol{\alpha}^{(0,L)}, T, R, L, \mathcal{J}, \gamma_0, \gamma_1, \tau, m, c, \boldsymbol{maxit}, \boldsymbol{TOL})$

```
1. On the coarsest level $k = L$, find the best parameter $\gamma = \gamma^*$ and $\boldsymbol{\alpha}^{(*,L)}$.
   $(\gamma^*, \boldsymbol{\alpha}^{(*,L)}) \leftarrow \mathrm{Continuation}(\mathcal{J}, \gamma_0, \gamma_1, \delta, \tau, \boldsymbol{\alpha}^{(0,L)}, m, c, \boldsymbol{maxit}, \boldsymbol{TOL})$
2. Repeatedly, interpolate $\boldsymbol{\alpha}$ to next fine level. $\boldsymbol{\alpha}^{(0,k-1)} \leftarrow \mathrm{Interpolate}(\boldsymbol{\alpha}^{(*,k)})$
3. Solve the registration problem using Algorithm 4 on level $k-1$ until level $k=1$.
   $(\boldsymbol{\varphi}(\boldsymbol{x},\boldsymbol{\alpha}^{(*,k)}), \boldsymbol{\alpha}^{(*,k)})) \leftarrow \mathrm{LBFGS}(\gamma, \mathcal{J}, \boldsymbol{\alpha}^{(0,k)}, m, \boldsymbol{maxit}, \boldsymbol{TOL})$
4. Finish with $\boldsymbol{\varphi}(\boldsymbol{x},\boldsymbol{\alpha}) = \boldsymbol{\varphi}(\boldsymbol{x},\boldsymbol{\alpha}^{(*,1)})$ and $\gamma = \gamma^*$.
```

4.3 Non-parametric Image Registration: Linear Curvature Model

Below we consider the non-parametric image registration model with regulariser

$$\mathcal{S}^{\mathrm{FMC}}(\boldsymbol{u}) = \int_\Omega \left[(\Delta u_1)^2 + (\Delta u_2)^2\right] \mathrm{d}\Omega \tag{4.8}$$

and boundary conditions $\nabla u_l \cdot \boldsymbol{n} = 0$, and $\nabla \Delta u_l \cdot \boldsymbol{n} = 0$ for $l = 1, 2$, i.e. the following minimisation problem (to be denoted by M2):

$$\min_{u_1,u_2} \mathcal{J}^{\mathrm{M2}} = \frac{1}{2}\int_\Omega (T(\boldsymbol{x}+\boldsymbol{u}) - R(\boldsymbol{x}))^2 \ \mathrm{d}\Omega + \frac{\gamma}{2}\sum_{l=1}^{2}\int_\Omega (\Delta u_l)^2 \ \mathrm{d}\Omega. \tag{4.9}$$

The solution of (4.9) can be sought by either the optimise-discretise approach (i.e. the EL equation to be discretised by a numerical method) or the discretise-optimise approach (i.e. the discretised functional to be optimised). In either case we obtain a nonlinear system of equations, and solve iteratively to yield the final solution.

Here we take the latter discretise-optimise approach. The discretised form of (4.9), by a finite difference method, is

$$\begin{aligned}\min_{u_1,u_2} \mathcal{J}^{\mathrm{M2}} = \frac{h_1h_2}{2}\sum_{i=0}^{N_1-1}\sum_{j=0}^{N_2-1}\left[T(\boldsymbol{x}_{i,j}+\boldsymbol{u}(\boldsymbol{x}_{i,j})) - R(\boldsymbol{x}_{i,j})\right]^2 \quad + \\ \gamma\frac{h_1h_2}{2}\sum_{l=1}^{2}\sum_{i=0}^{N_1-1}\sum_{j=0}^{N_2-1}\left[-4u_l(\boldsymbol{x}_{i,j}) + u_l(\boldsymbol{x}_{i+1,j}) + u_l(\boldsymbol{x}_{i-1,j}) + u_l(\boldsymbol{x}_{i,j+1}) + u_l(\boldsymbol{x}_{i,j-1})\right]^2 .\end{aligned} \tag{4.10}$$

Although the notation for $\boldsymbol{x}_{i,j} = (x_{1,i}, x_{2,j})$ and $\boldsymbol{u}_{i,j} = (u_{1,i,j}, u_{2,i,j})$ is clear, we need to re-define the solution vector

$$\boldsymbol{u} = \begin{bmatrix} \boldsymbol{u}_1 \\ \boldsymbol{u}_2 \end{bmatrix}_{2N_1N_2\times 1}, \quad \boldsymbol{x} = \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \end{bmatrix}_{2N_1N_2\times 1}$$

where

$$\boldsymbol{u}_1 = \begin{bmatrix} u_{1,0,0} & u_{1,1,0} & \cdots & u_{1,N_1-1,0} & u_{1,0,1} & \cdots & u_{1,N_1-1,1} & u_{1,0,2} & \cdots & u_{1,N_1-1,N_2-1} \end{bmatrix}^T,$$

$$\boldsymbol{u}_2 = \begin{bmatrix} u_{2,0,0} & u_{2,1,0} & \cdots & u_{2,N_1-1,0} & u_{2,0,1} & \cdots & u_{2,N_1-1,1} & u_{2,0,2} & \cdots & u_{2,N_1-1,N_2-1} \end{bmatrix}^T$$

and $\boldsymbol{x}_1, \boldsymbol{x}_2$ are similarly defined. Then, the first SSD term in (4.10) is given in vector notation as

$$\mathcal{D}^{\text{SSD}} = \frac{h_1 h_2}{2} \boldsymbol{\Theta}^T \boldsymbol{\Theta} \quad \text{with} \quad \boldsymbol{\Theta} = [T(\boldsymbol{x}+\boldsymbol{u}) - R(\boldsymbol{x})]_{N_1 N_2 \times 1}.$$

Define the matrix $\boldsymbol{G}$ of size $N_1 N_2 \times 2N_1 N_2$ by

$$\boldsymbol{G} = \nabla_{\boldsymbol{u}} T(\boldsymbol{x}+\boldsymbol{u}) = \frac{\partial T(\boldsymbol{x}+\boldsymbol{u})}{\partial \boldsymbol{u}} = \Big[\frac{\partial T(\boldsymbol{x}+\boldsymbol{u})}{\partial \boldsymbol{u}_1} \;\Big|\; \frac{\partial T(\boldsymbol{x}+\boldsymbol{u})}{\partial \boldsymbol{u}_2} \Big]$$

i.e.

$$\boldsymbol{G} = \Big[\frac{\partial T(\boldsymbol{x}+\boldsymbol{u})}{\partial u_{1,0,0}} \;\Big|\; \frac{\partial T(\boldsymbol{x}+\boldsymbol{u})}{\partial u_{1,1,0}} \;\Big|\; \cdots \;\Big|\; \frac{\partial T(\boldsymbol{x}+\boldsymbol{u})}{\partial u_{1,N_1-1,N_2-1}} \;\Big|\; \frac{\partial T(\boldsymbol{x}+\boldsymbol{u})}{\partial u_{2,0,0}} \;\Big|\; \frac{\partial T(\boldsymbol{x}+\boldsymbol{u})}{\partial u_{2,1,0}} \;\Big|\; \cdots \;\Big|\; \frac{\partial T(\boldsymbol{x}+\boldsymbol{u})}{\partial u_{2,N_1-1,N_2-1}} \Big].$$

To find the minimum of equation (4.10) we need to solve

$$\nabla \mathcal{J}^{\text{M2}} = \boldsymbol{0}$$

which yields a system of nonlinear equations with unknown $\boldsymbol{u}$. Let

$$\begin{aligned} \mathcal{J}^{\text{M2}} &= \mathcal{D}^{\text{SSD}} + \gamma \mathcal{S}^{\text{FMC}} \\ &= \frac{h_1 h_2}{2} \boldsymbol{\Theta}^T \boldsymbol{\Theta} + \gamma \boldsymbol{u}^T \boldsymbol{B} \boldsymbol{u}, \end{aligned}$$

then,

$$\begin{aligned} \nabla \mathcal{J}^{\text{M2}} &= \nabla \mathcal{D}^{\text{SSD}} + \gamma \nabla \mathcal{S}^{\text{FMC}} \\ &= \boldsymbol{\Theta}^T \boldsymbol{G} + \gamma \boldsymbol{B} \boldsymbol{u} \end{aligned}$$

where $\boldsymbol{B}$ is a constant matrix of size $2N_1 N_2 \times 2N_1 N_2$ that consist of the coefficients of $\boldsymbol{u}$. We can write $\boldsymbol{B}$ as

$$\boldsymbol{B} = \begin{bmatrix} L^T L & \boldsymbol{0}_{N_1 \times N_2} \\ \boldsymbol{0}_{N_1 \times N_2} & L^T L \end{bmatrix}$$

where L is a block tridiagonal matrix from the regularisation term. For example, for

$\Omega^h = [0,2]^2$, then

$$L = \begin{bmatrix} -2 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & -3 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -2 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & -3 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & -4 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & -3 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & -2 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & -3 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & -2 \end{bmatrix}.$$

We can solve equation (4.10) iteratively using the LBFGS method by Algorithm 4:

$$(\boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x}), \boldsymbol{u}(\boldsymbol{x})) \leftarrow \text{LBFGS}(\gamma, \mathcal{J}^{\text{M2}}, \boldsymbol{u}^{(0)}, m, \boldsymbol{maxit}, \boldsymbol{TOL}).$$

Here selecting the regularisation parameter γ and the corresponding initial guess for (4.10) can be done by the multilevel approach

$$(\boldsymbol{u}(\boldsymbol{x}), \gamma) \leftarrow \text{MLIR}(\boldsymbol{u}^{(0,L)}, T, R, L, \mathcal{J}^{\text{M2}}, \gamma_0, \gamma_1, \tau, m, 2, \boldsymbol{maxit}, \boldsymbol{TOL})$$

as in Algorithm 6 where on the coarsest level one would use

$$(\gamma^*, \boldsymbol{u}^*) \leftarrow \text{Continuation}(\mathcal{J}^{\text{M2}}, \gamma_0, \gamma_1, \tau, \boldsymbol{u}^{(0)}, m, 2, \boldsymbol{maxit}, \boldsymbol{TOL})$$

in Algorithm 5.

4.4 A Decomposition Model Combining Parametric and Non-parametric Deformation

In this section, we present our new model based on the above two models, the parametric model M1 (4.1) and the non-parametric model M2 (4.9), which is expected to outperform each individual method. We propose a general framework for image registration $\boldsymbol{\varphi} = \boldsymbol{\varphi}(\boldsymbol{x}) : \mathbb{R}^2 \to \mathbb{R}^2$ in the form

$$\boldsymbol{\varphi}(\boldsymbol{x}) = \boldsymbol{u}_p + \boldsymbol{u}_{np} \tag{4.11}$$

where the final transformation is a decomposition of the parametric and non-parametric transformations.

Based on [65, 66], the domain of the transformation consists of global and local. The transformation is called global if it is applied to the entire image and it is called local if it is applied to a small portion of the image. The parameters for the global transformation are used to calculate the displacement for all pixels in the image domain. For the local transformation, the parameters are valid for the small patch in the image domain. In the case of the non-parametric image registration models, the parameters are the pixels' level. Thus, they are classified as local transformation.

Our proposed idea is that, firstly, the overall deformation consists of both global and local displacement by means of the decomposition of the parametric and non-parametric transformation and, secondly, a large proportion of the overall deformation is captured by the parametric transformation. These three decompositions fulfill this framework:

$$\begin{aligned}
&\textbf{Model 3.1}\ \ \boldsymbol{\varphi}(\boldsymbol{x}) = \boldsymbol{u}_{\text{global}} + \boldsymbol{u}_{\text{local}}, \text{ for example, } \ \boldsymbol{\varphi}(\boldsymbol{x}) = \boldsymbol{u}_{\text{affine}} + \boldsymbol{u}_{\text{diff}}\\
&\textbf{Model 3.2}\ \ \boldsymbol{\varphi}(\boldsymbol{x}) = \boldsymbol{u}_{\text{local}_1} + \boldsymbol{u}_{\text{global+local}_2}, \text{ for example, } \ \boldsymbol{\varphi}(\boldsymbol{x}) = \boldsymbol{u}_{\text{landmark}} + \boldsymbol{u}_{\text{HWC}}\\
&\textbf{Model 3.3}\ \ \boldsymbol{\varphi}(\boldsymbol{x}) = \boldsymbol{u}_{\text{global}_1\text{+local}_1} + \boldsymbol{u}_{\text{global}_2\text{+local}_2}.
\end{aligned} \tag{4.12}$$

We recommended this particular choice

$$\boldsymbol{\varphi}(\boldsymbol{x}) = \boldsymbol{u}_{\text{cubic B-spline}} + \boldsymbol{u}_{\text{FMC}} \tag{4.13}$$

because both cubic B-spline or M1 [103, 82, 80] and FMC or M2 [25, 26, 67, 24] are capable of representing global and local displacement. We shall name this choice (of **Model 3.3**) as M3 and the new variation problem takes the particular form

$$\begin{aligned}
\min_{\boldsymbol{\alpha},\boldsymbol{u}_{np}(\boldsymbol{x})} \mathcal{J}^{\text{M3}}_{\gamma_p,\gamma_{np}}(\boldsymbol{\varphi}(\boldsymbol{x})) = \frac{1}{2}\|T(\boldsymbol{u}_p(\boldsymbol{x},\boldsymbol{\alpha}) + \boldsymbol{u}_{np}(\boldsymbol{x})) - R\|_2^2 \\
+\ \gamma_p\mathcal{S}_p(\boldsymbol{u}_p(\boldsymbol{x},\boldsymbol{\alpha}))\ +\ \frac{\gamma_{np}}{2}\mathcal{S}_{np}(\boldsymbol{u}_{np}(\boldsymbol{x}))
\end{aligned} \tag{4.14}$$

where $\mathcal{S}_p = \mathcal{S}^{TP}$ is as defined by (4.1) and $\mathcal{S}_{np} = \mathcal{S}^{\text{FMC}}$ is as in (4.8). **Models 3.1** and **3.2** can also be used if we have some prior knowledge of R and T. Problem (4.14) can be solved by the alternating minimisation resulting in two subproblems

$$\min_{\boldsymbol{\alpha}} \mathcal{J}^{I}_{\gamma_p} = \frac{1}{2}\|(T(\boldsymbol{u}_p(\boldsymbol{x},\boldsymbol{\alpha}) + \boldsymbol{u}_{np}(\boldsymbol{x})) - R)\|_2^2 + \gamma_p\mathcal{S}_p(\boldsymbol{u}_p(\boldsymbol{x},\boldsymbol{\alpha})) \tag{4.15}$$

and

$$\min_{\boldsymbol{u}_{np}(\boldsymbol{x})} \mathcal{J}^{II}_{\gamma_{np}} = \frac{1}{2}\|(T(\boldsymbol{u}_p(\boldsymbol{x},\boldsymbol{\alpha}) + \boldsymbol{u}_{np}(\boldsymbol{x})) - R)\|_2^2 + \frac{\gamma_{np}}{2}\mathcal{S}_{np}(\boldsymbol{u}_{np}). \tag{4.16}$$

We take the discretise-optimise approach for numerical solutions. At the kth iteration, the alternate updates are done as follows:

$$
\begin{aligned}
\boldsymbol{u}_p^{(k+1)}(\boldsymbol{x},\boldsymbol{\alpha}) \quad &\leftarrow \min_{\boldsymbol{\alpha}} \mathcal{J}^I_{\gamma_p} = \frac{1}{2}\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}(T(\boldsymbol{u}_p(\boldsymbol{x}_{i,j},\boldsymbol{\alpha})+\boldsymbol{u}_{np}(\boldsymbol{x}_{i,j}))-R(\boldsymbol{x}_{i,j}))^2 \\
&\qquad + \quad \gamma_p\mathcal{S}_p(u_{p,1})+\gamma_p\mathcal{S}_p(u_{p,2}) \qquad\qquad (4.17)\\
&= \frac{1}{2}\sum_{c=0}^{N-1}\sum_{d=0}^{N-1}(T(\boldsymbol{u}_p(\boldsymbol{x}_{c,d},\boldsymbol{\alpha})+\boldsymbol{u}_{np}(\boldsymbol{x}_{c,d}))-R(\boldsymbol{x}_{c,d}))^2+ \\
&\gamma_p\sum_{l=1}^{2}\sum_{c=0}^{N-1}\sum_{d=0}^{N-1}((u_{p,l_{x_1x_1}}(\boldsymbol{x}_{c,d},\alpha_l)^2)+2(u_{p,l_{x_1x_2}}(\boldsymbol{x}_{c,d},\alpha_l))^2 \\
&\qquad + (u_{p,l_{x_2x_2}}(\boldsymbol{x}_{c,d},\alpha_l))^2),
\end{aligned}
$$

$$
\begin{aligned}
\boldsymbol{u}_{np}^{(k+1)}(\boldsymbol{x}) \quad &\leftarrow \quad \min_{\boldsymbol{u}_{np}(\boldsymbol{x})} \mathcal{J}^{II}_{\gamma_{np}} = \frac{1}{2}\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}(T(\boldsymbol{u}_p(\boldsymbol{x}_{i,j},\boldsymbol{\alpha})+\boldsymbol{u}_{np}(\boldsymbol{x}_{i,j}))-R(\boldsymbol{x}_{i,j}))^2 \\
&\qquad + \quad \frac{\gamma_{np}}{2}\mathcal{S}_{np}(\boldsymbol{u}_{np}(\boldsymbol{x}_{i,j})) \qquad\qquad (4.18)\\
&= \frac{1}{2}\sum_{c=0}^{N-1}\sum_{d=0}^{N-1}(T(\boldsymbol{u}_p(\boldsymbol{x}_{c,d},\boldsymbol{\alpha})+\boldsymbol{u}_{np}(\boldsymbol{x}_{c,d}))-R(\boldsymbol{x}_{c,d}))^2 \\
&+ \frac{\gamma_{np}}{2}\sum_{l=1}^{2}\sum_{c=0}^{N-1}\sum_{d=0}^{N-1}\Big[-4u_{np,l}(\boldsymbol{x}_{c,d})+u_{np,l}(\boldsymbol{x}_{c+1,d})+u_{np,l}(\boldsymbol{x}_{c-1,d}) \\
&\qquad + u_{np,l}(\boldsymbol{x}_{c,d+1})+u_{np,l}(\boldsymbol{x}_{c,d-1})\Big]^2.
\end{aligned}
$$

As before, to solve (4.17), the $\boldsymbol{u}_p(\boldsymbol{x},\boldsymbol{\alpha})$ are defined by cubic B-splines with coefficients $\boldsymbol{\alpha}$:

$$
u_{p,1}(\boldsymbol{x},\alpha_1)=\sum_{i=-1}^{n_x-2}\sum_{j=-1}^{n_x-2}B_{i,j}(\boldsymbol{x})\alpha_{1,i,j}, \qquad u_{p,2}(\boldsymbol{x},\alpha_2)=\sum_{i=-1}^{n_x-2}\sum_{j=-1}^{n_x-2}B_{i,j}(\boldsymbol{x})\alpha_{2,i,j}
$$

Furthermore, Algorithm 4 can be used to update $\boldsymbol{\alpha}$ and hence $\boldsymbol{u}_p(\boldsymbol{x},\boldsymbol{\alpha})$. To solve equation (4.18), noting $\boldsymbol{u}_p(\boldsymbol{x},\boldsymbol{\alpha})$ is fixed, we follow the same Algorithm 4.

An alternating minimisation method for model (4.14) is given in Algorithm 7:

Algorithm 7 Alternating minimisation algorithm for the decomposition model.
$(\gamma_p, \gamma_{np}, \boldsymbol{u}_p(\boldsymbol{x}, \boldsymbol{\alpha}), \boldsymbol{u}_{np}(\boldsymbol{x})) \leftarrow \text{Decomposition}(T, R, \boldsymbol{u}_{np}^{(0,1)}, \boldsymbol{\alpha}^{(0,1)}, L, \gamma_0, \gamma_1, \tau, m, \boldsymbol{maxit}, \boldsymbol{TOL})$

```
1. Initialize $\gamma_p^{(0)} = \gamma_{p,1}, \gamma_{np}^{(0)} = \gamma_{np,1}$. Restrict $\boldsymbol{u}_{np}^{(0,1)}$ and $\boldsymbol{\alpha}^{(0,1)}$ to the coarsest grid $L$. $\boldsymbol{u}_{np}^{(0,L)} \leftarrow \text{Restrict}(\boldsymbol{u}_{np}^{(0,1)})$ and $\boldsymbol{\alpha}^{(0,L)} \leftarrow \text{Restrict}(\boldsymbol{\alpha}^{(0,1)})$.
2. For $k_3 = 0, 1, \dots, \boldsymbol{maxit}$ : (Outer iteration for alternating minimisation).
    (a) Compute $\boldsymbol{\alpha}^{(k_3+1,1)}$ and $\gamma_p^{(k_3+1)}$ using equation (4.15) c.f Algorithm 6 with $\boldsymbol{\alpha}^{(k_3,L)}$ as initial guess. $(\boldsymbol{u}_p^{(k_3+1,1)}(\boldsymbol{x}, \boldsymbol{\alpha}), \gamma_p^{(k_3+1)}) \leftarrow \text{MLIR}(\boldsymbol{\alpha}^{(k_3,L)}, T(\boldsymbol{u}_{np}^{(k_3,1)}(\boldsymbol{x})), R, L, \mathcal{J}_{\gamma_p}^{I}, \gamma_{p,0}, \gamma_{p,1}, \tau, m, 1, \boldsymbol{maxit}, \boldsymbol{TOL})$
    (b) Restrict $\boldsymbol{u}_p^{(k_3+1,1)}$ at fine level to level $L$. $\boldsymbol{u}_p^{(k_3+1,L)} \leftarrow \text{Restrict}(\boldsymbol{u}_p^{(k_3+1,1)})$ and find the respective $\boldsymbol{\alpha}^{(k_3+1,L)}$ from $\boldsymbol{u}_p^{(k_3+1,L)}(\boldsymbol{x}, \boldsymbol{\alpha})$.
    (c) Compute $\boldsymbol{u}_{np}^{(k_3+1)}(\boldsymbol{x})$ and $\gamma_{np}^{(k_3+1)}$ using equation (4.16) c.f Algorithm 6 with $\boldsymbol{u}_{np}^{(k_3,L)}(\boldsymbol{x})$ as initial guess. $(\boldsymbol{u}_{np}^{(k_3+1,1)}(\boldsymbol{x}), \gamma_{np}^{(k_3+1)}) \leftarrow \text{MLIR}(\boldsymbol{u}_{np}^{(k_3,L)}, T(\boldsymbol{u}_p^{(k_3+1,1)}(\boldsymbol{x}, \boldsymbol{\alpha})), R, L, \mathcal{J}_{\gamma_{np}}^{II}, \gamma_{np,0}, \gamma_{np,1}, \tau, m, 2, \boldsymbol{maxit}, \boldsymbol{TOL})$
    (d) Restrict $\boldsymbol{u}_{np}^{(k_3+1,1)}(\boldsymbol{x})$ to $\boldsymbol{u}_{np}^{(k_3+1,L)}(\boldsymbol{x})$. $\boldsymbol{u}_{np}^{(k_3+1,L)}(\boldsymbol{x}) \leftarrow \text{Restrict}(\boldsymbol{u}_{np}^{(k_3+1,1)}(\boldsymbol{x}))$.
    (e) If $\|\gamma_p^{(k_3+1)} - \gamma_p^{(k_3)}\| < \boldsymbol{TOL}$, $\|\gamma_{np}^{(k_3+1)} - \gamma_{np}^{(k_3)}\| < \boldsymbol{TOL}$ or $\mathcal{J}^{\text{M3}}_{\gamma_p^{(k_3+1)}, \gamma_{np}^{(k_3+1)}}(\boldsymbol{u}_p^{(k_3+1)}(\boldsymbol{x}, \boldsymbol{\alpha}), \boldsymbol{u}_{np}^{(k_3+1)}(\boldsymbol{x})) > \mathcal{J}^{\text{M3}}_{\gamma_p^{(k_3)}, \gamma_{np}^{(k_3)}}(\boldsymbol{u}_p^{(k_3)}(\boldsymbol{x}, \boldsymbol{\alpha}), \boldsymbol{u}_{np}^{(k_3)}(\boldsymbol{x}))$ exit, else continue with the next iteration $k_3$.
3. End for. Accept $\gamma_p = \gamma_p^{(k_3+1)}, \gamma_{np} = \gamma_{np}^{(k_3+1)}$.
4. Compute the final registration algorithm on the finest level only for equation (4.15) . $(\boldsymbol{u}_p(\boldsymbol{x}, \boldsymbol{\alpha}), \boldsymbol{\alpha}) \leftarrow \text{LBFGS}(\gamma_p, \mathcal{J}_{\gamma_p}^{I}, \boldsymbol{\alpha}^{(k_3+1,1)}, m, \boldsymbol{maxit}, \boldsymbol{TOL})$
5. Compute the final registration algorithm on the finest level only for equation (4.16) . $(\boldsymbol{u}_{np}(\boldsymbol{x}), \boldsymbol{u}_{np}(\boldsymbol{x})) \leftarrow \text{LBFGS}(\gamma_{np}, \mathcal{J}_{\gamma_{np}}^{II}, \boldsymbol{u}_{np}^{(k_3+1,1)}, m, \boldsymbol{maxit}, \boldsymbol{TOL})$
6. Finish the iteration with $\gamma_p, \gamma_{np}, \boldsymbol{u}_p(\boldsymbol{x}, \boldsymbol{\alpha})$, and $\boldsymbol{u}_{np}(\boldsymbol{x})$.
```

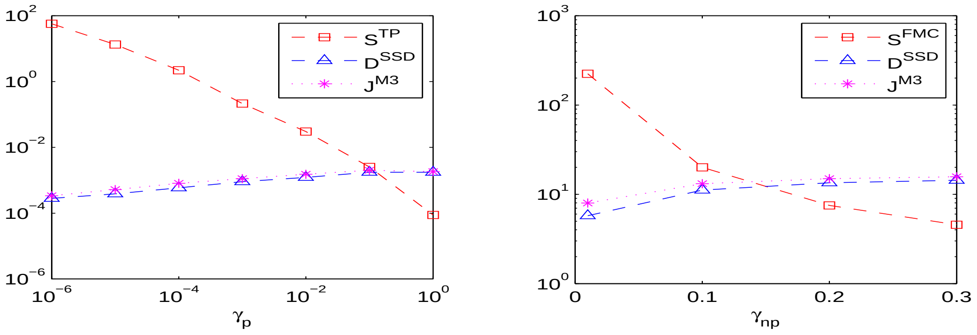
4.5 Optimal Values for the Regularisation Parameters γ_p and γ_{np}

An appropriate choice for the regularisation parameters is a crucial aspect in solving an ill-posed problem. Several methods exist such that generalised cross validation and detection of the corner of the L-curve, but none of them are directly applicable to image registration due to the non-linearity of the fidelity term. The L-curve approach involves a plot using a log-log scale of the norm of the solution and the residual norm where the optimal regularisation parameter is given at the 'vertex' of the 'L'. Thus, most of the authors choose the regularisation parameters based on prior information: how much fitting one requires in the joint objective function between fitting and regularisation terms. This is equivalent to the discrepancy principle where the regularisation term is limited to some upper bound.

Our approach for the decomposition model is based on the continuation of the regularisation parameters to balance the fitting and regularisation terms. We allow the parameters to vary within a specified interval $[\gamma_{p,0}, \gamma_{p,1}]$ and $[\gamma_{np,0}, \gamma_{np,1}]$ for the parametric and non-parametric parts of the decomposition model, respectively. Our idea is simple: we consider the choice of the two parameters in turn (i.e. fix one and optimise the other) and our optimisation is based on the observations that i) the fitting energy increases and the regularisation energy reduces when the regularisation parameter increases; (ii) the underlying deformation (image transformation) has non-physical folding when the regularisation parameter is extremely small. Since smaller values will allow larger displacement, the resulting transform does not preserve the topology of the grid. This phenomenon is indicated by folding and cracking of the deformed grid and can be observed via the determinant of the Jacobian matrix $J(x_{1,c}, x_{2,d})$. We aim for

$$J(x_{1,c}, x_{2,d}) = \begin{vmatrix} \frac{\partial \varphi_1}{\partial x_{1,c}} & \frac{\partial \varphi_1}{\partial x_{2,d}} \\ \frac{\partial \varphi_2}{\partial x_{1,c}} & \frac{\partial \varphi_2}{\partial x_{2,d}} \end{vmatrix} > 0 \qquad (4.19)$$

to ensure that the transformation is plausible and meaningful. Since the Jacobian is positive everywhere, by the inverse function theorem, the transformation $\boldsymbol{\varphi}(\boldsymbol{x})$ is locally invertible. In other words, $\boldsymbol{\varphi}(\boldsymbol{x})$ is bijective. In our decomposition method M3, we find the lower limits of $[\gamma_{p,0}, \gamma_{p,1}]$ and $[\gamma_{np,0}, \gamma_{np,1}]$ based on the calculation of (4.19). That is, it is always possible to start with a small $\gamma_{p,0}$ (e.g. 10^{-6}) and implement the model until $J(x_{1,c}, x_{2,d}) > 0$ is satisfied automatically. Subsequently, we increase the parameter till both the fitting and the regularisation terms are approximately balanced, as illustrated in Fig.4.1.



(a) (b)

Figure 4.1: Plots of three energy terms to aid choice of γ_p and γ_{np}.

The profile plot for $\mathcal{J}^{\text{M3}}$ and its two constituents is shown in Figure 4.1 for increasing values of γ_p and γ_{np}. We can observe that smaller values of γ_p and γ_{np} give smaller values of $\mathcal{J}^{\text{M3}}$; however at both beginnings, (4.19) is not satisfied. Therefore using this idea for test images in Section 5 we attain $[\gamma_{p,0}, \gamma_{p,1}] = [0.0001, 0.1]$ and $[\gamma_{np,0}, \gamma_{np,1}] = [0.1, 0.2]$.

4.6 Numerical Results

We perform four numerical experiments to show that our decomposition model (Algorithm 7, i.e. M3) has better performance than the individual models, M1 and M2 through comparing their relative reduction of the similarity measure indicated by ε where

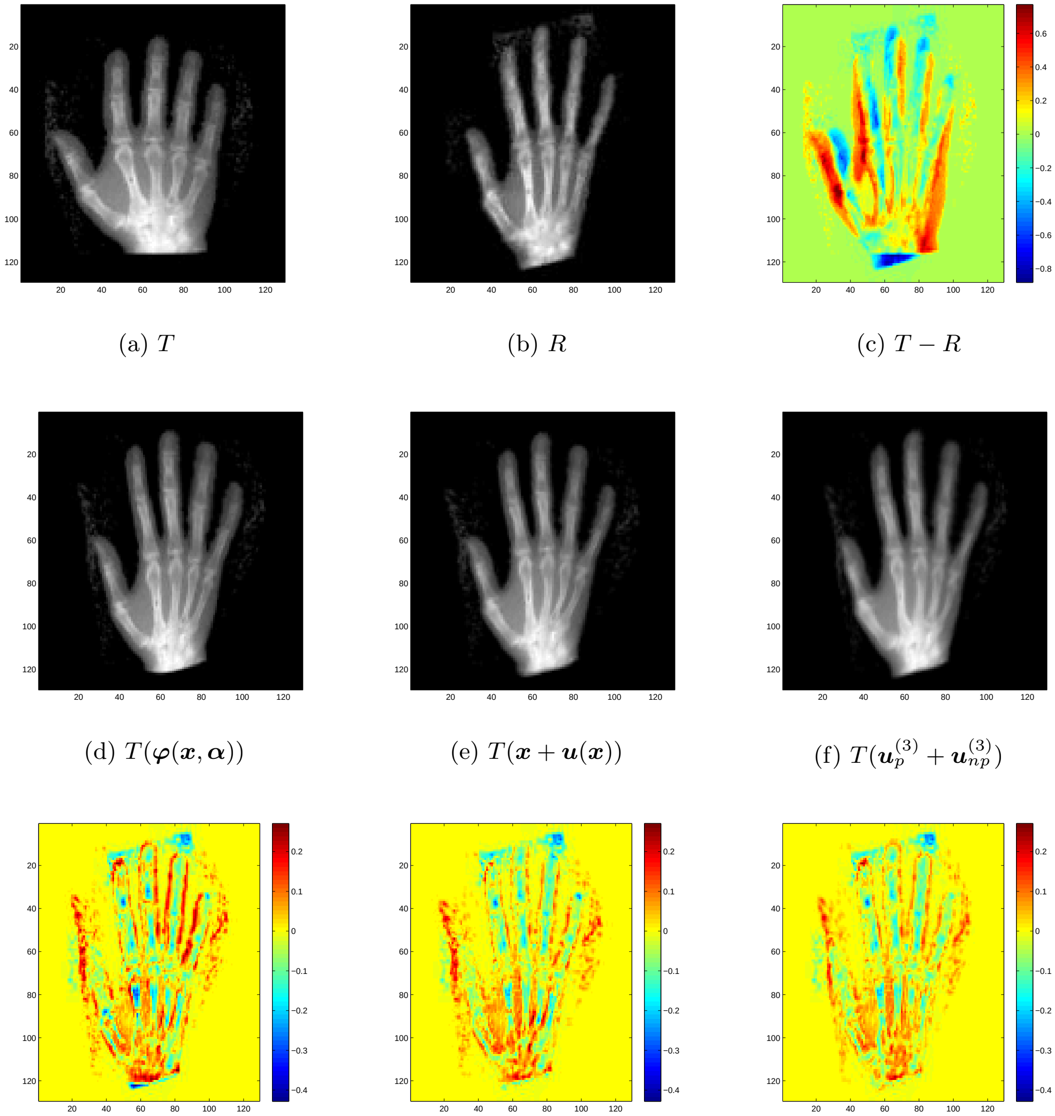
$$\varepsilon = \frac{\mathcal{D}(T, R, \boldsymbol{\varphi}^{(*)})}{\mathcal{D}(T, R, \boldsymbol{\varphi}^{(0)})}. \tag{4.20}$$

The current pairwise differences before and after registration are presented as difference images with a scale bar to highlight the regions of large differences.

The first experiment is to illustrate that M3 gives similar results in visual quality to M1 and M2 for a smooth problem, although M3 has the smallest error value ε. The second experiment compares M3 with M1 and M2 for an artificial deformation problem to show the advantage of M3 over M1 and M2 in both the visual quality and ε value. The third experiment uses medical MR images of human brain of different individuals in atlas construction. In all tests, the reference and template images are given before hand. However, we can interchange these two images because it is a matter of choice fact.

4.6.1 Test 1: A Pair of Smooth X-ray Images

Here, we take the smooth X-ray images from [68] of size 129 × 129. We obtain the desirable result by M3 after three outer iterations in Algorithm 7. In this test, all three models M1, M2 and M3 are able to solve the problem reasonably as shown in Figure 4.2. The lowest value of ε is given by M3 followed by M2 and M1; for this smooth problem, M3 shows better results than M1 and M2 though M1 and M2 give acceptable results.

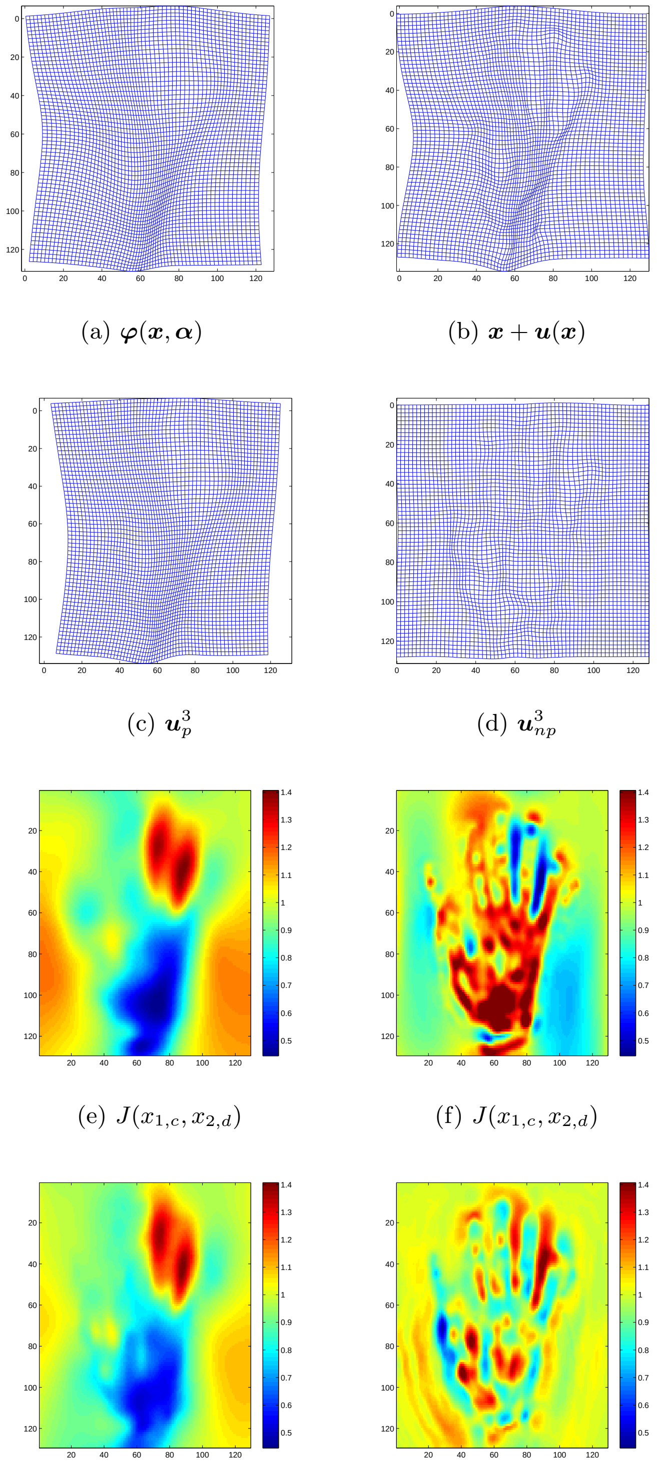


(a) T (b) R (c) $T - R$

(d) $T(\boldsymbol{\varphi}(\boldsymbol{x}, \boldsymbol{\alpha}))$ (e) $T(\boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x}))$ (f) $T(\boldsymbol{u}_p^{(3)} + \boldsymbol{u}_{np}^{(3)})$

(g) $\varepsilon = 0.1150$ (h) $\varepsilon = 0.0836$ (i) $\varepsilon = 0.0702$

Figure 4.2: Top row and left to right: template, reference and the difference between the template and reference images. Middle row and left to right: results of Test 1 using M1, M2, and M3. Bottom row shows the differences of the transform template images (middle row) and reference images. All three models are able to register Test 1 but a smaller value of ε is given by M3.



(a) $\boldsymbol{\varphi}(\boldsymbol{x}, \boldsymbol{\alpha})$

(b) $\boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x})$

(c) $\boldsymbol{u}_p^3$

(d) $\boldsymbol{u}_{np}^3$

(e) $J(x_{1,c}, x_{2,d})$

(f) $J(x_{1,c}, x_{2,d})$

(g) $J(x_{1,c}, x_{2,d})$

(h) $J(x_{1,c}, x_{2,d})$

Figure 4.3: First to second row and left to right: deformation field applied to the regular grid for Test 1 using $\mathrm{M1}, \mathrm{M2}, \mathrm{M3}$ after the parametric part and M3 after the non-parametric part. Third to fourth row and left to right: the values of the determinant of the Jacobian matrix for the corresponding deformation on the top row. It is clear that the determinant of the Jacobian matrix is positive everywhere.

The deformation field and the determinant of the Jacobian matrix for all methods are shown in Figure 4.3. We can observe that the transformations are smooth and the topology of the deformed grid is preserved since we have positive values of the determinant of the Jacobian matrix.

4.6.2 Test 2: A Pair of Lena Images

This Test 2 is similarly tested in [63] where the Lena image is artificially deformed. From Fig. 4.4, we see that M1 indeed performs better than M2 and the best result is given by M3. Our new approach results in a significantly improved registration as indicated visually by the smaller amount of darker region in the bottom row in Figure 4.4. The deformation field and the determinant of the Jacobian matrix for all methods are shown in Figure 4.5. We can observe that the determinant of the Jacobian matrix is positive everywhere.

4.6.3 Test 3: A Pair of Brain MR Images

Our third Test 3 of medical images in size 257 × 257 is taken from the Internet Brain Segmentation Repository (IBSR) where 20 normal MR brain images and their manual segmentations are provided. We choose a pair of two individuals with different sizes of ventricle to illustrate how large deformation is modeled by M1-3. Figure 4.6 shows the test and the registration results. We can see that all three models are able to register this Test 3 where the lowest value of ε is given by M3. After registration, we evaluate the mean squared error (MSE) and the dice metric for white and grey matter where the original values before and after registration are given in Table 4.1. The dice metric is defined as

$$\text{Dice metric} = \frac{2(T \cap R)}{(|T| + |R|)} \tag{4.21}$$

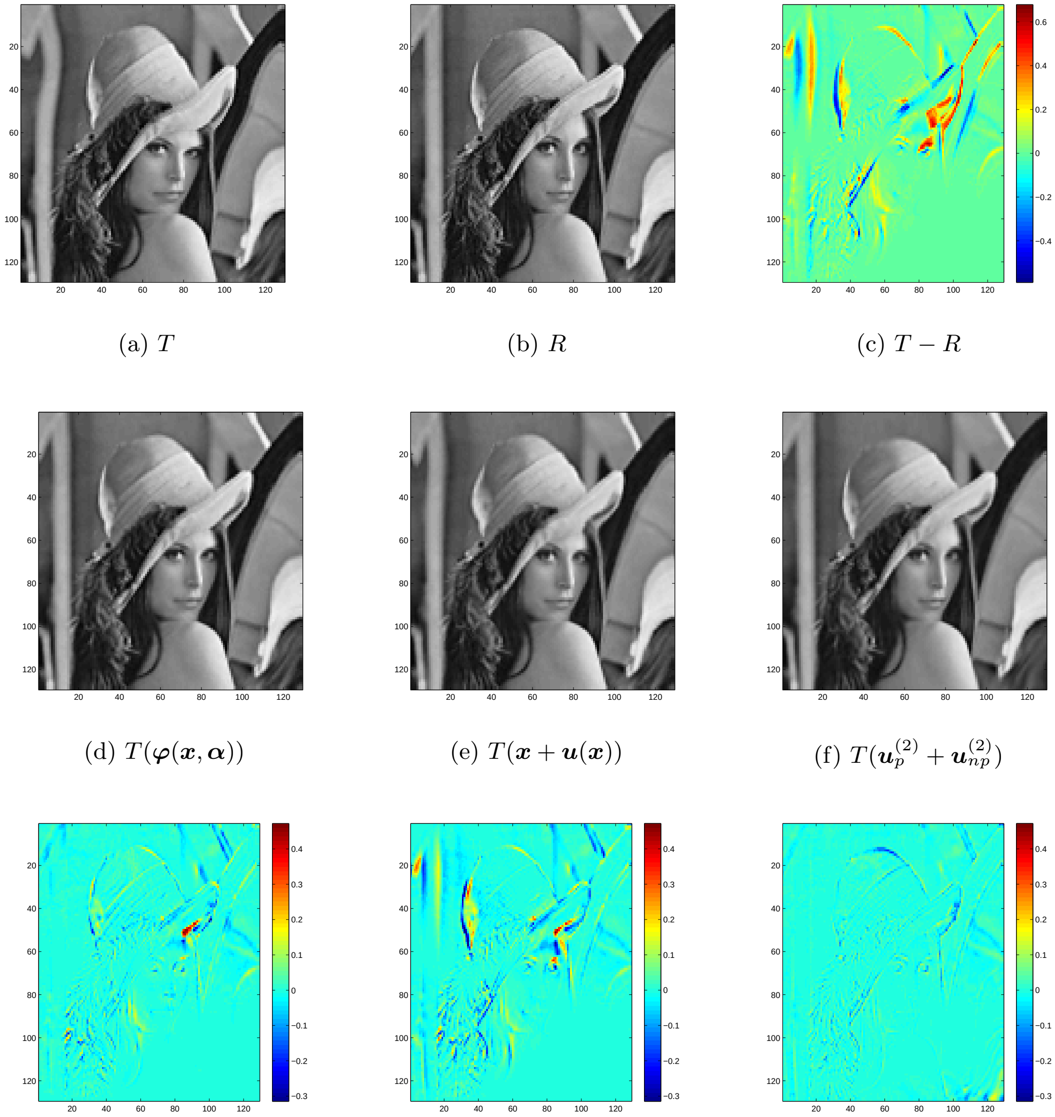
while the MSE is

$$MSE = \frac{1}{N_1 N_2} \sum_{i=0}^{N_1-1} \sum_{j=0}^{N_2-1} (T_{ij} - R_{ij})^2.$$

Here, for the dice metric, the larger it is the better the registration is while for MSE the smaller the better. From Table 4.1, we see that M3 performs better than M1 and M2.

Measure	T	M1	M2	M3
Mean squared error (MSE)	0.0337	0.0124	0.0118	0.0029
Dice metric white matter	0.5057	0.5673	0.5638	0.5742
Dice metric grey matter	0.5636	0.7241	0.7275	0.7311

Table 4.1: Comparison of MSE, and the dice metric for white and grey matter for segmented images of Test 3 before registration, and after registration using M1, M2 and M3. Clearly M3 is the best.



(a) T

(b) R

(c) $T - R$

(d) $T(\boldsymbol{\varphi}(\boldsymbol{x}, \boldsymbol{\alpha}))$

(e) $T(\boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x}))$

(f) $T(\boldsymbol{u}_p^{(2)} + \boldsymbol{u}_{np}^{(2)})$

(g) $\varepsilon = 0.1385$

(h) $\varepsilon = 0.2635$

(i) $\varepsilon = 0.0623$

Figure 4.4: Top row and left to right: template, reference and the difference between the template and reference images. Middle row and left to right are the results of Test 2 using M1, M2, and M3. Bottom row shows the differences of the transform template images (middle row) and reference image. The best result is given by M3 where we can see that the method gives the smallest error as depicted on the bottom row.

The deformation field and the determinant of the Jacobian matrix for all methods are shown in Figure 4.7. Since we have positive values of the determinant of the Jacobian matrix, the transformation maintains a diffeomorphism.

We also try to register Test 3 (cropped) in size 129 × 129 using all three models and we observe the same result where MSE is the least by M3 and the dice metric is the largest for white and grey matter also by M3 as shown in Table 4.2.

Measure	T	M1	M2	M3
Mean squared error (MSE)	0.0293	0.0089	0.0089	0.0083
Dice metric white matter	0.3059	0.4337	0.3855	0.4403
Dice metric grey matter	0.4603	0.5828	0.5930	0.6012

Table 4.2: Comparison of MSE, and the dice metric for white and grey matter for segmented images of Test 3 for $N = N_1 = N_2 = 129$ before registration, and after registration using M1, M2 and M3. Our method M3 outperforms M1 and M2.

4.6.4 Test 4: A Challenging Example of Large Deformation

Designing a registration model capable of solving both smooth and non-smooth problems without folding in the deformation field is a difficult task. We present Test 4 as a hard problem to register even using non-parametric image registration method. To solve this particular test we change $(\delta_{x_1}, \delta_{x_2}) = (8, 8)$ instead of using $(\delta_{x_1}, \delta_{x_2}) = (4, 4)$ in M1 and M3. It is because a larger spacing of the control points is able to recover larger deformation compared to a smaller spacing of the control points. Figure 4.8 shows the obtained results: we can observe that the features (corners of the boxes) are well captured by M3 in comparison to M1 which distorts the features. As somewhat expected, M2 cannot solve the problem at all. Deformation field given by M2 is a local minimum since the problem is non-convex due to the sum of squared difference as the fitting term. Meanwhile, M1 and M3 manage to obtain a global minimiser for this particular problem due to the large spacing of the control points.

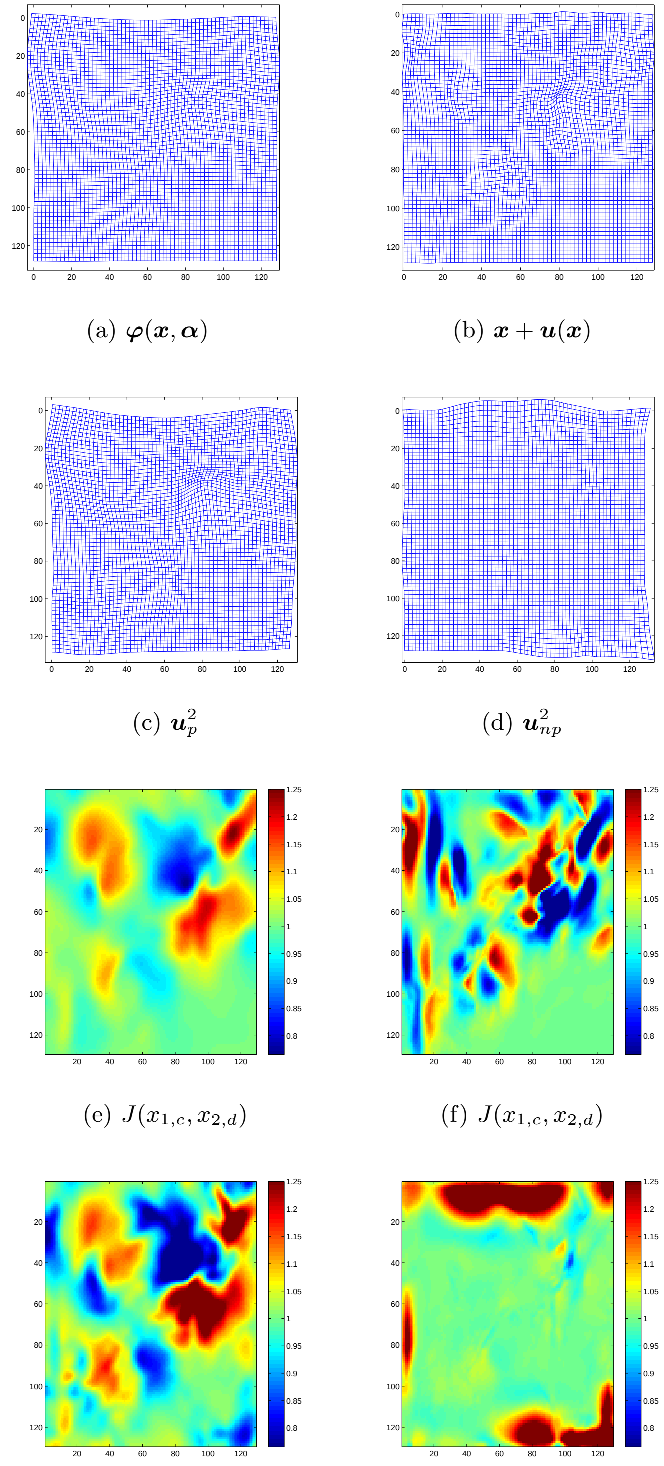
The deformation field and the determinant of the Jacobian matrix for all methods are shown in Figure 4.9. Clearly, the transformations are smooth and locally plausible. We also calculate MSE and the dice metric for Test 4 as shown in Table 4.3. Clearly M3 outperforms the individual methods.

Measure	T	M1	M2	M3
Mean squared error (MSE)	0.0203	0.0014	0.0108	0.00004
Dice metric	0.6667	0.9588	0.7900	0.9888

Table 4.3: MSE, and the dice metric for Test 4 before registration, and after registration using M1, M2 and M3. MSE is decreasing for all three models with the lowest value given by M3. The dice metrics are increasing for all models where the highest value is given by M3. Our method M3 outperforms the individual methods.

4.7 Conclusion

A pair of given images can be registered using either a parametric image registration model or a non-parametric image registration model. Both models have advantages and disadvantages: a parametric model is fast and easy to solve as it only involves a relatively small number of parameters; but most of the models tend to exclude the fine details of images. While a non-parametric model is slow and expensive to solve, it is able to capture fine details. We propose a decomposition framework consisting of both parametric and non-parametric components of an optimal transformation that consists both global and local transformations. In particular, we combine cubic B-spline FFD method with linear curvature as an example for the general framework. The resulting decomposition model is robust and relatively fast to implement in comparison with the individual models. Regularisation parameters are robustly obtained from multi-resolution ideas. In comparison with a non-parametric model equipped with the FFD parametric method as pre-registration, our algorithm works by alternating between these two models and outperforms either method individually.



(a) $\boldsymbol{\varphi}(\boldsymbol{x}, \boldsymbol{\alpha})$

(b) $\boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x})$

(c) $\boldsymbol{u}_p^2$

(d) $\boldsymbol{u}_{np}^2$

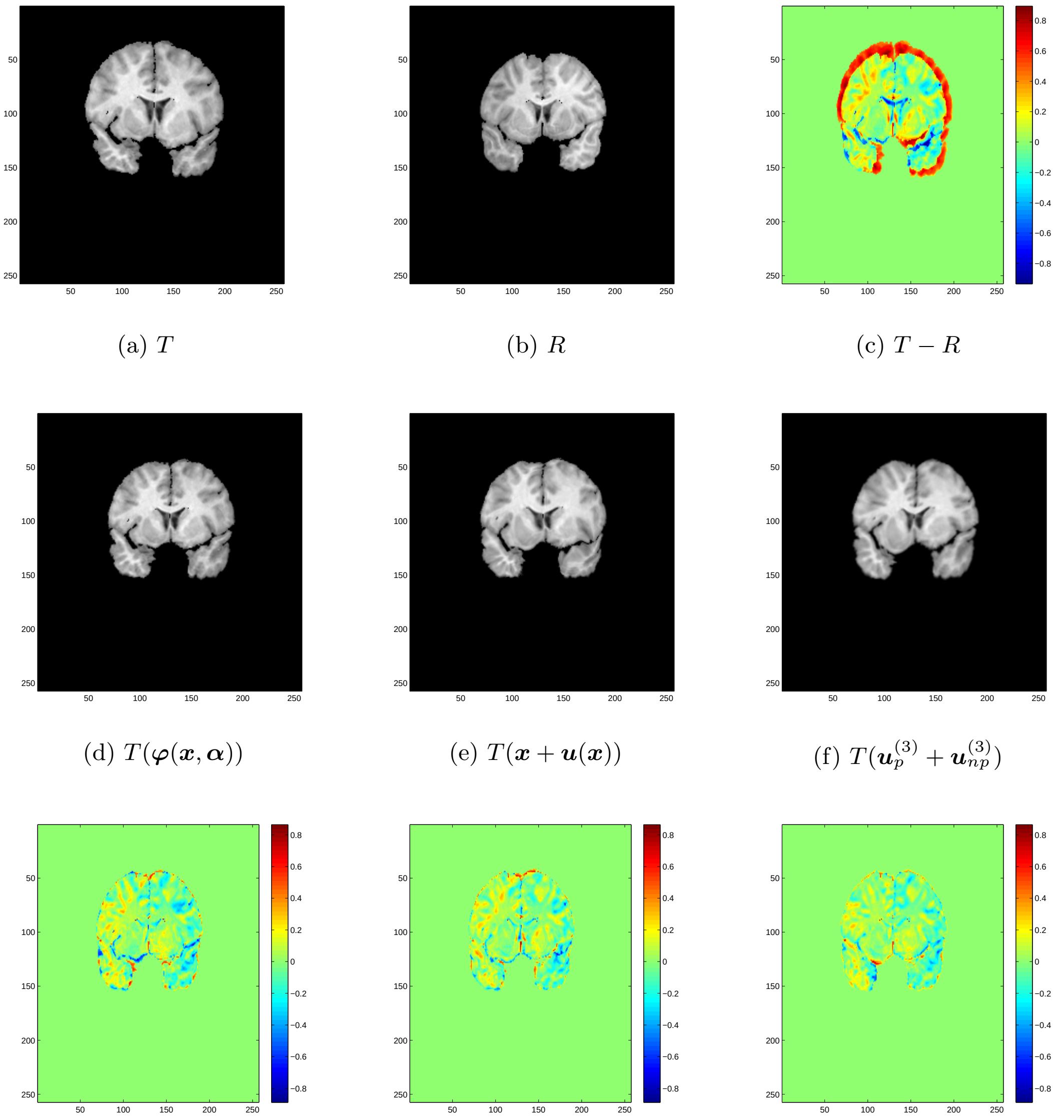
(e) $J(x_{1,c}, x_{2,d})$

(f) $J(x_{1,c}, x_{2,d})$

(g) $J(x_{1,c}, x_{2,d})$

(h) $J(x_{1,c}, x_{2,d})$

Figure 4.5: First to second row and left to right: deformation field applied to the regular grid for Test 2 using $\mathrm{M1}, \mathrm{M2}, \mathrm{M3}$ after the parametric part and M3 after the non-parametric part. Third to fourth row and left to right: the values of the determinant of the Jacobian matrix for the corresponding deformation on the top row. It is clear that the determinant of the Jacobian matrix is positive everywhere.



(a) T

(b) R

(c) $T - R$

(d) $T(\boldsymbol{\varphi}(\boldsymbol{x}, \boldsymbol{\alpha}))$

(e) $T(\boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x}))$

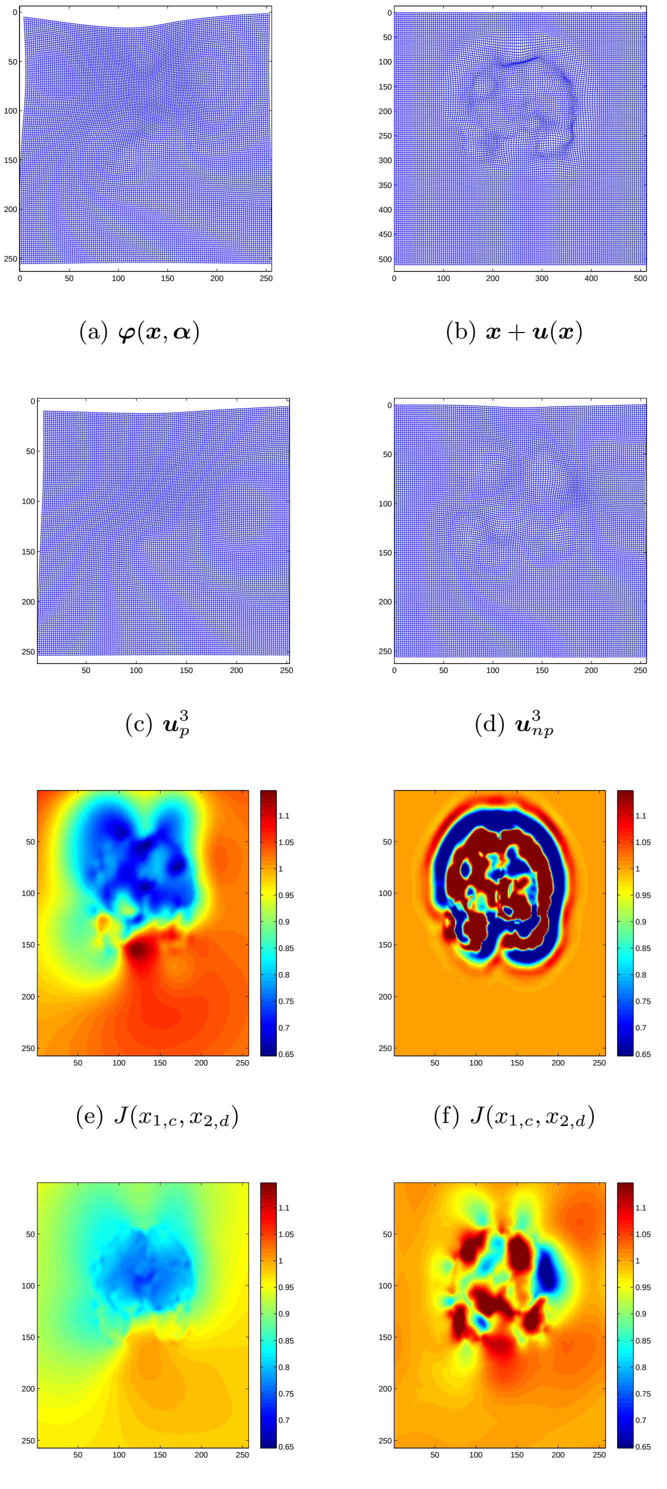
(f) $T(\boldsymbol{u}_p^{(3)} + \boldsymbol{u}_{np}^{(3)})$

(g) $\varepsilon = 0.2323$

(h) $\varepsilon = 0.1914$

(i) $\varepsilon = 0.1590$

Figure 4.6: First to second row and left to right: template, reference and the difference between the template and reference images. Middle row and left to right, are the results of Test 3 using M1, M2 and M3. Bottom row is the differences of the transform template images (middle row) and reference image. The best result is given by M3 where we can see that the method gives the smallest error as depicted on the bottom row.



(a) $\boldsymbol{\varphi}(\boldsymbol{x}, \boldsymbol{\alpha})$

(b) $\boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x})$

(c) $\boldsymbol{u}_p^3$

(d) $\boldsymbol{u}_{np}^3$

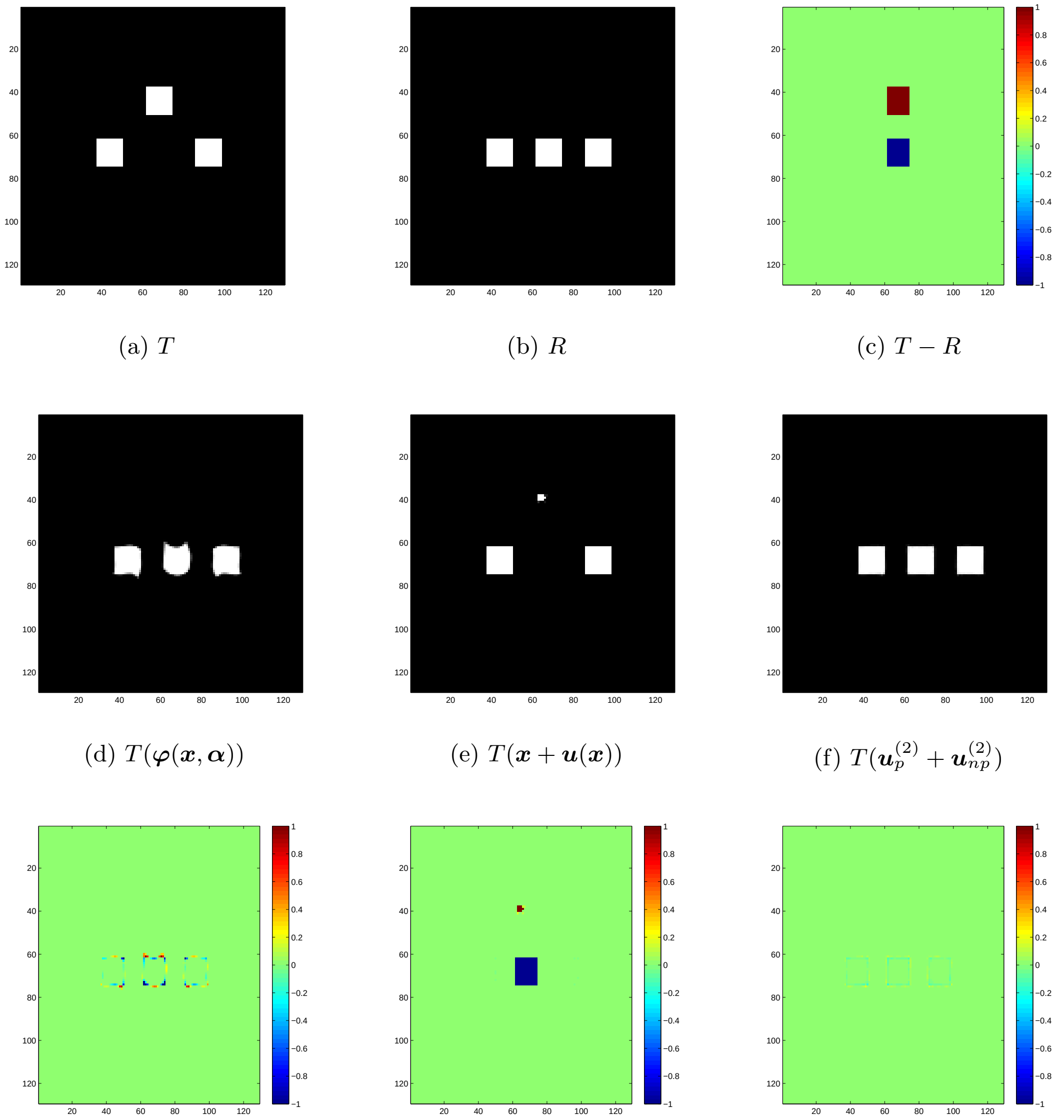
(e) $J(x_{1,c}, x_{2,d})$

(f) $J(x_{1,c}, x_{2,d})$

(g) $J(x_{1,c}, x_{2,d})$

(h) $J(x_{1,c}, x_{2,d})$

Figure 4.7: Top row and left to right: deformation field applied to the regular grid for Test 3 using $\mathrm{M1}, \mathrm{M2}, \mathrm{M3}$ after the parametric part and M3 after the non-parametric part. Bottom row and left to right: the values of the determinant of the Jacobian matrix for the corresponding deformation on the top row. It is clear that the determinant of the Jacobian matrix is positive everywhere.

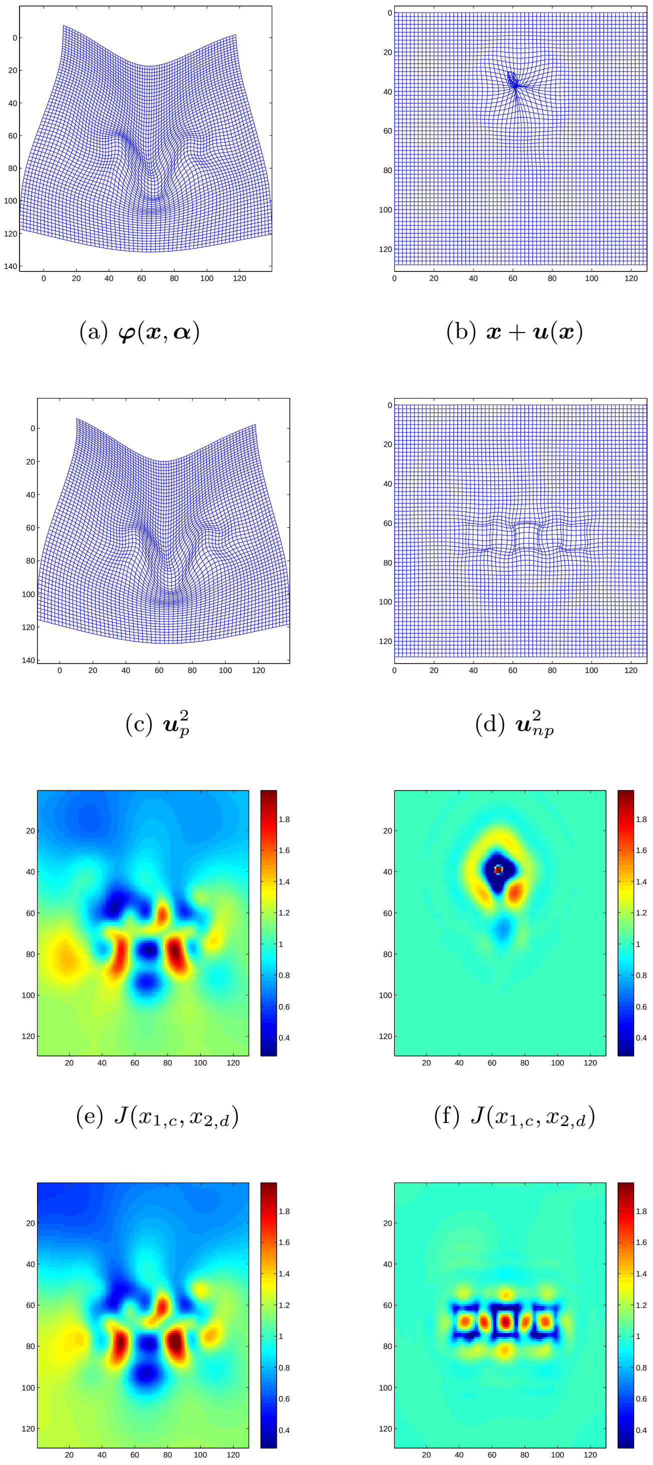


(a) T (b) R (c) $T - R$

(d) $T(\boldsymbol{\varphi}(\boldsymbol{x}, \boldsymbol{\alpha}))$ (e) $T(\boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x}))$ (f) $T(\boldsymbol{u}_p^{(2)} + \boldsymbol{u}_{np}^{(2)})$

(g) $\varepsilon = 0.0680$ (h) $\varepsilon = 0.5299$ (i) $\varepsilon = 0.0020$

Figure 4.8: Top row and left to right: template, reference and the difference between the template and reference images. Middle row and left to right: transformed template using M1, M2 and M3. Bottom row: the respective differences between the transformed template with the reference images. The corners of the boxes are well captured with M3 compared to M1.



(a) $\boldsymbol{\varphi}(\boldsymbol{x}, \boldsymbol{\alpha})$

(b) $\boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x})$

(c) $\boldsymbol{u}_p^2$

(d) $\boldsymbol{u}_{np}^2$

(e) $J(x_{1,c}, x_{2,d})$

(f) $J(x_{1,c}, x_{2,d})$

(g) $J(x_{1,c}, x_{2,d})$

(h) $J(x_{1,c}, x_{2,d})$

Figure 4.9: First to second row and left to right: deformation field applied to the regular grid for Test 4 using M1, M2, M3 after the parametric part and M3 after the non-parametric part. Third to fourth row and left to right: the values of the determinant of the Jacobian matrix for the corresponding deformation on the top row. It is clear that the determinant of the Jacobian matrix is positive everywhere.

Chapter 5

Multi-modality Image Registration using the Decomposition Model

So far, we have considered a decomposition model combining parametric and non-parametric deformation in Chapter 4. Both categories of model have been introduced in Chapter 3. In this chapter, we extend the decomposition model of parametric and non-parametric deformation which we developed in Chapter 4 for application to multi-modality images. In this case, the reference and template images come from different imaging modalities. For example, the reference image may be a computer tomography (CT) scan, which is useful for the quantification of cancerous tissues for the dose calculation in treatment planning, and the template image may be a magnetic resonance (MRI) image which is much better for the visualisation of soft tissues compared to the CT scanner. Given the very different resulting images, the intensity values are not directly comparable and so the use of traditional similarity measures such as the sum of the squared difference is no longer valid. We explore two similarity measures for multi-modality images given by mutual information and the normalised gradient field in Section 5.2 and 5.3 respectively. We introduce the decomposition model for multi-modality images in Section 5.4 and an alternating minimisation method to solve the model in Section 5.5. We use three sets of experiments in order to evaluate the benefit of these two similarity measures with the decomposition model to show the advantages of the normalised gradient field over mutual information with the decomposition model.

5.1 Introduction

The broad range of imaging machinery used in medical applications makes the registration of images from different modalities a challenging task. It is an open and active area of research because, despite the visual difference, the alignment of multi-modal images are complementary to each other. One of the most common applications of multi-modality image registration is in the process of the detection of breast cancer.

Several modalities such as MRI, mammography and ultrasound are combined in order to have an accurate measure of the cancerous tissues. The registration process has to deal with not only the geometric distortion caused by patients' movements but with intensity distortion such as the bias field effect which commonly appears in MRI. In addition, what makes the task more difficult is that there is no functional relationship between the intensity values of corresponding objects in different images. One of the remedies is to use the landmark registration method where clinicians identify several corresponding feature points in images resulting from the different modalities. However, this particular approach is time consuming, requires an expert to extract the points and there exists the possibility of mismatching the points which can result in inaccurate alignment of the images.

One of the commonly used similarity measures for multi-modality images is mutual information (MI). It was first introduced independently by Maes et al. in [64] and Viola and Wells in [96] and there is an assumption made based on the image information. This particular measure aims to find a statistical intensity relationship between the reference and template images. When two images are aligned, the amount of shared information is maximised. It has been successfully applied to rigid and affine image registration tasks. See [75, 77, 86] for more details. For non-rigid image registration, the best implementation of MI is not trivial because it is a global measure, therefore its local estimation is difficult and using MI as the distance measure increases the non-convexity of registration problem [37].

Real images are often distorted by spatially varying intensity inhomogeneities. As such, MRI images are affected by additive or multiplicative bias fields. Image registration models based on mutual information are at a disadvantage with the appearance of the bias field. In [33], the authors propose an alternative measure known as the normalised gradient field (NGF), a novel similarity measure for multi-modality image registration which is more reliable and robust than MI. NGF is based on the alignment of the edges in the reference and template images. In [44], NGF is used to register dynamic contrast enhanced (DCE)-MRI using the linear elastic image registration model.

In this chapter, we extend the decomposition model [49] for multi-modality images using MI and NGF. The decomposition model is based on combining parametric and non-parametric models where we particularly choose the cubic B-spline and linear curvature models by Fischer and Modersitzki [24, 25]. First, we introduce the mathematical background for mutual information and the normalised gradient field as distance measures for image registration. Second, we review the decomposition model of parametric and non-parametric image registration using MI and NGF. Third, we present the numerical algorithm to solve these two models. Then, we present numerical tests and finally, we conclude the two models.

5.2 Mutual Information

In this section, we introduce the mathematical background for mutual information as a distance measure in image registration. We recall the mathematical setting for image registration followed by the definitions of entropy and mutual information. Then, we derive the discrete mutual information and the gradient of this particular measure for optimisation purposes later.

Mathematical Preliminaries for Image Registration: Assume that we are given two images, the reference R and template T, which are compactly bounded and supported operators $T, R : \mathbb{R}^2 \to \mathbb{R}^+$. The image domain is denoted as $\Omega^h = [0, N_1] \times [0, N_2]$ and the pixel location is given by

$$\boldsymbol{x}_{i,j} = (x_{1,i}, x_{2,j}) \in \Omega^h, \quad 0 \leq i < N_1, \quad 0 \leq j < N_2 \tag{5.1}$$

where $x_{1,i} = ih_1$, $x_{2,j} = jh_2$, h_1 and h_2 are the width and height respectively of each pixel and images R and T are of size $N_1 \times N_2$. For ease of computation we use $h = h_1 = h_2$ and $N = N_1 = N_2$. We aim to align T and R such that the transform template image $T(\boldsymbol{\varphi})$ is aligned geometrically with R. The transformation $\boldsymbol{\varphi}$ is a vector valued function where $\boldsymbol{\varphi} : \mathbb{R}^2 \to \mathbb{R}^2$.

Entropy: The mutual information of two images T and R is a measure of how much information we know about the image T with the knowledge of the second image R. The probability distribution of T, $p_T(t)$ is defined as the number of pixels in image T that have pixel value t normalised by N^2

$$p_T(t) = \frac{1}{N^2}\Big(\# \text{ of } \ t_i = t\Big), i = 1, \dots N^2, \tag{5.2}$$

and similarly for image R

$$p_R(r) = \frac{1}{N^2}\Big(\# \text{ of } \ r_i = r\Big), i = 1, \dots N^2. \tag{5.3}$$

The ability of images to convey information can be measured in bits using Shannon entropy. The information content of a single event is proportional to the log of the inverse of the probability of an event. It is weighted by the probability that the event occurs and summed over all events to give the total information content known as entropy. We define entropy as the amount of uncertainty and the Shannon entropy for a random variable $X = \{x_1, x_2, ..., x_n\}$ is defined as

$$H(X) = \sum_i p_i \log \frac{1}{p_i} = -\sum_i p_i \log p_i = -\mathbb{E}\left[\log p_i\right] \tag{5.4}$$

where p_i are the probabilities of the variable x_i that occur in X. Similarly for T and R we have,

$$H(T) = -\sum_i p_T(t_i) \log p_T(t_i), \ \ H(R) \ \ = -\sum_i p_R(r_i) \log p_R(r_i). \tag{5.5}$$

Example 5.2.1 *Suppose we have a random variable X such that*

$$X = \begin{cases} 1, & \textit{with probability } p; \\ 2, & \textit{with probability } 1-p. \end{cases} \tag{5.6}$$

Then the entropy for X is given by

$$H(X) = -p \log p - (1-p)\log(1-p). \tag{5.7}$$

We can see that the entropy is not dependent on the value that the random variable takes (either 1 or 2 in this case), but it only depends on the probability distribution.

Entropy has the following properties:

1. Non-negativity: $H(X) \geq 0$, entropy is always non-negative. $H(X) = 0$ if and only if X is deterministic.

2. Chain rule: The decomposition of entropy is as follows

$$H(X_1, X_2, \ldots) = \sum_{i=1}^{n} H(X_i|X^{i-1}), \tag{5.8}$$

   where $X^{i-1} = \{X_1, X_2, \ldots, X_{i-1}\}$ and $H(X_i|X^{i-1})$ is defined as the conditional entropy.

3. Monotonicity: Conditioning always reduces entropy

$$H(X|Y) \leq H(X). \tag{5.9}$$

4. Maximum entropy: Let χ be the set from which the random variable X takes its values, then

$$H(X) \leq \log|\chi|. \tag{5.10}$$

   The above bound is achieved when X is uniformly distributed.

5. Non-increasing under functions: Let X be a random variable and let $g(X)$ be some deterministic function of X. We have

$$H(X) \geq H(g(X)). \tag{5.11}$$

The joint probability distribution of T and R denoted as $p_{T,R}(t,r)$ is calculated as the number of times out of N that a pixel in T contains t and the same pixel in R contains r normalised by the number of pixels,

$$p_{T,R}(t,r) = \frac{1}{N^2}\Big(\# \text{ of } \ t_i = t \ \text{ and } \ r_i = r\Big), i = 1, \ldots N^2. \tag{5.12}$$

Definition 5.2.2 ***Differential Entropy of a Continuous Random Variable.*** *The differential entropy for a continuous random variable X with probability density* *function (pdf) $f(x)$ is defined as*

$$h(X) = -\int f(x) \log f(x)\, \mathrm{d}x = -\mathbb{E}[\log(f(x))]. \tag{5.13}$$

For a pair of continuous random variables (X, Y) with joint pdf $f(x, y)$, the joint entropy is given by

$$h(X, Y) = -\int\int f(x, y) \log f(x, y)\, \mathrm{d}x\, \mathrm{d}y, \tag{5.14}$$

and the conditional entropy is

$$h(X|Y) = -\int\int f(x, y) \log f(x|y)\, \mathrm{d}x\, \mathrm{d}y. \tag{5.15}$$

Mutual Information: The probability distribution of the intensity values for all pixels in the image can be estimated using the histogram of the image or based on the Parzen window. We will use the latter approach to calculate the mutual information of T and R denoted as $MI(T, R)$ where

$$\begin{aligned} MI(T, R) &= \sum_{t,r} p_{T,R}(t, r) \log \frac{p_{T,R}(t, r)}{p_T(t) p_R(t)} \\ &= H(R) - H(R|T) \end{aligned} \tag{5.16}$$

where $H(R)$ is the entropy of the reference image and $H(R|T)$ is the conditional entropy, based on the conditional probabilities $p_{T,R}(r|t)$. When T and R are aligned, the amount of information they contain about each other is maximal. From equations (5.16) and (5.8) we have,

$$\begin{aligned} MI(T, R) &= H(R) - H(R|T) \\ &= H(R) - (H(T, R) - H(T)) \\ &= H(R) + H(T) - H(T, R). \end{aligned} \tag{5.17}$$

It is true that $0 \leq MI(T, R) \leq H(R)$, where $MI(T, R) = 0$ when R and T have no features in common and $MI(T, R) = H(R) = H(T)$ when $T = R$. Properties of the mutual information are:

1. Symmetry: $MI(T, R) = MI(R, T)$.
2. $MI(T, T) = H(T)$.
3. $MI(T, R) \leq H(T)$, $MI(T, R) \leq H(R)$. That is, the information which the images share can never be greater than the information in the images themselves.
4. $MI(T, R) \geq 0$, the uncertainty about T cannot be increased by learning about R.
5. $MI(T, R) = 0$ if and only if T and R are independent.

Definition 5.2.3 ***Mutual Information***. *The mutual information between two continuous random variables X, Y with joint pdf $f(x, y)$ is given by*

$$MI(X,Y) = \int\int f(x,y)\log\frac{f(x,y)}{f(x)f(y)}\,\mathrm{d}x\,\mathrm{d}y,$$

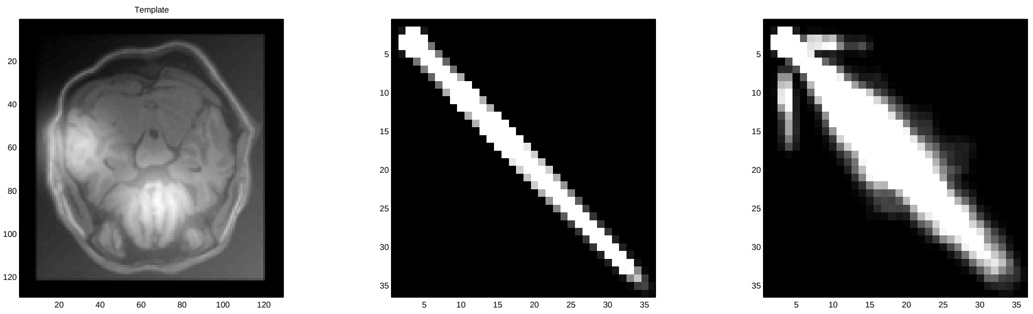
where $f(x)$ and $f(y)$ are the marginal pdf for X and Y respectively.

Definition 5.2.4 ***Mutual Information Distance Measure***. *The mutual information distance measure $\mathcal{D}^{MI}$ is defined by $\mathcal{D}^{MI} : \mathbb{R}^d \to \mathbb{R}$ where*

$$\mathcal{D}^{MI}(T,R;\boldsymbol{\varphi}) = -\int_{\mathbb{R}^2} p_{T(\boldsymbol{\varphi}),R}(t,r)\log\frac{p_{T(\boldsymbol{\varphi}),R}(t,r)}{p_{T(\boldsymbol{\varphi})}(t)p_R(r)}\,\mathrm{d}t\,\mathrm{d}r \tag{5.18}$$

where $\boldsymbol{\varphi}$ is the transformation $\boldsymbol{\varphi} : \mathbb{R}^2 \to \mathbb{R}^2$ for two dimensional images. Mutual information is a measure of similarity between given images. When $T(\boldsymbol{\varphi})$ and R are aligned the information contained is maximal.

Example 5.2.5 ***Mutual Information for Rotation Image*** *R. We have an image R from [71] and we find the second image T which is the translated version of R with a translation of 2 pixels in the x_1 direction. We plot the joint probability density of $p_{R,R}(r,r)$ and $p_{T,R}(t,r)$ which is given in Figures 5.1 (b) and (c) respectively. We can see that the density, $p_{R,R}(r,r)$ is very 'sharp' because $R = R$ and $p_{T,R}(t,r)$ is smeared out because $T \neq R$.*



(a) Image R.

(b) Joint probability distribution of image R with itself.

(c) Joint probability distribution of image R with T where T is the translated version of R.

Figure 5.1: Illustration of the image R and the joint probability density for R. (a) is the image R, (b) shows that the density is very 'sharp' because $R = R$ and (c) is smeared out because $T \neq R$.

Considering non-parametric image registration, we can model the transformation $\boldsymbol{\varphi}$ as

$$\boldsymbol{\varphi}(\boldsymbol{x}) = \boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x}). \tag{5.19}$$

Thus, we will derive the Gâteaux derivative of $\mathcal{D}^{\mathrm{MI}}$ with perturbation on $\boldsymbol{u}$,

$$\delta\mathcal{D}^{\mathrm{MI}}(\boldsymbol{u};\boldsymbol{\eta}) =< \nabla_{\boldsymbol{u}}\mathcal{D}^{\mathrm{MI}}, \boldsymbol{\eta} >_{\mathcal{H}} \tag{5.20}$$

where $< \cdot, \cdot >$ is the usual scalar product on a Hilbert space $\mathcal{H}$. Recall that

$$< \boldsymbol{u}, \boldsymbol{\eta} >_{\mathcal{H}} = \int_{\Omega} \boldsymbol{u}(\boldsymbol{x}) \cdot \boldsymbol{\eta}(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} = \int_{\Omega} < \boldsymbol{u}(\boldsymbol{x}), \boldsymbol{\eta}(\boldsymbol{x}) >_{\mathbb{R}^2} \mathrm{d}\boldsymbol{x}. \tag{5.21}$$

$\nabla_{\boldsymbol{u}} \mathcal{D}^{\mathrm{MI}}$ is given by

$$\nabla_{\boldsymbol{u}} \mathcal{D}^{\mathrm{MI}} = \frac{1}{|\Omega|} \left[k \star \frac{\partial L(R, T(\boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x})))}{\partial t} \right] \nabla_{\boldsymbol{u}} T(\boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x})) \tag{5.22}$$

where

$$\frac{\partial L(R, T(\boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x})))}{\partial t} = \frac{1}{p_{R,T(\boldsymbol{x}+\boldsymbol{u}(\boldsymbol{x}))(r,t)}} \frac{\partial p_{R,T(\boldsymbol{x}+\boldsymbol{u}(\boldsymbol{x}))(r,t)}}{\partial t} - \frac{1}{p_{T(\boldsymbol{x}+\boldsymbol{u}(\boldsymbol{x}))(t)}} \frac{\partial p_{T(\boldsymbol{x}+\boldsymbol{u}(\boldsymbol{x}))}(t)}{\partial t}, \tag{5.23}$$

and $\star$ denotes the convolution operator

$$[k_1 \star k_2](m, n) = \int_{\mathbb{R}^2} k_1(m - r, n - t) k_2(r, t) \, \mathrm{d}r \, \mathrm{d}t. \tag{5.24}$$

$|\Omega|$ denotes the area of the image domain $\Omega \subset \mathbb{R}^2$ and k is a smooth bidimensional density kernel used to estimate the joint probability density function of image R and $T(\boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x}))$, given by $p_{R,T(\boldsymbol{x}+\boldsymbol{u}(\boldsymbol{x}))}(r, t)$. Mutual information is not differentiable if we use the histogram based calculation to estimate the probability density function. Thus, we approximate the density with a smooth and differentiable approach using the Parzen window technique.

5.2.1 Parzen Windowing for Probability Estimation

Definition 5.2.6 ***Parzen Window***. *Let k be a function of unit integral $\left(\int_{-\infty}^{\infty} k(\tau) d\tau = 1\right)$. Let $\{t_i\}$ be a set of M samples of a random variable T with probability density function $p_T(t)$. The Parzen window estimate of $p_T(t)$ is given by*

$$p_T(t) = \frac{1}{M} \sum_{i=1}^{M} \frac{k\left((t - t_i)/\sigma\right)}{\sigma} \tag{5.25}$$

where $\sigma > 0$ is the width of the Parzen window kernel.

Parzen windowing is used to estimate the probability distribution using kernel estimation. k denotes the kernel function and σ is the predefined parameter for the kernel function. For example, in [67], k is a Gaussian kernel function given by

$$k(x; \sigma) = \frac{1}{\sqrt{2\pi}\sigma} \exp\Big(- \frac{x^2}{2\sigma^2} \Big). \tag{5.26}$$

We use the cubic spline kernel for estimating the joint probability density function as was done in [68] since the requirements for the kernel function are that it be a bell-shaped function, smooth, compactly supported and of integral one. For example, when

$\sigma = 1$, we have

$$k(x) = \begin{cases} (x+2)^3, & -2 \leq x < -1; \\ -x^3 - 2(x+1)^3 + 6(x+1), & -1 \leq x < 0; \\ x^3 + 2(x-1)^3 - 6(x-1), & 0 \leq x < 1; \\ (2-x)^2, & 1 \leq x < 2; \\ 0, & \text{elsewhere} \end{cases} \tag{5.27}$$

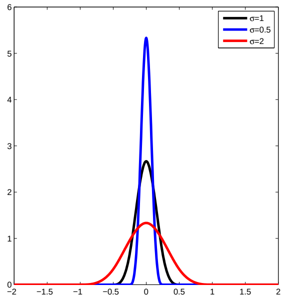


Figure 5.2: Illustration of the spline kernel $k^\sigma(x)$ for $\sigma = 0.5, 1, 2$.

and the derivative $k'(x)$ can be easily obtained from equation (5.27). The spline kernels for $\sigma = 0.5, 1, 2$ are illustrated in Figure 5.2. From Figure 5.2, we can see the effect of different values of σ. It controls the width and height of the spline kernel. From equation (5.26), if t_1 happens to cluster tightly with some sample that their Parzen windows $k((t_1 - t_i)/\sigma))/\sigma$ overlap often, then $p_T(t_1)$ will have large value. If at t_2, the samples are scattered and happen to be not dense, then only a few overlaps happen. Thus $p_T(t_2)$ will have a lower value. For ease of implementation, we will use

$$k^\sigma(x) = \frac{k(x/\sigma)}{\sigma}, \tag{5.28}$$

and we can write equation (5.25) as

$$p_T(t) = \frac{1}{M} \sum_{i=1}^{M} k^{\sigma_T}(t - t_i). \tag{5.29}$$

5.2.2 Discretisation of Mutual Information Distance Measure

Given images T and R with ranges $[t_0, t_N]$ and $[r_0, r_N]$ respectively where usually $t_0 = r_0 = 0$ and $t_n = r_n = 255$. Let h_t and h_r be defined as

$$h_t = \frac{t_n - t_0}{n_t} \text{ and } h_r = \frac{r_n - r_0}{n_r} \tag{5.30}$$

where n_t and n_r are predefined bin numbers, which are needed to discretise the intensity values of T and R. We have

$$t_i = t_0 + (i - 0.5)h_t \;\text{ and }\; r_j = r_0 + (j - 0.5)h_j, \;\; i = 0, \dots, n_t, \;\; j = 0, \dots, n_r. \tag{5.31}$$

The discrete set of intensity values in T and R are denoted by L_T and L_R respectively. We define the joint discrete Parzen histogram as

$$h(t, r) = \frac{1}{n_t n_r} \sum_{\boldsymbol{x} \in \Omega^h} k^{\sigma_T}(t - t_i(\boldsymbol{x})) k^{\sigma_R}(r - r_j(\boldsymbol{x})) \tag{5.32}$$

and the estimation of the joint probability distribution is given by

$$p(t, r) = p_{T,R}(t, r) = \frac{1}{\sum_{t \in L_T} \sum_{r \in L_R} h(t, r)} h(t, r). \tag{5.33}$$

We add a small tolerance ε to the argument of the logarithm to handle cases when we would have $0 \log 0$. Thus the joint entropy can be calculated as

$$H(T, R) = n_t n_r \sum_{i=1}^{n_t} \sum_{j=1}^{n_r} p_{i,j} \log(p_{i,j} + \varepsilon), \tag{5.34}$$

where $p_{i,j} = p(t_i, r_j)$ and the discretised marginal densities and entropy can be computed as

$$\begin{aligned} p_T(t_k) &= n_r \sum_{j=1}^{n_r} p_{k,j} \;\text{ and }\; H(T) = n_t \sum_{k=1}^{n_t} p_T(t_k) \log(p_T(t_k) + \varepsilon), \\ p_R(r_k) &= n_t \sum_{i=1}^{n_t} p_{i,k} \;\text{ and }\; H(R) = n_r \sum_{k=1}^{n_r} p_R(r_k) \log(p_R(r_k) + \varepsilon). \end{aligned} \tag{5.35}$$

5.3 Normalised Gradient Field

The sum of the squared distance measure assumes that the intensity values of R and the transformed template $T(\boldsymbol{\varphi})$ are equal. Meanwhile, mutual information makes the assumption that there is statistical dependency between $T(\boldsymbol{\varphi})$ and R. A trade-off between the SSD and MI is given by the normalised gradient field (NGF) distance measure which is based on the alignment of the edges of R and $T(\boldsymbol{\varphi})$. The features in $T(\boldsymbol{\varphi})$ and R can be identified by the intensity changes which are indicated by the gradient of $T(\boldsymbol{\varphi})$ and R. Since we are not interested in the magnitude of the gradient because there is no intensity relationship, we normalise the gradient with the magnitude of the gradient

$$\boldsymbol{n}T(\boldsymbol{x}) = \frac{\nabla T}{\|\nabla T\|_{\epsilon_T}}, \;\; \boldsymbol{n}R(\boldsymbol{x}) = \frac{\nabla R}{\|\nabla R\|_{\epsilon_R}} \tag{5.36}$$

where

$$\|\nabla T\|_{\epsilon_T} = \sqrt{\|\nabla T\|_2^2 + \epsilon_T^2}, \;\; \|\nabla R\|_{\epsilon_R} = \sqrt{\|\nabla R\|_2^2 + \epsilon_R^2}, \tag{5.37}$$

ϵ_T and ϵ_R are added into the calculation of NGF to overcome the problem when dividing by zero. Based on [33, 34], the values are given by

$$\epsilon_T = \frac{\eta}{V} \int_\Omega |\nabla T(\boldsymbol{x})| \, \mathrm{d}\Omega \tag{5.38}$$

where $\mathrm{d}\Omega = \mathrm{d}x_1 \, \mathrm{d}x_2$ and η is the estimated noise level and V is the volume of the domain Ω and similarly for ϵ_R. These parameters act as threshold values for the edges. When $\|\nabla T(\boldsymbol{x})\|_2^2 > \epsilon_T$, the feature is considered to be an edge and when $\|\nabla T(\boldsymbol{x})\|_2^2 < \epsilon_T$, it is considered as noise. For two vectors $\boldsymbol{a}$ and $\boldsymbol{b}$, the dot product is given by

$$\boldsymbol{a} \cdot \boldsymbol{b} = \|\boldsymbol{a}\|_2 \|\boldsymbol{b}\|_2 \cos\theta \tag{5.39}$$

where $\|\boldsymbol{a}\|_2$ and $\|\boldsymbol{b}\|_2$ are the norm of the vectors $\boldsymbol{a}$ and $\boldsymbol{b}$, and the cross product is

$$\|\boldsymbol{a} \times \boldsymbol{b}\|_2 = \|\boldsymbol{a}\|_2 \|\boldsymbol{b}\|_2 \sin\theta. \tag{5.40}$$

where θ is the angle between $\boldsymbol{a}$ and $\boldsymbol{b}$. Then,

$$\cos\theta = \frac{\boldsymbol{a} \cdot \boldsymbol{b}}{\|\boldsymbol{a}\|_2 \|\boldsymbol{b}\|_2} \text{ and } \sin\theta = \frac{\|\boldsymbol{a} \times \boldsymbol{b}\|_2}{\|\boldsymbol{a}\|_2 \|\boldsymbol{b}\|_2}. \tag{5.41}$$

Based on the dot and cross products of two vectors, we can defined the NGF similarity measure as

$$\mathcal{D}^{\mathrm{NGFc}}(T, R) = \frac{1}{2} \int_\Omega d^c(T, R) \, \mathrm{d}\Omega, \;\; d^c(T, R) = \|\boldsymbol{n}T(\boldsymbol{x}) \times \boldsymbol{n}R(\boldsymbol{x})\|_2^2 \tag{5.42}$$

and

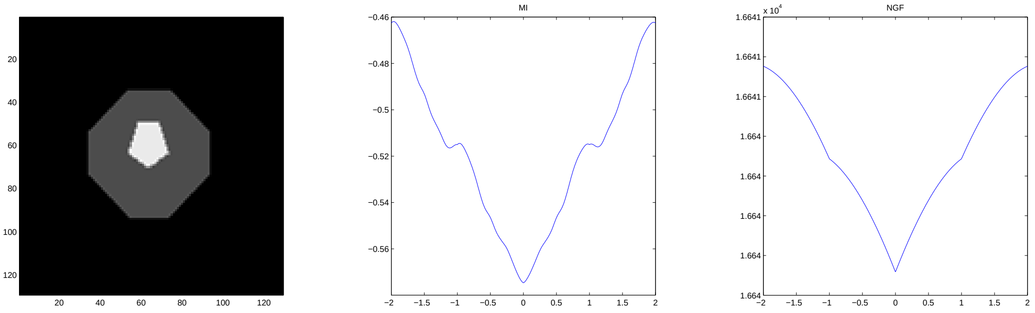
$$\mathcal{D}^{\mathrm{NGFd}}(T, R) = -\frac{1}{2} \int_\Omega d^d(T, R) \, \mathrm{d}\Omega, \;\; d^d(T, R) = (\boldsymbol{n}T(\boldsymbol{x}) \cdot \boldsymbol{n}R(\boldsymbol{x}))^2 \tag{5.43}$$

which are equivalent from an optimisation point of view [33, 34]. In this chapter we will use

$$\mathcal{D}^{\mathrm{NGF}}(T, R) = \int_\Omega 1 - \Big((\boldsymbol{n}T(\boldsymbol{x}))^T \boldsymbol{n}R(\boldsymbol{x}) \Big)^2 \, \mathrm{d}\Omega \tag{5.44}$$

as the normalised gradient field distance measure for image registration.

Example 5.3.1 *Consider that we have an image R as in Figure 5.3 (a). We transform R so that we have the template image T which is the translated version by 2 pixels in the x_1 direction. Then, we plot the two measures $\mathcal{D}^{MI}(T, R)$ and $\mathcal{D}^{NGF}(T, R)$ as shown in Figures 5.3 (b) and (c) respectively. We can see that $\mathcal{D}^{MI}(T, R)$ is highly non-convex since there are local and global minima. In contrast, $\mathcal{D}^{NGF}(T, R)$ has only one global minimum.*



(a) Reference, R

(b) $\mathcal{D}^{\mathrm{MI}}(T,R)$

(c) $\mathcal{D}^{\mathrm{NGF}}(T,R)$

Figure 5.3: Comparison of two distance measures. (a) is the reference image R, (b) is mutual information and (c) is normalised gradient field as the distance measures. We can see that (b) is highly non-convex. Thus, non-convexity of registration problems increases with mutual information as the distance measure.

5.3.1 Discretisation of Normalised Gradient Field

The NGF can be considered as the L_2 norm of r, the residual of the alignment of the normalised gradients of two images at a pixel position $\boldsymbol{x}$,

$$r^h(\boldsymbol{x}) = 1 - \left(\left(\boldsymbol{n}^h T(\boldsymbol{x}) \right)^T \boldsymbol{n}^h R(\boldsymbol{x}) \right)^2, \tag{5.45}$$

for discrete images T and R of size $N \times N$ using finite difference method. The images are discretised on a uniform mesh using vertex centred discretisation where $\boldsymbol{x}_{i,j}$ denotes the pixel position. It is also possible to use a non-uniform mesh with finite difference method. The gradient is calculated using

$$\partial_{x_1} T^h(\boldsymbol{x}_{i,j}) = \frac{T^h(\boldsymbol{x}_{i+1,j}) - T^h(\boldsymbol{x}_{i-1,j})}{2h}, \partial_{x_2} T^h(\boldsymbol{x}_{i,j}) = \frac{T^h(\boldsymbol{x}_{i,j+1}) - T^h(\boldsymbol{x}_{i,j-1})}{2h} \tag{5.46}$$

where the first order central finite difference scheme is used to approximate the first order derivatives. We use lexicographical ordering to reorder T and R into a row vector of size $N^2 \times 1$. Then, we have matrices G_{x_1} and G_{x_2} of sizes $N^2 \times N^2$

$$G_{x_1} = \frac{1}{2h^2} \begin{bmatrix} -1 & 1 & 0 & \dots & \dots \\ -1 & 0 & 1 & 0 & \dots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \dots & -1 & 0 & 1 \\ \vdots & \dots & 0 & -1 & 1 \end{bmatrix}, \ G_{x_2} = \frac{1}{2h^2} \begin{bmatrix} -1 & -1 & 0 & \dots & \dots \\ 1 & 0 & -1 & 0 & \dots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \dots & 1 & 0 & -1 \\ \vdots & \dots & 0 & 1 & 1 \end{bmatrix}, \tag{5.47}$$

that represent the discrete gradient operator in x_1 and x_2 respectively. We can calculate

$$E_{T,i} = \sqrt{(G_{x_1} T^h)_i + (G_{x_2} T^h)_i + \epsilon^2}, \ E_{R,i} = \sqrt{(G_{x_1} R^h)_i + (G_{x_2} R^h)_i + \epsilon^2}, i = 1, \dots, N^2. \tag{5.48}$$

Thus,

$$r_i = \left(\frac{(G_{x_1}T^h)_i}{E_{T_i}}\right)\left(\frac{(G_{x_1}R^h)_i}{E_{R_i}}\right) + \left(\frac{(G_{x_2}T^h)_i}{E_{T_i}}\right)\left(\frac{(G_{x_2}R^h)_i}{E_{R_i}}\right) \tag{5.49}$$

and

$$\mathcal{D}^{\mathrm{NGF}}(T^h, R^h) = h^2 \sum_{i=1}^{N^2} 1 - r_i^2. \tag{5.50}$$

5.3.2 Gradient of Normalised Gradient Field

In this subsection, we compute the discrete gradient of NGF.

Definition 5.3.2 ***The diag operator.*** *Let $A = diag[(b_0, ..., b_l)(d_0, ..., d_l), N] \in \mathbb{R}^{N\times N}$. Then the entries of the matrix A take values b_i on the diagonals with offset d_i,i.e*

$$a_{i,j} = \begin{cases} b_k, & \textit{if } i - j = d_k; \\ 0, & \textit{otherwise.} \end{cases} \tag{5.51}$$

Example 5.3.3 *A tridiagonal matrix of size 5×5 with entries b_0 on the lower diagonal, b_1 on the main diagonal and b_2 on the upper diagonal is given by*

$$diag[(b_0, b_1, b_2)(-1, 0, 1), 5] = \begin{bmatrix} b_1 & b_2 & 0 & 0 & 0 \\ b_0 & b_1 & b_2 & 0 & 0 \\ 0 & b_0 & b_1 & b_2 & 0 \\ 0 & 0 & b_0 & b_1 & b_2 \\ 0 & 0 & 0 & b_0 & b_1 \end{bmatrix}. \tag{5.52}$$

Let

$$\begin{aligned} \nabla T^h(\boldsymbol{\varphi}) &= \Big[\mathrm{diag}[(G_{x_1}T^h)(0), N^2], \mathrm{diag}[(G_{x_2}T^h)(0), N^2]\Big]^T, \\ \nabla R^h(\boldsymbol{\varphi}) &= \Big[\mathrm{diag}[(G_{x_1}R^h)(0), N^2], \mathrm{diag}[(G_{x_2}R^h)(0), N^2]\Big]^T, \\ r &= r_1 \odot r_2 = [r_{1,i} r_{2,i}]_i = \big(\mathrm{diag}[(r_1)(0), N^2]\big)\, r_2, \\ r_{1,i} &= (G_{x_1}T^h)_i (G_{x_1}R^h)_i + (G_{x_2}T^h)_i (G_{x_2}R^h)_i, \\ r_{2,i} &= \frac{1}{E_{T,i}E_{R,i}}. \end{aligned} \tag{5.53}$$

Then,

$$\begin{aligned} dr_1 &= \Big(\mathrm{diag}[(G_{x_1}R^h)(0), N^2]\Big)\, G_{x_1} + \Big(\mathrm{diag}[(G_{x_2}R^h)(0), N^2]\Big)\, G_{x_2}, \\ dr_2 &= \left(-\mathrm{diag}\left[\left(\frac{1}{(E_T)^3(E_R)}\right)(0), N^2\right]\right)\Big(\Big(\mathrm{diag}[(G_{x_1}T^h)(0), N^2]\Big) G_{x_1}, \\ &\quad + \Big(\mathrm{diag}[(G_{x_2}T^h)(0), N^2]\Big) G_{x_2}\Big), \\ dr &= dr_1 r_2 + r_1 dr_2, \end{aligned} \tag{5.54}$$

and

$$d\mathcal{D}^{\mathrm{NGF}}(T^h(\boldsymbol{\varphi}), R^h) = \big(2h^2 r^T dr\big)\, dT(\boldsymbol{\varphi}). \tag{5.55}$$

5.4 A Decomposition Model for Multi-modality Image Registration

A general framework for image registration is given by

$$\min_{\boldsymbol{\varphi}(\boldsymbol{x})} \mathcal{J}(T, R, \boldsymbol{\varphi}(\boldsymbol{x})) = \mathcal{D}(T, R, \boldsymbol{\varphi}(\boldsymbol{x})) + \gamma \mathcal{S}(\boldsymbol{\varphi}(\boldsymbol{x})) \tag{5.56}$$

where $\mathcal{D}(T, R, \boldsymbol{\varphi}(\boldsymbol{x}))$ is a similarity measure which quantifies the difference between T and R, $\mathcal{S}(\boldsymbol{\varphi}(\boldsymbol{x}))$ is the smoothness or regularisation term and γ is the regularisation parameter. The decomposition model of parametric and non-parametric image registration [49] decomposes the transformation field as

$$\boldsymbol{\varphi}(\boldsymbol{x}) = \boldsymbol{u}_p(\boldsymbol{x}) + \boldsymbol{u}_{np}(\boldsymbol{x}) \tag{5.57}$$

where $\boldsymbol{u}_p(\boldsymbol{x})$ and $\boldsymbol{u}_{np}(\boldsymbol{x})$ are the transformation field from parametric and non-parametric models respectively. In [49], the authors recommend this particular choice:

$$\boldsymbol{\varphi}(\boldsymbol{x}) = \boldsymbol{u}_{\text{cubic B-spline}} + \boldsymbol{u}_{\text{FMC}} \tag{5.58}$$

where $\boldsymbol{u}_{\text{cubic B-spline}}$ is the cubic B-spline based model [82] and $\boldsymbol{u}_{\text{FMC}}$ is the Fischer and Modersitzki linear curvature model [24, 25]. The functional minimisation problem for the decomposition model is given by

$$\begin{aligned} \min_{\boldsymbol{u}_{\text{cubic B-spline}}, \boldsymbol{u}_{\text{FMC}}} \mathcal{J}_{\gamma_p, \gamma_{np}} &= \mathcal{D}(T, R, \boldsymbol{u}_{\text{cubic B-spline}}, \boldsymbol{u}_{\text{FMC}}) \\ &\quad + \gamma_p \mathcal{S}^{\text{TP}}(\boldsymbol{u}_{\text{cubic B-spline}}) + \gamma_{np} \mathcal{S}^{\text{FMC}}(\boldsymbol{u}_{\text{FMC}}). \end{aligned} \tag{5.59}$$

The regularisation terms for equation (5.59) are as follows:

$$\begin{aligned} \mathcal{S}^{\text{TP}}(\boldsymbol{u}) &= \sum_{l=1}^{2} \int_\Omega (u_{l,x_1x_1})^2 + (2u_{l,x_1x_2})^2 + (u_{l,x_2x_2})^2)\, \mathrm{d}\Omega \\ \mathcal{S}^{\text{FMC}}(\boldsymbol{u}) &= \int_\Omega (\Delta u_1)^2 + (\Delta u_2)^2\, \mathrm{d}\Omega \end{aligned} \tag{5.60}$$

where $u_{l,x_1x_1} = \frac{\partial^2 u_l}{\partial x_1^2}$ and similarly for u_{l,x_1x_2} and u_{l,x_2x_2}. For multi-modality images, we will use two distance measures

$$\begin{aligned} \mathcal{D}^{\text{MI}}(T, R; \boldsymbol{\varphi}) &= \int_\Omega p_{T(\boldsymbol{\varphi}),R}(t, r) \log \frac{p_{T(\boldsymbol{\varphi}),R}(t, r)}{p_{T(\boldsymbol{\varphi})}(t) p_R(r)}\, \mathrm{d}t\, \mathrm{d}r, \\ \mathcal{D}^{\text{NGF}}(T, R; \boldsymbol{\varphi}) &= \int_\Omega 1 - \Big((\boldsymbol{n}T(\boldsymbol{\varphi}))^T \boldsymbol{n}R(\boldsymbol{x}) \Big)^2 \mathrm{d}\Omega \end{aligned} \tag{5.61}$$

as discussed in the previous section. Denote $\boldsymbol{u}_{\text{cubic B-spline}} = \boldsymbol{u}_p$ and $\boldsymbol{u}_{\text{FMC}} = \boldsymbol{u}_{np}$ for a more general framework. Since the parametric transformation fields $\boldsymbol{u}_p$ are dependent on a certain parameter of $\boldsymbol{\alpha}$, we will minimise (5.59) with respect to the parameter $\boldsymbol{\alpha}$.

5.5 Alternating Minimisation for the Decomposition Model

Problem (5.59) can be solved by alternating minimisation of two subproblems

$$\min_{\boldsymbol{\alpha}} \mathcal{J}^{I}_{\gamma_p} = \mathcal{D}(T, R, \boldsymbol{u}_p(\boldsymbol{x}, \boldsymbol{\alpha}), \boldsymbol{u}_{np}) + \gamma_p \mathcal{S}^{\mathrm{TP}}(\boldsymbol{u}_p(\boldsymbol{x}, \boldsymbol{\alpha})) \tag{5.62}$$

and

$$\min_{\boldsymbol{u}} \mathcal{J}^{II}_{\gamma_p} = \mathcal{D}(T, R, \boldsymbol{u}_p(\boldsymbol{x}, \boldsymbol{\alpha}), \boldsymbol{u}_{np}) + \gamma_{np} \mathcal{S}^{\mathrm{FMC}}(\boldsymbol{u}_{np}(\boldsymbol{x})). \tag{5.63}$$

We use the discretise then optimise scheme for numerical solutions. At the kth iteration, the alternate updates are done as follows:

Fixing $\boldsymbol{u}^{(k)}_{np}(\boldsymbol{x})$, we solve problem (5.62):

$$\begin{aligned}
\boldsymbol{u}^{(k+1)}_{p}(\boldsymbol{x}, \boldsymbol{\alpha}) \leftarrow \min_{\boldsymbol{\alpha}} \mathcal{J}^{I}_{\gamma_p} &= \sum_{i,j=0}^{N-1} \mathcal{D}(T(\boldsymbol{u}_p(\boldsymbol{x}_{i,j}, \boldsymbol{\alpha}) + \boldsymbol{u}_{np}(\boldsymbol{x}_{i,j})), R(\boldsymbol{x}_{i,j})) \\
&+ \gamma_p \mathcal{S}^{\mathrm{TP}}(\boldsymbol{u}_p(\boldsymbol{x}_{i,j}, \boldsymbol{\alpha})) \\
&= \sum_{i,j=0}^{N-1} \mathcal{D}(T(\boldsymbol{u}_p(\boldsymbol{x}_{i,j}, \boldsymbol{\alpha}) + \boldsymbol{u}_{np}(\boldsymbol{x}_{i,j})), R(\boldsymbol{x}_{i,j})) \\
&+ \gamma_p \sum_{l=1}^{2} \sum_{i,j=0}^{N-1} \left(u_{p,l,x_1x_1}(\boldsymbol{x}_{i,j}, \alpha)\right)^2 + 2\left(u_{p,l,x_1x_2}(\boldsymbol{x}_{i,j}, \alpha)\right)^2 \\
&+ \left(u_{p,l,x_2x_2}(\boldsymbol{x}_{i,j}, \alpha)\right)^2 .
\end{aligned} \tag{5.64}$$

Fixing $\boldsymbol{u}^{(k+1)}_{p}(\boldsymbol{x}, \boldsymbol{\alpha})$, we solve problem (5.63):

$$\begin{aligned}
\boldsymbol{u}^{(k+1)}_{np}(\boldsymbol{x}) \leftarrow \min_{\boldsymbol{u}_{np}(\boldsymbol{x})} \mathcal{J}^{I}_{\gamma_{np}} &= \sum_{i,j=0}^{N-1} \mathcal{D}(T(\boldsymbol{u}_p(\boldsymbol{x}_{i,j}, \boldsymbol{\alpha}) + \boldsymbol{u}_{np}(\boldsymbol{x}_{i,j})), R(\boldsymbol{x}_{i,j})) \\
&+ \gamma_{np} \mathcal{S}^{\mathrm{FMC}}(\boldsymbol{u}_{np}(\boldsymbol{x}_{i,j})) \\
&= \sum_{i,j=0}^{N-1} \mathcal{D}(T(\boldsymbol{u}_p(\boldsymbol{x}_{i,j}, \boldsymbol{\alpha}) + \boldsymbol{u}_{np}(\boldsymbol{x}_{i,j})), R(\boldsymbol{x}_{i,j})) \\
&+ \gamma_{np} \sum_{l=1}^{2} \sum_{i,j=0}^{N-1} \Big[-4\boldsymbol{u}_{np,l}(\boldsymbol{x}_{i,j}) + \boldsymbol{u}_{np,l}(\boldsymbol{x}_{i+1,j}) \\
&+ \boldsymbol{u}_{np,l}(\boldsymbol{x}_{i-1,j}) + \boldsymbol{u}_{np,l}(\boldsymbol{x}_{i,j+1}) + \boldsymbol{u}_{np,l}(\boldsymbol{x}_{i,j-1}) \Big]^2 .
\end{aligned} \tag{5.65}$$

To solve (5.64), $\boldsymbol{u}_p(\boldsymbol{x}, \boldsymbol{\alpha})$ are defined by the cubic B-splines with coefficient $\boldsymbol{\alpha}$:

$$u_{p,1}(\boldsymbol{x}, \alpha_1) = \sum_{i,j} B_{i,j}(\boldsymbol{x})\alpha_{1,i,j}, u_{p,2}(\boldsymbol{x}, \alpha_2) = \sum_{i,j} B_{i,j}(\boldsymbol{x})\alpha_{2,i,j} \tag{5.66}$$

where $\boldsymbol{\alpha} = (\alpha_1, \alpha_2)^T$, is the lattice of control points which are the parameters for the cubic B-spline model. $B_{i,j}(\boldsymbol{x})$ are given by

$$B_{i,j}(\boldsymbol{x}) = \begin{cases} B_l(\mu)B_m(\nu), & i = \widetilde{i} + l, j = \widetilde{j} + m, \text{ for } l, m = 0, 1, 2, 3; \\ 0, & \text{elsewhere,} \end{cases} \tag{5.67}$$

where $B_l(\mu)$ and $B_m(\nu)$ are cubic B-spline basis functions as follows:

$$\begin{aligned} B_0(\mu) &= (1-\mu)^3/6, \\ B_1(\mu) &= (3\mu^3 - 6\mu^2 + 4)/6, \\ B_2(\mu) &= (-3\mu^3 + 3\mu^2 + 3\mu + 1)/6, \\ B_3(\mu) &= \mu^3/6, \end{aligned} \tag{5.68}$$

$\widetilde{i} = \lfloor \frac{x_1}{\delta x_1} \rfloor - 1, \widetilde{j} = \lfloor \frac{x_2}{\delta x_2} \rfloor - 1$, $\mu = \frac{x_1}{\delta x_1} - \lfloor \frac{x_1}{\delta x_1} \rfloor$ and $\nu = \frac{x_2}{\delta x_2} - \lfloor \frac{x_2}{\delta x_2} \rfloor$. The spacing of the control points δx_1 and δx_2 are the predefined parameters as illustrated in details in Chapter 4.

Next is to solve (5.65) iteratively using the LBFGS method. An alternating minimisation method for model (5.59) is given in Algorithm 8.

```
Algorithm 8 Alternating minimisation algorithm for the decomposition model.
$(\gamma_p, \gamma_{np}, \boldsymbol{u}_p(\boldsymbol{x}, \boldsymbol{\alpha}), \boldsymbol{u}_{np}(\boldsymbol{x})) \leftarrow \text{Decomposition}(T, R, \boldsymbol{u}_{np}^{(0,1)}, \boldsymbol{\alpha}^{(0,1)}, L, \gamma_0, \gamma_1, \tau, m, \boldsymbol{maxit}, \boldsymbol{TOL})$

1. Initialise $\gamma_p^{(0)} = \gamma_{p,1}, \gamma_{np}^{(0)} = \gamma_{np,1}$. Restrict $\boldsymbol{u}_{np}^{(0,1)}$ and $\boldsymbol{\alpha}^{(0,1)}$ to the coarsest grid $L$. $\boldsymbol{u}_{np}^{(0,L)} \leftarrow \text{Restrict}(\boldsymbol{u}_{np}^{(0,1)})$ and $\boldsymbol{\alpha}^{(0,L)} \leftarrow \text{Restrict}(\boldsymbol{\alpha}^{(0,1)})$.
2. For $k_3 = 0, 1, \ldots, \boldsymbol{maxit}$ : (Outer iteration for alternating minimisation).
    (a) Compute $\boldsymbol{\alpha}^{(k_3+1,1)}$ and $\gamma_p^{(k_3+1)}$ using equation (5.64) with $\boldsymbol{\alpha}^{(k_3,L)}$ as initial guess. $(\boldsymbol{u}_p^{(k_3+1,1)}(\boldsymbol{x}, \boldsymbol{\alpha}), \gamma_p^{(k_3+1)}) \leftarrow \text{MLIR}(\boldsymbol{\alpha}^{(k_3,L)}, T(\boldsymbol{u}_{np}^{(k_3,1)}(\boldsymbol{x})), R, L, \mathcal{J}_{\gamma_p}^{I}, \gamma_{p,0}, \gamma_{p,1}, \tau, m, 1, \boldsymbol{maxit}, \boldsymbol{TOL})$.
    (b) Restrict $\boldsymbol{u}_p^{(k_3+1,1)}$ at fine level to level $L$. $\boldsymbol{u}_p^{(k_3+1,L)} \leftarrow \text{Restrict}(\boldsymbol{u}_p^{(k_3+1,1)})$ and find the respective $\boldsymbol{\alpha}^{(k_3+1,L)}$ from $\boldsymbol{u}_p^{(k_3+1,L)}(\boldsymbol{x}, \boldsymbol{\alpha})$.
    (c) Compute $\boldsymbol{u}_{np}^{(k_3+1)}(\boldsymbol{x})$ and $\gamma_{np}^{(k_3+1)}$ using equation (5.65) with $\boldsymbol{u}_{np}^{(k_3,L)}(\boldsymbol{x})$ as initial guess. $(\boldsymbol{u}_{np}^{(k_3+1,1)}(\boldsymbol{x}), \gamma_{np}^{(k_3+1)}) \leftarrow \text{MLIR}(\boldsymbol{u}_{np}^{(k_3,L)}, T(\boldsymbol{u}_p^{(k_3+1,1)}(\boldsymbol{x}, \boldsymbol{\alpha})), R, L, \mathcal{J}_{\gamma_{np}}^{II}, \gamma_{np,0}, \gamma_{np,1}, \tau, m, 2, \boldsymbol{maxit}, \boldsymbol{TOL})$.
    (d) Restrict $\boldsymbol{u}_{np}^{(k_3+1,1)}(\boldsymbol{x})$ to $\boldsymbol{u}_{np}^{(k_3+1,L)}(\boldsymbol{x})$. $\boldsymbol{u}_{np}^{(k_3+1,L)}(\boldsymbol{x}) \leftarrow \text{Restrict}(\boldsymbol{u}_{np}^{(k_3+1,1)}(\boldsymbol{x}))$.
    (e) If $\|\gamma_p^{(k_3+1)} - \gamma_p^{(k_3)}\| < \boldsymbol{TOL}$, $\|\gamma_{np}^{(k_3+1)} - \gamma_{np}^{(k_3)}\| < \boldsymbol{TOL}$ or $\mathcal{J}^{\text{M3}}_{\gamma_p^{(k_3+1)}, \gamma_{np}^{(k_3+1)}}(\boldsymbol{u}_p^{(k_3+1)}(\boldsymbol{x}, \boldsymbol{\alpha}), \boldsymbol{u}_{np}^{(k_3+1)}(\boldsymbol{x})) > \mathcal{J}^{\text{M3}}_{\gamma_p^{(k_3)}, \gamma_{np}^{(k_3)}}(\boldsymbol{u}_p^{(k_3)}(\boldsymbol{x}, \boldsymbol{\alpha}), \boldsymbol{u}_{np}^{(k_3)}(\boldsymbol{x}))$ exit, else continue with the next iteration $k_3$.
3. End for. Accept $\gamma_p = \gamma_p^{(k_3+1)}, \gamma_{np} = \gamma_{np}^{(k_3+1)}$.
4. Compute the final registration algorithm on the finest level only. $(\boldsymbol{u}_p(\boldsymbol{x}, \boldsymbol{\alpha}), \boldsymbol{\alpha}) \leftarrow \text{LBFGS}(\gamma_p, \mathcal{J}_{\gamma_p}^{I}, \boldsymbol{\alpha}^{(k_3+1,1)}, m, \boldsymbol{maxit}, \boldsymbol{TOL})$.
5. Compute the final registration algorithm on the finest level only. $(\boldsymbol{u}_{np}(\boldsymbol{x}), \boldsymbol{u}_{np}(\boldsymbol{x})) \leftarrow \text{LBFGS}(\gamma_{np}, \mathcal{J}_{\gamma_{np}}^{II}, \boldsymbol{u}_{np}^{(k_3+1,1)}, m, \boldsymbol{maxit}, \boldsymbol{TOL})$.
6. Finish the iteration with $\gamma_p, \gamma_{np}, \boldsymbol{u}_p(\boldsymbol{x}, \boldsymbol{\alpha})$, and $\boldsymbol{u}_{np}(\boldsymbol{x})$.
```

We use the same algorithms as in Chapter 4 to solve model problems (5.62) and (5.63).

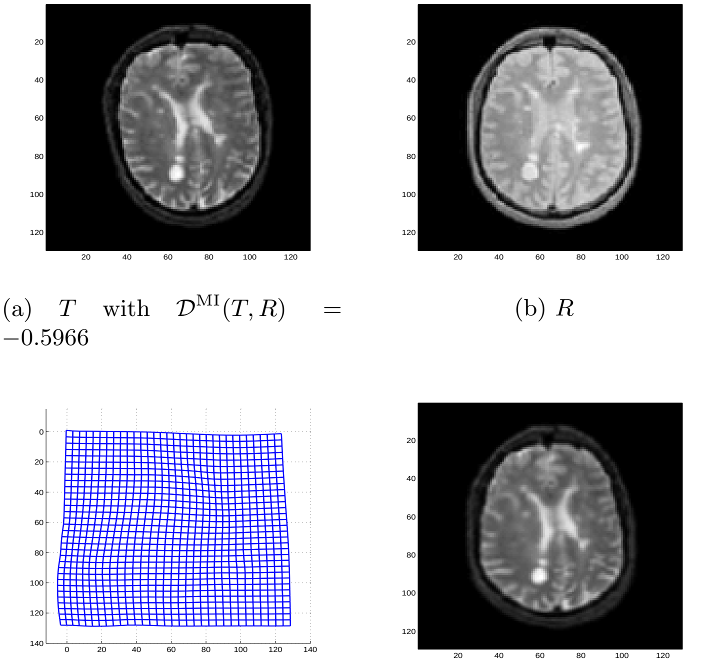
5.6 Numerical Results

We use three sets of tests to show the performance of the decomposition model for multi-modality images. In Test 1, we have a reference image from photon density weighted MRI and a template image which comes from T2-MRI. For Test 2, we have synthetic images from [18] to illustrate the type of images where the mutual information and decomposition models are at a disadvantage. We obtain a good result using the normalised gradient field and the decomposition model for Test 2. Meanwhile, for Test

3, we use images from [71] to illustrate the case where both models fail to deliver good registration results. Such images therefore require an alternative registration which we may provide.

5.6.1 Test 1: Photon Density Weighted MRI and T2-MRI

The results of the mutual information and decomposition models for Test 1 are shown in Figure 5.4. In Figure 5.4, we can observe that the decomposition model with mutual information as the distance measure is able to solve the problem of real medical image registration where the reference and template images are from photon density weighted MRI and T2-MRI respectively. We show the results for Test 1 using the normalised gradient field in Figure 5.5. The transformed template image for the normalised gradient field and the decomposition model is shown in Figure 5.5 (b). We have an acceptable level of the transformed template image where it appears to be similar to the reference image except at the middle right part of the brain.



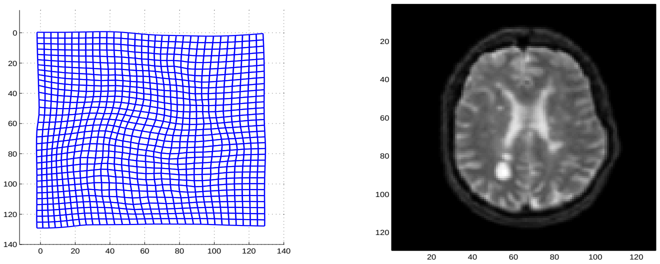
(a) T with $\mathcal{D}^{\mathrm{MI}}(T,R) = -0.5966$

(b) R

(c) $\boldsymbol{\varphi}(\boldsymbol{x})$

(d) $T(\boldsymbol{\varphi}(\boldsymbol{x}))$ with $\mathcal{D}^{\mathrm{MI}}(T(\boldsymbol{\varphi}(\boldsymbol{x})),R) = -0.8013$

Figure 5.4: Test 1: Results of mutual information as the distance measure with the decomposition model for multi-modality images. We can see that the model delivers a good alignment between the transformed template image in (d) and the reference image in (b).



(a) $\boldsymbol{\varphi}(\boldsymbol{x})$

(b) $T(\boldsymbol{\varphi}(\boldsymbol{x}))$ with $\mathcal{D}^{\mathrm{MI}}(T(\boldsymbol{\varphi}(\boldsymbol{x})), R) = -0.9709$

Figure 5.5: Test 1: Results of normalised gradient as the distance measure with the decomposition model for multi-modality images. The resulting transformed template in (b) is in alignment with the reference image except at the middle part of the brain. Smaller value of $\mathcal{D}^{\mathrm{MI}}(T(\boldsymbol{\varphi}(\boldsymbol{x})), R)$ in (b) than in Figure 5.4 (d) indicating higher similarity between the transformed template and the reference images.

5.6.2 Test 2: Synthetic Images

In Test 2, we aim to illustrate the type of images where mutual information and the decomposition model fail to deliver good registration results. Since mutual information uses the statistical dependency of the intensity values between the reference and template images, the model fails to register this type of problem because only the reference image has the square object inside the circle. However, we have a good result using normalised gradient field as shown in Figure 5.7.

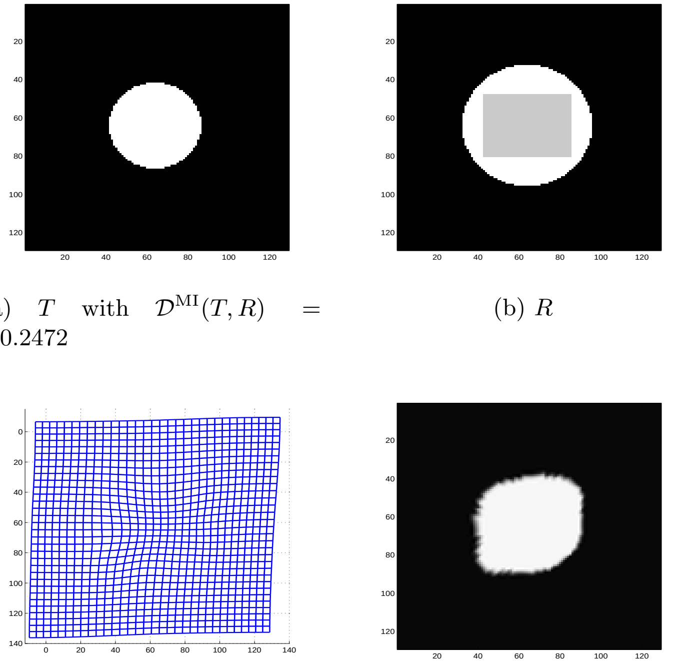
5.6.3 Test 3: Bias Field Registration

Bias fields or intensity inhomogeneities are common problems in medical image analysis where some part of the same object in the image appears to become darker than the rest of the object. It is a very common problem in MRI. In Figure 5.8, we show the results of mutual information with a strong bias field in the template image.

The normalised gradient field and the decomposition model also fail to register Test 3 as depicted in Figure 5.9. We can see that in the figure, the outermost boundary of the brain in the transformed template image is not aligned with the one in the reference image.

5.7 Conclusion

We have shown the extension of the decomposition model to multi-modality images using mutual information and the normalised gradient field as the distance measures. From the numerical tests, we can observe that the normalised gradient field and decomposition model work better than mutual information. However both models are at a disadvantage when there is a strong bias field in the images. We will address this in



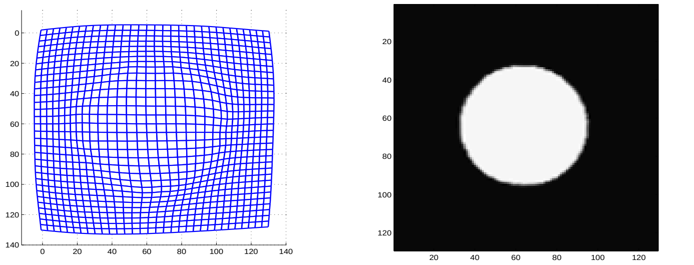
(a) T with $\mathcal{D}^{\mathrm{MI}}(T,R) = -0.2472$

(b) R

(c) $\boldsymbol{\varphi}(\boldsymbol{x})$

(d) $T(\boldsymbol{\varphi}(\boldsymbol{x}))$ with $\mathcal{D}^{\mathrm{MI}}(T(\boldsymbol{\varphi}(\boldsymbol{x})), R) = -0.3777$

Figure 5.6: Test 2: Results of mutual information as the distance measure with the decomposition model for multi-modality images. We can see that the model fails for the deformed circle in the template image in (a) due to the existence of the inner square in the reference image (b).

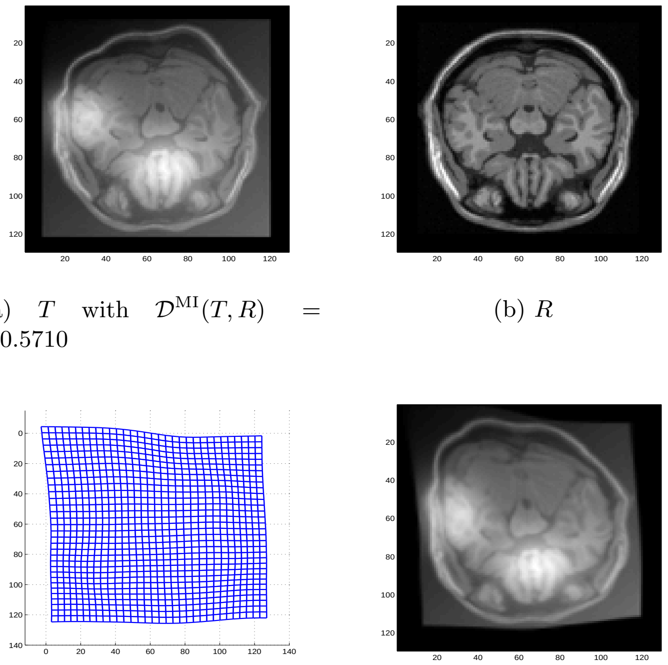


(a) $\boldsymbol{\varphi}(\boldsymbol{x})$

(b) $T(\boldsymbol{\varphi}(\boldsymbol{x}))$ with $\mathcal{D}^{\mathrm{MI}}(T(\boldsymbol{\varphi}(\boldsymbol{x})), R) = -0.4902$

Figure 5.7: Test 2: Results of normalised gradient as the distance measure with the decomposition model for multi-modality images. We can see the model is able to solve this particular problem. Smaller value of $\mathcal{D}^{\mathrm{MI}}(T(\boldsymbol{\varphi}(\boldsymbol{x})), R)$ in (b) than in Figure 5.6 (d) indicating higher similarity between the transformed template and the reference images.

the future work.



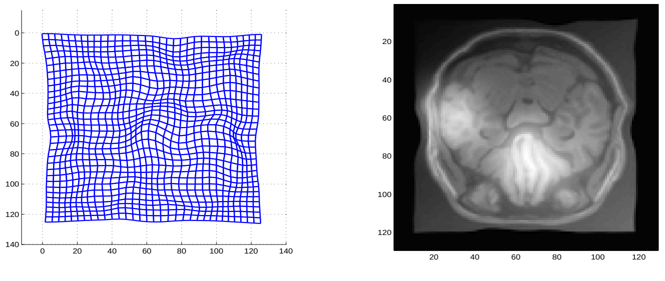
(a) T with $\mathcal{D}^{\mathrm{MI}}(T, R) = -0.5710$

(b) R

(c) $\boldsymbol{\varphi}(\boldsymbol{x})$

(d) $T(\boldsymbol{\varphi}(\boldsymbol{x}))$ with $\mathcal{D}^{\mathrm{MI}}(T(\boldsymbol{\varphi}(\boldsymbol{x})), R) = -0.5742$

Figure 5.8: Test 3: Results of mutual information as the distance measure with the decomposition model for multi-modality images. We can see that the model fails to register the template with the reference image due to the strong bias field in (a).



(a) $\boldsymbol{\varphi}(\boldsymbol{x})$

(b) $T(\boldsymbol{\varphi}(\boldsymbol{x}))$ with $\mathcal{D}^{\mathrm{MI}}(T(\boldsymbol{\varphi}(\boldsymbol{x})), R) = -0.6773$

Figure 5.9: Test 3: Results of normalised gradient field as the distance measure with the decomposition model for multi-modality images. We can see that the model fails to register the template with the reference image due to the strong bias field in (a). Smaller value of $\mathcal{D}^{\mathrm{MI}}(T(\boldsymbol{\varphi}(\boldsymbol{x})), R)$ in (b) than in Figure 5.8 (d) indicating higher similarity between the transformed template and the reference images.

Chapter 6

A Novel Variational Model for Image Registration using Gaussian Curvature

We have presented the first part of our main work which was the decomposition model for mono and multi modality image registration. We now present the second part of our main work which is a vital part of this project. In this chapter, we propose a novel variational model for image registration using Gaussian curvature as a regulariser. The model is motivated by the surface restoration work in geometric processing. An effective numerical solver is provided for the model using an augmented Lagrangian method. Numerical experiments show that the new model outperforms three competing models which are based on, respectively, the linear curvature, the mean curvature and the diffeomorphic demon models, in terms of robustness and accuracy.

6.1 Introduction

In previous work on non-parametric image registration, higher order regularisation models [26, 19] were found to be the most robust while the diffeomorphic demon model [93] offers the most physical transform in terms of (nearly) bijective mapping. Diffusion and total variation regularisation models based on first order derivatives are less complicated to implement but are at a disadvantage compared to higher order regularisation models based on second order derivatives due to two reasons. First, the former methods penalise rigid displacement. They cannot properly deal with transformations involving translation and rotation. Second, low order regularisation is less effective than high order regularisation in producing smooth transformations, which are important in several applications including medical imaging. The work of [24, 25, 26] proposed a high order regularisation model called as linear curvature, which is an approximation of the surface (mean) curvature and the model is invariant to affine registration. This work was later refined in [41, 39, 40] where a related approximation to the sum of the squares of the principal curvatures was suggested and higher order boundary conditions were recommended. Without any approximation to the mean curvature, the work in [19, 18] developed useful numerical algorithms for models based on nonlinear mean curvature regularisation and observed advantages over the linear curvature models for image registration. However the effect of mesh folding (bijective maps) was not considered. The diffeomorphic demon model [94] is widely used due to its use of bijective maps; but the bijection is not precisely imposed. Another useful idea of enforcing bijection, beyond the framework we consider, is via minimising the Beltrami coefficient which measures the distortion of the quasi-conformal map of registration transforms [56].

In this chapter, we propose a high order registration model based on Gaussian curvature and hope to achieve large and smooth deformation without mesh folding. Although the Gaussian curvature is closely related to the mean curvature, it turns out that our new model based on the Gaussian curvature is much better. The motivation of the proposed model comes from two sources. Firstly, we are inspired by the work of Elsey and Esedoglu [21] in geometry processing where the Gaussian curvature of the image surface is used in a variational formulation. The authors proposed the Gaussian curvature as a natural analogue of the total variation of the Rudin, Osher and Fatemi (ROF) model [78] for geometry processing. Aiming to generalise the ROF model to surface fairing, where the convex shapes in 3D have similar interpretation to the monotone functions in 1D problems for the ROF model, they showed that, based on the Gauss Bonnet theorem, the complete analogue of the total variation regularisation for surface fairing is the energy functional of the Gaussian curvature. Secondly, a very important fact pointed out in [21] is that the mean curvature of the surface is *not* a suitable choice for surface fairing because the model is not effective for preserving important features such as creases and corners on the surface (although the model is still effective for removing noise). Their claims are also supported by the work of [60] where the authors illustrated several advantages of Gaussian curvature over mean curvature and total variation in removing noise in 2D images. First, Gaussian curvature preserves important structures such as edges and corners. Second, only Gaussian curvature can preserve structures with low gradient. Third, the model is effective in removing noise on small scale features. Thus, we believe that Gaussian curvature is a more natural physical quantity of the surface than mean curvature. Here we investigate the potential of using Gaussian curvature to construct a high order regularisation model for non-parametric image registration of mono-modal images.

Below, we list three popular models selected for tests and comparisons.

Model LC. The first is the linear curvature model considered in [25, 26, 67, 24], where

$$\mathcal{S}^{\mathrm{LC}}(\boldsymbol{u}) = \int_{\Omega} \Big[(\Delta u_1)^2 + (\Delta u_2)^2 \Big] \,\mathrm{d}\Omega. \tag{6.1}$$

This term is an approximation of the surface curvature $\iota(u_l)$ through the mapping

$(x, y) \to (x, y, u_l(x, y)),\ l = 1, 2$, where

$$\iota(u_l) = \nabla \cdot \frac{\nabla u_l}{\sqrt{|\nabla u_l|^2 + 1}} \ \approx \ \Delta u_l \tag{6.2}$$

when $|\nabla u_l| \approx 0$. The Euler-Lagrange equation with $\mathcal{S}^{\text{LC}}$ as the regularisation term is given by a fourth order PDE

$$\gamma \Delta^2 \boldsymbol{u} + \boldsymbol{f}(\boldsymbol{u}) = 0 \tag{6.3}$$

with boundary conditions $\Delta u_l = 0, \nabla \Delta u_l \cdot \boldsymbol{n} = 0,\ l = 1, 2$ and $\boldsymbol{n}$ the unit outward normal vector. The model consists of the second order derivative information of the displacement field and results in smoother deformations compared to those obtained using first order derivative models based on elastic and diffusion energies. It is refined in [41, 39, 40] with nonlinear boundary conditions. The affine linear transformation belongs to the kernel $\mathcal{S}^{\text{LC}}(\boldsymbol{u})$ which is not the case in elastic or diffusion registration.

Model MC. Next is the mean curvature model [19, 18]

$$\mathcal{S}^{\text{MC}}(\boldsymbol{u}) = \int_\Omega \Big[\Bbbk(\iota(u_1)) + \Bbbk(\iota(u_2)) \Big] \,\mathrm{d}\Omega,$$

where $\Bbbk(s) = \frac{1}{2}s^2$ and ι is as defined in (6.2). The Euler-Lagrange equation with $\mathcal{S}^{\text{MC}}$ as the regularisation term is given by:

$$\gamma \nabla \cdot \Big(\frac{1}{\sqrt{|\nabla u_l|^2 + 1}} \nabla \Bbbk'(\iota(u_l)) - \frac{\nabla u_l \cdot \nabla \Bbbk'(\iota(u_l))}{(\sqrt{|\nabla u_l|^2 + 1})^3} \nabla u_l \Big) + f_l(\boldsymbol{u}) = 0, \quad l = 1, 2 \tag{6.4}$$

with boundary condition $\nabla u_l \cdot \boldsymbol{n} = \nabla \iota(u_l) \cdot \boldsymbol{n} = 0,\ l = 1, 2$. One can use the multigrid method to solve equation (6.4) as in [19]; refer also to [18] for multi-modality image registration work.

Model D. Finally, the so-called demon registration method of Thirion [88] where pixels in the image act as demons that force pulling and pushing actions in a similar approach to Maxwell's for solving the Gibbs paradox in thermodynamics. The original demon registration model is a special case of diffusion registration but it has been much studied and improved since 1998; see [74, 67, 100, 63]. The energy functional for the basic demon method is given by

$$\mathcal{S}(\boldsymbol{u}) = \|R(\boldsymbol{x}) - T(\boldsymbol{x} + \widetilde{\boldsymbol{u}} + \boldsymbol{u})\|^2 + \frac{\sigma_i^2}{\sigma_x^2} \|\boldsymbol{u}\|^2 \tag{6.5}$$

where $\widetilde{\boldsymbol{u}}$ is the current displacement field, while σ_i^2 and σ_x^2 account for noise in the image intensity and the spatial uncertainty respectively. Equation (6.5) can be linearised using first order Taylor expansion,

$$\mathcal{J}(\boldsymbol{u}) = \|R(\boldsymbol{x}) - T(\boldsymbol{x} + \widetilde{\boldsymbol{u}}) + J\boldsymbol{u}\|^2 + \frac{\sigma_i^2}{\sigma_x^2} \|\boldsymbol{u}\|^2 \tag{6.6}$$

where J is given by

$$J = -\frac{\nabla R + \nabla T(\boldsymbol{x} + \widetilde{\boldsymbol{u}})}{2}$$

for an efficient second order minimisation. The first order condition of (6.6) leads to the new update for $\widetilde{\boldsymbol{u}}$

$$\boldsymbol{u} = -\frac{R(\boldsymbol{x}) - T(\boldsymbol{x} + \widetilde{\boldsymbol{u}})}{\|J\|^2 + \frac{\sigma_i^2}{\sigma_x^2}} J.$$

The additional use of $\boldsymbol{v}$ for $\boldsymbol{\varphi} = \exp(\boldsymbol{v})$ helps to achieve a nearly diffeormorphic transformation (mapping), where $\boldsymbol{v}$ is the stationary velocity field of the displacement field $\boldsymbol{u}$; see [95]. It should be remarked that the three main steps of the model cannot be combined into a single energy functional.

In a discrete setting, since the image domain Ω is a square, all variational models are discretised by finite differences on a uniform grid. Refer to [67]. The vertex grid is defined by

$$\Omega^h = \left\{ \boldsymbol{x}_{i,j} = (x_i, y_j) \;\middle|\; 0 \leq i \leq N_1 - 1,\ 0 \leq j \leq N_2 - 1 \right\}$$

where we shall re-use the notation T and R for discrete images of size $N_1 \times N_2$.

6.2 Mathematical Background of the Gaussian Curvature

In differential geometry, the Gaussian curvature problem seeks to identify a hypersurface of $\mathbb{R}^{d+1}$ as a graph $z = u(\boldsymbol{x})$ over $\boldsymbol{x} \in \Omega \subset \mathbb{R}^d$ so that, at each point of the surface, the Gaussian curvature is prescribed. Let $\kappa(\boldsymbol{x})$ denote the Gaussian curvature which is a real valued function in $\Omega \subset \mathbb{R}^d$. The problem is modelled by the following equation

$$\det(D^2 u) - \kappa(\boldsymbol{x})(1 + |Du|^2)^{(d+2)/2} = 0 \tag{6.7}$$

where D is the first order derivative operator. Equation (6.7) is one of the Monge-Ampere equations. For $d = 2$, we have

$$\kappa(\boldsymbol{x}) \equiv -\kappa^{GC} = \frac{u_{xx}u_{yy} - u_{xy}u_{yx}}{(1 + u_x^2 + u_y^2)^2}. \tag{6.8}$$

In [21], the authors define a regularisation term using the Gaussian curvature of a closed surface based on the Gauss-Bonnet theorem.

Theorem 6.2.1 ***Gauss-Bonnet Theorem.*** *For a compact C^2 surface $\partial\Sigma$, we have*

$$\int_{\partial\Sigma} \kappa^{GC} \, \mathrm{d}\sigma = 2\pi\chi$$

where $\mathrm{d}\sigma$ is the length element to the surface and χ is the Euler characteristic of the surface.

Using this Theorem, it was shown in [21] that the complete analogy of the total variation regularisation for surface fairing is the energy functional of the Gaussian curvature. The analogous term $\mathcal{S}$, to the total variation of a function, that appears in the ROF model [78], is given by

$$\mathcal{S} = \int_{\partial\Sigma} |\kappa(\boldsymbol{x})| \, \mathrm{d}\sigma$$

where $\mathrm{d}\sigma$ is the length element to the surface $\partial\Sigma$.

Gaussian curvature is one of the fundamental second order geometric properties of a surface. According to the Gauss's Theorema Egregium, Gaussian curvature is intrinsic. For a local isometric mapping $f : \partial\Sigma \to \partial\Sigma'$ between two surfaces, Gaussian curvature remains invariant i.e. if $p \in \partial\Sigma$ and $p' \in \partial\Sigma'$, then $\kappa^{GC}(p) = \kappa^{GC}(p')$ and the mapping f is smooth and diffeomorphic.

We can also use a level set function to define the Gaussian curvature. Denote by ϕ the zero level set of the surface generated through the mapping $(x, y) :\to (x, y, u(x, y))$. Then, let $\phi = u(x, y) - z$ and $\nabla\phi = (u_x, u_y, -1)^T$, where $u_x = \frac{\partial u}{\partial x}$ and $u_y = \frac{\partial u}{\partial y}$. The Gaussian curvature of the level set is given by

$$\kappa^{GC} = \frac{\nabla\phi H^*(\phi) \nabla\phi^T}{|\nabla\phi|^4}, \tag{6.9}$$

where $\nabla\phi = (\phi_x, \phi_y, \phi_z)^T$, $|\nabla\phi| = \sqrt{\phi_x^2 + \phi_y^2 + \phi_z^2}$, $H(\phi)$ is the Hessian matrix and $H^*(\phi)$ is the adjoint matrix for $H(\phi)$. We have

$$H(\phi) = \begin{bmatrix} u_{xx} & u_{xy} & 0 \\ u_{yx} & u_{yy} & 0 \\ 0 & 0 & 0 \end{bmatrix}, \qquad H^*(\phi) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & u_{xx}u_{yy} - u_{yx}u_{xy} \end{bmatrix}.$$

Thus,

$$\kappa^{GC} = \frac{u_{yx}u_{xy} - u_{xx}u_{yy}}{(u_x^2 + u_y^2 + 1)^2}.$$

This is why we set $\kappa^{GC} = -\kappa(\boldsymbol{x})$ in equation (6.8). We shall use $|\kappa^{GC}|$ to measure the Gaussian curvature as in [21] for a monotonically increasing function (since the functional should be nonnegative).

6.3 Image Registration based on Gaussian Curvature

Before introducing our new image registration model, we first state some facts which support the use of Gaussian curvature.

6.3.1 Advantages of Gaussian Curvature

The total variation and Gaussian curvature. We use the volume-based analysis introduced in [60] to compare two denoising models, based respectively on Gaussian curvature and total variation:

$$\frac{\partial u}{\partial t} = \nabla \cdot \Big(\kappa \Big(\Big| \frac{u_{xy}^2 - u_{xx}u_{yy}}{(u_x^2 + u_y^2 + 1)^2} \Big| \Big) \nabla u \Big), \tag{6.10}$$

$$\frac{\partial u}{\partial t} = \nabla \cdot \frac{\nabla u}{|\nabla u|}. \tag{6.11}$$

Consider, for each $\alpha > 0$, the time change of the volume $v_{t,\alpha} = \{(x,y,z) \mid 0 < z < |u(x,y,t) - \alpha|\}$ which is enclosed by the surface $z = u(x,y,t)$ and the plane $z = \alpha$. Assume $|u(x,y,t) - \alpha| = (u(x,y,t) - \alpha)s$ with s either positive ($s = 1$) or negative ($s = -1$) at all points. Denote by $c_{t,\alpha}$ the closed curve defined by the level set $u(x,y,t) = \alpha$ and by $d_{t,\alpha}$ the 2D region enclosed by $c_{t,\alpha}$. The volume change of $v_{t,\alpha}$ in time is given by

$$V = \frac{\partial}{\partial t}\int_{v_{t,\alpha}} \mathrm{d}z\,\mathrm{d}A = \frac{\partial}{\partial t}\int_{v_{t,\alpha}}\int_0^{|u(x,y,t)-\alpha|} \mathrm{d}z\,\mathrm{d}A = \frac{\partial}{\partial t}\int_{d_{t,\alpha}} |u(x,y,t) - \alpha|\,\mathrm{d}A$$

where $\mathrm{d}A$ is the area element. We now consider how V changes from evolving (6.10) or (6.11).

If u is the solution of equation (6.11), then from Gauss' theorem

$$V = \frac{\partial}{\partial t}\int_{d_{t,\alpha}} |u(x,y,t) - \alpha|\,\mathrm{d}A = s\int_{d_{t,\alpha}} \frac{\partial u}{\partial t}\,\mathrm{d}A = s\int_{d_{t,\alpha}} \nabla \cdot \frac{\nabla u}{|\nabla u|}\,\mathrm{d}A = s\int_{c_{t,\alpha}} \frac{\nabla u}{|\nabla u|} \cdot \boldsymbol{n}\,\mathrm{d}\sigma$$

where $\mathrm{d}\sigma$ is the length element and $\boldsymbol{n}$ is the unit normal vector to the curve $c_{t,\alpha}$ which is represented as $\boldsymbol{n} = s\frac{\nabla u}{|\nabla u|}$. Then

$$V = s^2\int_{c_{t,\alpha}} \frac{\nabla u}{|\nabla u|} \cdot \frac{\nabla u}{|\nabla u|}\,\mathrm{d}\sigma = \int_{c_{t,\alpha}} \mathrm{d}\sigma = |c_{t,\alpha}|$$

where $|c_{t,\alpha}|$ is the length of the curve $c_{t,\alpha}$. Furthermore, the volume variation in time is

$$\begin{aligned}\int_{d_{t+\delta t,\alpha}} |u(x,y,t+\delta t) - \alpha|\,\mathrm{d}A &\approx \int_{d_{t,\alpha}} |u - \alpha|\,\mathrm{d}A + s\delta t\int_{d_{t,\alpha}} \frac{\partial u}{\partial t}\,\mathrm{d}A \\ &= \int_{d_{t,\alpha}} \left[|u - \alpha| + s\delta t\frac{|c_{t,\alpha}|}{|d_{t,\alpha}|}\right]\mathrm{d}A\end{aligned}$$

where $|d_{t,\alpha}|$ denotes the area of the region $d_{t,\alpha}$. We can see that the change in u from t to $t + \delta t$ is proportional to the ratio $\frac{|c_{t,\alpha}|}{|d_{t,\alpha}|}$. So, when this ratio is large (indicating possibly a noise presence), the total variation model reduces it and hence removes noise. However, important features of u which have a large level set ratio are removed also and are not preserved by the total variation model (6.11).

Using similar calculations as before for the Gaussian curvature scheme (6.10), we have

$$\begin{aligned}V &= \frac{\partial}{\partial t}\int_{d_{t,\alpha}} |u(x,y,t) - \alpha|\,\mathrm{d}A = s\int_{d_{t,\alpha}} \nabla \cdot \left(\kappa\Big(\Big|\frac{u_{xy}^2 - u_{xx}u_{yy}}{(u_x^2 + u_y^2 + 1)^2}\Big|\Big)\nabla u\right)\mathrm{d}A \\ &= s\int_{c_{t,\alpha}} \left(\kappa\Big(\Big|\frac{u_{xy}^2 - u_{xx}u_{yy}}{(u_x^2 + u_y^2 + 1)^2}\Big|\Big)\nabla u\right) \cdot \boldsymbol{n}\,\mathrm{d}\sigma \; = \int_{c_{t,\alpha}} \left(\kappa\Big(\Big|\frac{u_{xy}^2 - u_{xx}u_{yy}}{(u_x^2 + u_y^2 + 1)^2}\Big|\Big)\right)|\nabla u|\,\mathrm{d}\sigma.\end{aligned} \tag{6.12}$$

From here, we observe that the quantity V for the subdomain $v_{t,\alpha}$ is dependent on the product of the variation *and* the Gaussian curvature on the level curve. The function κ in (6.12) controls and scales the speed of the volume change in contrast to the total variation scheme where V depends only on the variation of the level curve. Consider a point $p = (x_0, y_0, \alpha)$ where $\alpha = u(x_0, y_0)$. The Gaussian curvature $\kappa = \kappa_1\kappa_2$ is based on two principal curvatures κ_1 and κ_2 where κ_1 is the curvature of the level curve passing the point p, and κ_2 is the curvature of the path which passes through the point p and which is orthogonal to the level curve. If the Gaussian curvature on one level curve is zero then there is no change in V regardless of variation on the level curves. In contrast, with total variation, if there is a variation in the level curve, then there is a change in V. Based on this observation, we believe that the Gaussian curvature model is better than the total variation model for preserving features on surfaces.

The mean curvature and Gaussian curvature. The mean curvature (MC) $\iota = (\kappa_1 + \kappa_2)/2$ is also widely used. Next, we show that, though closely related, Gaussian curvature (GC) is better than mean curvature for surfaces in three ways.

First, Gaussian curvature is invariant under rigid and isometric transformations. In contrast, mean curvature is invariant under rigid transformations but not under isometric transformations. Rigid transformations preserve distance between two points while isometric transformations preserve length along surfaces and preserve angles between curves on surfaces. To illustrate invariance, consider a surface

$$z_1(x, y) = ax^2 + by^2,$$

whose Gaussian curvature and mean curvature are respectively

$$\kappa = \frac{0 - (2a)(2b)}{(1 + 4a^2x^2 + 4b^2y^2)^2}, \qquad \iota = \frac{(1 + 4b^2y^2)(2a) + (1 + 4a^2x^2)(2b)}{(1 + 4a^2x^2 + ab^2y^2)^{3/2}}.$$

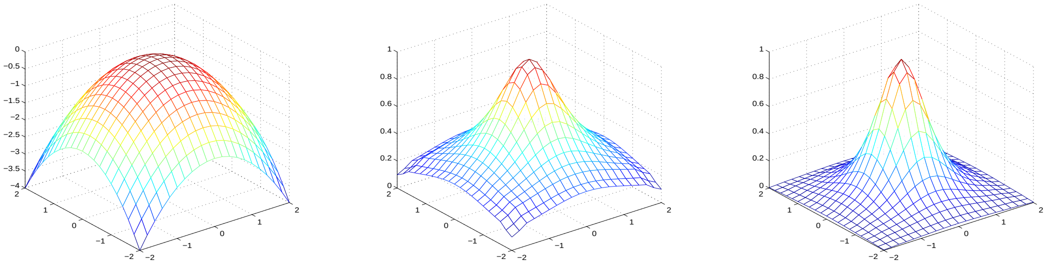
If we flip the surface upside down (isometric transformation) where $z_1'(x, y) = -ax^2 - by^2$, we will have the same value for the Gaussian curvature and a different value for the mean curvature. Thus, Gaussian curvature is invariant under isometric transformation.

Second, Gaussian curvature can be used to localise the tip of a surface better than mean curvature. Consider

$$z_2(x, y) = -\frac{1}{2}(x^2 + y^2)$$

as shown in Figure 6.1 (a). Then, we compute the mean and Gaussian curvature for the surface as depicted in Figures 6.1 (b) and (c) respectively. For Figure 6.1 (b), we display the negative of the mean curvature for better assessment and visualisation. For both figures, the maximal values are given at the centre of the tip. The value given by the Gaussian curvature is sharper than that of the mean curvature. The highest point of the Gaussian curvature is better distinguished from its neighbourhood compared to the highest point of the mean curvature.

Third, Gaussian curvature can locate saddle points better than mean curvature.



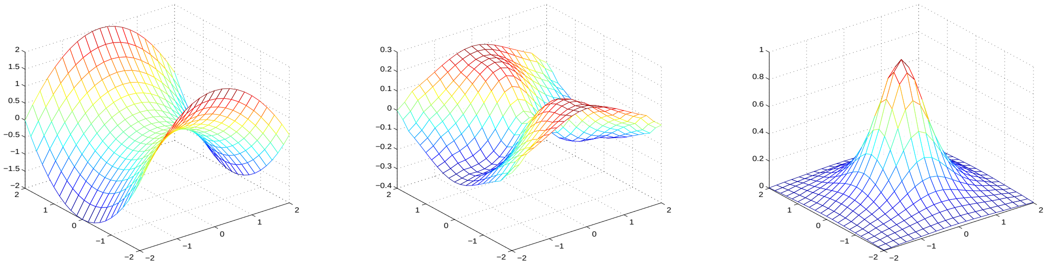
(a) Surface $z_2(x, y)$ (b) Negative mean curvature (c) Gaussian curvature

Figure 6.1: Representation of a surface with GC and MC. (a) shows a surface model with a tip point. (b) is the negative mean curvature and (c) is the Gaussian curvature. The highest point in (c) is better localised than in (b).

Take

$$z_3(x, y) = -\frac{1}{2}(x^2 - y^2)$$

as one example. The surface along with its mean and Gaussian curvatures are given in Figures 6.2 (a), (b) and (c) respectively. The mean curvature for this surface appears complex where the largest value is not at the saddle point and the saddle point cannot be easily located. However, Gaussian curvature gets its highest value at the saddle point and is therefore able to accurately identify the saddle point within its neighbourhood.



(a) Surface $z_3(x, y)$ (b) Negative mean curvature (c) Negative Gaussian curvature

Figure 6.2: Location of a surface's saddle point by GC and MC. (a) is the surface with a saddle point. (b) is the negative mean curvature and (c) is the negative Gaussian curvature. The highest point in (b) is not at the saddle point and for (c), the saddle point is better distinguished from its neighbourhood.

In addition to these three examples and observations, a very important fact pointed out in [21] is that the mean curvature of the surface is not a suitable choice for surface fairing because the model is not effective for preserving important features such as creases and corners on the surface (although the model is effective for removing noise). This is true when we are referring to surface fairing (surface denoising) but not necessarily true for 2D image denoising. From the recent work done in image denoising [21, 60], we observed several advantages of Gaussian curvature over total variation and mean curvature. Therefore, we might conjecture that Gaussian curvature may outperform existing models in image registration. To our knowledge there exists no previous work on this topic.

6.3.2 The Proposed Registration Model

Now we return to the problem of how to align or register two image functions $T(\boldsymbol{x}), R(\boldsymbol{x})$. Let the desired and unknown displacement fields between T and R be the surface map $(x, y) :\to (x, y, u_l(x, y))$ where $l = 1, 2$ and with $\boldsymbol{u} = (u_1, u_2)$. We propose our Gaussian curvature based image registration model as

$$\min_{\boldsymbol{u}\in C^2(\Omega)} \mathcal{J}_\gamma(\boldsymbol{u}(\boldsymbol{x})) = \frac{1}{2}\int_\Omega (T(\boldsymbol{x}+\boldsymbol{u}) - R(\boldsymbol{x}))^2\,\mathrm{d}\Omega + \gamma\mathcal{S}^{\mathrm{GC}}(\boldsymbol{u}(\boldsymbol{x})) \tag{6.13}$$

where

$$\mathcal{S}^{\mathrm{GC}}(\boldsymbol{u}(\boldsymbol{x})) = \sum_{l=1}^{2}\mathcal{S}^{GC}(u_l), \quad \mathcal{S}^{GC}(u_l) = \int_\Omega \left|\frac{u_{l,xy}u_{l,yx} - u_{l,xx}u_{l,yy}}{(u_{l,x}^2 + u_{l,y}^2 + 1)^2}\right|\mathrm{d}\Omega.$$

The above model (6.13) leads to two Euler Lagrange equations:

$$\begin{cases} \gamma\nabla\cdot\left(\dfrac{4|u_{1,xy}u_{1,yx} - u_{1,xx}u_{1,yy}|}{\boldsymbol{\mathcal{N}}_1^3}\nabla u_1\right) + \gamma\nabla\cdot\boldsymbol{B}_{1,1} + \gamma\nabla\cdot\boldsymbol{B}_{1,2} + f_1 = 0 \\ \gamma\nabla\cdot\left(\dfrac{4|u_{2,xy}u_{2,yx} - u_{2,xx}u_{2,yy}|}{\boldsymbol{\mathcal{N}}_2^3}\nabla u_2\right) + \gamma\nabla\cdot\boldsymbol{B}_{2,1} + \gamma\nabla\cdot\boldsymbol{B}_{2,2} + f_2 = 0 \end{cases} \tag{6.14}$$

where

$$\begin{aligned} &\boldsymbol{\mathcal{N}}_l = u_{l,x}^2 + u_{l,y}^2 + 1, \quad \boldsymbol{B}_{l,1} = \left(\left(-\frac{\mathcal{S}_l u_{l,yy}}{\boldsymbol{\mathcal{N}}_l}\right)_x, \left(\frac{\mathcal{S}_l u_{l,xy}}{\boldsymbol{\mathcal{N}}_l}\right)_x\right) \\ &\boldsymbol{B}_{l,2} = \left(\left(\frac{\mathcal{S}_l u_{l,yx}}{\boldsymbol{\mathcal{N}}_l}\right)_y, \left(-\frac{\mathcal{S}_l u_{l,xx}}{\boldsymbol{\mathcal{N}}_l}\right)_y\right), \quad \mathcal{S}_l = \mathrm{sign}(u_{l,xy}u_{l,yx} - u_{l,xx}u_{l,yy}) \\ &\boldsymbol{f} = (f_1, f_2)^T = (T(\boldsymbol{x}+\boldsymbol{u}) - R(\boldsymbol{x}))\nabla_{\boldsymbol{u}}T(\boldsymbol{x}+\boldsymbol{u}), \; l = 1, 2. \end{aligned}$$

The original boundary conditions for (6.14) which is obtain from the derivation of the Euler-Lagrange equations are

$$\Big(\frac{\mathcal{S}_l(u_{l,y})_y}{\boldsymbol{\mathcal{N}}_l^2}, \frac{\mathcal{S}_l(u_{l,y})_x}{\boldsymbol{\mathcal{N}}_l^2}\Big)\cdot\boldsymbol{n} = 0, \;\; \Big(\frac{\mathcal{S}_l(u_{l,x})_y}{\boldsymbol{\mathcal{N}}_l^2}, \frac{\mathcal{S}_l(u_{l,x})_x}{\boldsymbol{\mathcal{N}}_l^2}\Big)\cdot\boldsymbol{n} = 0 \tag{6.15}$$

for $l = 1, 2$ and $\boldsymbol{n}$ is the normal vector at the boundary $\partial\Omega$.

6.3.3 Derivation of the Euler-Lagrange Equations (6.14)

Let $q_1 = u_x$ and $q_2 = u_y$, then we can write the Gaussian curvature regularisation term as

$$\mathcal{S}^{GC}(q_1, q_2) = \int_\Omega \left|\frac{q_{1,x}q_{2,y} - q_{1,y}q_{2,x}}{(1 + q_1^2 + q_2^2)^2}\right|\mathrm{d}x\,\mathrm{d}y.$$

From the optimality condition

$$\frac{d\mathcal{S}^{GC}(q_1,q_2)}{dq_1} = 0 \text{ and } \frac{d\mathcal{S}^{GC}(q_1,q_2)}{dq_2} = 0, \tag{6.16}$$

then,

$$\frac{d}{d\epsilon_1}\mathcal{S}^{GC}(q_1+\epsilon_1\varphi_1, q_2)\Big|_{\epsilon_1=0} = 0 \tag{6.17}$$

and

$$\frac{d}{d\epsilon_2}\mathcal{S}^{GC}(q_1, q_2+\epsilon_2\varphi_2)\Big|_{\epsilon_2=0} = 0. \tag{6.18}$$

We have,

$$\begin{aligned}&\frac{d}{d\epsilon_1}\int_\Omega \Big|\frac{(q_1+\epsilon_1\varphi_1)_x q_{2,y} - (q_1+\epsilon_1\varphi_1)_y q_{2,x}}{(1+(q_1+\epsilon_1\varphi_1)^2+q_2^2)^2}\Big| \,\mathrm{d}x\,\mathrm{d}y\Big|_{\epsilon=0}\\ &= \int_\Omega S\frac{d}{d\epsilon_1}\left[\frac{(q_1+\epsilon_1\varphi_1)_x q_{2,y} - (q_1+\epsilon_1\varphi_1)_y q_{2,x}}{(1+(q_1+\epsilon_1\varphi_1)^2+q_2^2)^2}\right]\mathrm{d}x\,\mathrm{d}y\Big|_{\epsilon=0}\\ &= 0\end{aligned} \tag{6.19}$$

where $S = \mathrm{sign}\left(\frac{q_{1,x}q_{2,y}-q_{1,y}q_{2,x}}{(1+q_1^2+q_2^2)^2}\right)$. From (6.19),

$$\begin{aligned}&\int_\Omega S\left[\frac{\varphi_{1,x}q_{2,y}-\varphi_{1,y}q_{2,x}}{(1+q_1^2+q_2^2)^2} + (q_{1,x}q_{2,y}-q_{1,y}q_{2,x})(-4\varphi_1 q_1(1+q_1^2+q_2^2)^{-3})\right]\mathrm{d}x\,\mathrm{d}y\\ &= \int_\Omega \frac{S\varphi_{1,x}q_{2,y}}{\Gamma^2} - \frac{S\varphi_{1,y}q_{2,x}}{\Gamma^2} - \frac{4SDq_1\varphi_1}{\Gamma^3}\,\mathrm{d}x\,\mathrm{d}y\\ &= 0.\end{aligned}$$

where $\Gamma = 1+q_1^2+q_2^2, D = q_{1,x}q_{2,y}-q_{1,y}q_{2,x}$.

Using the Green's theorem

$$\int_{\partial\Omega}\phi\boldsymbol{\omega}\cdot\boldsymbol{n}\,\mathrm{d}s - \int_\Omega \phi\mathrm{div}(\boldsymbol{\omega})\,\mathrm{d}x\,\mathrm{d}y = \int_\Omega \nabla\phi\cdot\boldsymbol{\omega}\,\mathrm{d}x\,\mathrm{d}y \tag{6.20}$$

then,

$$\begin{aligned}&\int_\Omega \frac{S\varphi_{1,x}q_{2,y}}{\Gamma^2} - \frac{S\varphi_{1,y}q_{2,x}}{\Gamma^2}\,\mathrm{d}x\,\mathrm{d}y = \int_{\partial\Omega}\varphi_1\left(\frac{Sq_{2,y}}{\Gamma^2},\frac{Sq_{2,x}}{\Gamma^2}\right)\cdot\boldsymbol{n}\,\mathrm{d}s - \int_\Omega \varphi_1\mathrm{div}\left(\frac{Sq_{2,y}}{\Gamma^2},\frac{Sq_{2,x}}{\Gamma^2}\right)\\ &= 0\end{aligned}$$

where $\phi = \varphi_1, \boldsymbol{\omega} = \left(\frac{Sq_{2,y}}{\Gamma^2},\frac{Sq_{2,x}}{\Gamma^2}\right)$. Setting the boundary integral to zero, then we have

$$\int_\Omega \varphi_1\mathrm{div}\left(\frac{Sq_{2,y}}{\Gamma^2},\frac{Sq_{2,x}}{\Gamma^2}\right)\mathrm{d}x\,\mathrm{d}y = 0.$$

Finally, we use the fundamental lemma of the calculus of variation:

$$\nabla\cdot\left(\frac{Sq_{2,y}}{\Gamma^2},\frac{Sq_{2,x}}{\Gamma^2}\right) - \frac{4SDq_1}{\Gamma^3} = 0.$$

Similarly, for $\frac{d}{d\epsilon_2}\mathcal{S}^{GC}(q_1,q_2+\epsilon_2\varphi_2)\Big|_{\epsilon_2=0} = 0$, we finally obtain equation (6.14).

From the derivation of the Euler-Lagrange equation we obtain the following boundary conditions

$$\Big(\frac{Sq_{2,y}}{\Gamma^2}, \frac{Sq_{2,x}}{\Gamma^2}\Big) \cdot \boldsymbol{n} = 0$$

and

$$\Big(\frac{Sq_{1,y}}{\Gamma^2}, \frac{Sq_{1,x}}{\Gamma^2}\Big) \cdot \boldsymbol{n} = 0.$$

Since $q_1 = u_x$ and $q_2 = u_y$, we have

$$(u_y)_y = 0, (u_y)_x = 0, (u_x)_y = 0, \text{ and } (u_x)_x = 0. \tag{6.21}$$

However, later in Section 6.3.4 we choose two Neumann boundary conditions for solving the Euler-Lagrange equations in (6.14) which satisfied the original boundary conditions in (6.21).

6.3.4 Augmented Lagrangian Method

The augmented Lagrangian method (ALM) is often used for solving constraint minimisation problems by replacing the original problem with an unconstrained problem. The method is similar to the penalty method where the constraints are incorporated in the objective functional and the problem is solved using alternating minimisation of each of the sub-problems. However, in ALM, there are additional terms in the objective functional, known as Lagrange multiplier terms, which arise when incorporating the constraints. Similar work on the augmented Lagrangian method in image restoration can be found in [109, 110].

To proceed, we introduce two new dual variables $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$ where $\boldsymbol{q}_1 = \nabla u_1(\boldsymbol{x})$ and $\boldsymbol{q}_2 = \nabla u_2(\boldsymbol{x})$. Consequently we obtain a system of second order PDEs which are more amenable to effective solution.

We obtain the following refined model for Gaussian curvature image registration

$$\begin{aligned}\min_{u_1,u_2,\boldsymbol{q}_1,\boldsymbol{q}_2} \mathcal{J}(u_1,u_2,\boldsymbol{q}_1,\boldsymbol{q}_2) = \mathcal{D}(T,R,\boldsymbol{u}(\boldsymbol{x})) + \gamma\mathcal{S}^{GC}(\boldsymbol{q}_1) + \gamma\mathcal{S}^{GC}(\boldsymbol{q}_2)\\ \text{s.t } \boldsymbol{q}_1 = \nabla u_1(\boldsymbol{x}), \boldsymbol{q}_2 = \nabla u_2(\boldsymbol{x})\end{aligned}$$

and further reformulate $\mathcal{J}(u_1,u_2,\boldsymbol{q}_1,\boldsymbol{q}_2)$ to get the augmented Lagrangian functional

$$\begin{aligned}\mathcal{L}^{GC}(u_1,u_2,\boldsymbol{q}_1,\boldsymbol{q}_2;\boldsymbol{\mu}_1,\boldsymbol{\mu}_2) =& \frac{1}{2}\|T(\boldsymbol{x}+\boldsymbol{u}(\boldsymbol{x})) - R(\boldsymbol{x})\|_2^2 + \gamma\mathcal{S}^{GC}(\boldsymbol{q}_1) + \gamma\mathcal{S}^{GC}(\boldsymbol{q}_2)\\ &+ \langle\boldsymbol{\mu}_1, \boldsymbol{q}_1 - \nabla u_1\rangle + \langle\boldsymbol{\mu}_2, \boldsymbol{q}_2 - \nabla u_2\rangle\\ &+ \frac{r}{2}\|\boldsymbol{q}_1 - \nabla u_1\|_2^2 + \frac{r}{2}\|\boldsymbol{q}_2 - \nabla u_2\|_2^2\end{aligned} \tag{6.22}$$

where $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2$ are the Lagrange multipliers, the inner products are defined via the usual integration in Ω and r is a positive constant. We use an alternating minimisation procedure to find the optimal values of $u_1, u_2, \boldsymbol{q}_1, \boldsymbol{q}_2$ and $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2$ where the process involves only two main steps.

Step 1. For the first step we need to update $\boldsymbol{q}_1, \boldsymbol{q}_2$ for any given $u_1, u_2, \boldsymbol{\mu}_1, \boldsymbol{\mu}_2$. The objective functional is given by

$$\min_{\boldsymbol{q}_1,\boldsymbol{q}_2} \gamma \mathcal{S}^{GC}(\boldsymbol{q}_1) + \gamma \mathcal{S}^{GC}(\boldsymbol{q}_2) + \langle \boldsymbol{\mu}_1, \boldsymbol{q}_1 \rangle + \langle \boldsymbol{\mu}_2, \boldsymbol{q}_2 \rangle + \frac{r}{2}\|\boldsymbol{q}_1 - \nabla u_1\|^2 + \frac{r}{2}\|\boldsymbol{q}_2 - \nabla u_2\|^2.$$

This sub-problem can be solved using the following Euler Lagrange equations:

$$\begin{cases} -\gamma\Big(\Big(\frac{(-q_{1,1})_y}{\Gamma_1^2}\Big)_x + \Big(\frac{-(q_{1,1})_x}{\Gamma_1^2}\Big)_y\Big) - \gamma\frac{4S_1D_1q_{1,2}}{\Gamma_1^3} + \mu_{1,2} + r(q_{1,2} - (u_1)_y) = 0, \\ -\gamma\Big(\Big(\frac{(q_{1,2})_y}{\Gamma_1^2}\Big)_x + \Big(\frac{-(q_{1,2})_x}{\Gamma_1^2}\Big)_y\Big) - \gamma\frac{4S_1D_1q_{1,1}}{\Gamma_1^3} + \mu_{1,1} + r(q_{1,1} - (u_1)_x) = 0 \end{cases} \tag{6.23}$$

where $D_1 = \det(\nabla \boldsymbol{q}_1) = (q_{1,1})_x(q_{1,2})_y - (q_{1,1})_y(q_{1,2})_x$, $\Gamma_1 = 1 + u_{1,x}^2 + u_{1,y}^2$ and $S_1 = \text{sign}\Big(\frac{D_1}{(\|\nabla u_1\|^2+1)^2}\Big)$. We have a closed form solution for this step, if solving alternatingly, where

$$q_{1,1} = \frac{\Gamma_1^3\Big(-\gamma\Big(\Big(\frac{(q_{1,2})_y}{\Gamma_1^2}\Big)_x + \Big(\frac{-(q_{1,2})_x}{\Gamma_1^2}\Big)_y\Big) + \mu_{1,1} - r(u_1)_x\Big)}{-r\Gamma_1^3 + \gamma 4S_1D_1},$$
$$q_{1,2} = \frac{\Gamma_1^3\Big(-\gamma\Big(\Big(\frac{(q_{1,1})_y}{\Gamma_1^2}\Big)_x + \Big(\frac{-(q_{1,1})_x}{\Gamma_1^2}\Big)_y\Big) + \mu_{1,2} - r(u_1)_y\Big)}{-r\Gamma_1^3 + \gamma 4S_1D_1}.$$

Similarly, we solve $q_{2,1}, q_{2,2}$ from

$$\begin{cases} -\gamma\Big(\Big(\frac{(-q_{2,1})_y}{\Gamma_2^2}\Big)_x + \Big(\frac{-(q_{2,1})_x}{\Gamma_2^2}\Big)_y\Big) - \gamma\frac{4S_2D_2q_{2,2}}{\Gamma_2^3} + \mu_{2,1} + r(q_{2,2} - (u_2)_y) = 0, \\ -\gamma\Big(\Big(\frac{(q_{2,2})_y}{\Gamma_2^2}\Big)_x + \Big(\frac{-(q_{2,2})_x}{\Gamma_2^2}\Big)_y\Big) - \gamma\frac{4S_2D_2q_{2,1}}{\Gamma_2^3} + \mu_{2,1} + r(q_{2,1} - (u_2)_x) = 0 \end{cases} \tag{6.24}$$

where $D_2 = \det(\nabla \boldsymbol{q}_2) = (q_{2,1})_x(q_{2,2})_y - (q_{2,1})_y(q_{2,2})_x$, $\Gamma_2 = 1 + u_{2,x}^2 + u_{2,y}^2$ and $S_2 = \text{sign}\Big(\frac{D_2}{(\|\nabla u_2\|^2+1)^2}\Big)$.

Step 2. For the second step we need to update u_1, u_2 for any given $\boldsymbol{q}_1, \boldsymbol{q}_2$ and $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2$ with the following functional

$$\min_{u_1,u_2} \frac{1}{2}\|T(\boldsymbol{x} + \boldsymbol{u}) - R(\boldsymbol{x})\|_2^2 - \langle \boldsymbol{\mu}_1, \nabla u_1 \rangle - \langle \boldsymbol{\mu}_2, \nabla u_2 \rangle + \frac{r}{2}\|\boldsymbol{q}_1 - \nabla u_1\|^2 + \frac{r}{2}\|\boldsymbol{q}_2 - \nabla u_2\|^2.$$

Thus, we have the following Euler Lagrange equations:

$$\begin{cases} -r\Delta u_1 + f_1 + \nabla \cdot \boldsymbol{\mu}_1 + r\nabla \cdot \boldsymbol{q}_1 = 0 \\ -r\Delta u_2 + f_2 + \nabla \cdot \boldsymbol{\mu}_2 + r\nabla \cdot \boldsymbol{q}_2 = 0 \end{cases} \tag{6.25}$$

with Neumann boundary conditions $\nabla u_l \cdot \boldsymbol{n} = 0, l = 1, 2$. To solve equation (6.25), first, we linearise the force term $\boldsymbol{f}$ using the Taylor expansion

$$\begin{aligned} f_l(u_1^{(k+1)}, u_2^{(k+1)}) &= f_l(u_1^{(k)}, u_2^{(k)}) + \partial_{u_1} f_l(u_1^{(k)}, u_2^{(k)})\delta u_l^{(k)} + \partial_{u_2} f_1(u_1^{(k)}, u_2^{(k)})\delta u_2^{(k)} + \ldots \\ &\approx f_l(u_1^{(k)}, u_2^{(k)}) + \sigma_{l,1}^{(k)}\delta u_1^{(k)} + \sigma_{l,2}\delta u_2^{(k)} \end{aligned} \tag{6.26}$$

where

$$\sigma_{l,1}^{(k)} = \partial_{u_1} f_l(u_1^{(k)}, u_2^{(k)}), \sigma_{l,2} = \partial_{u_2} f_l(u_1^{(k)}, u_2^{(k)}), \delta u_1^{(k)} = u_1^{(k+1)} - u_1^{(k)}, \delta u_2^{(k)} = u_2^{(k+1)} - u_2^{(k)}.$$

Second, we approximate $\sigma_{l,1}^{(k)}$ and $\sigma_{l,2}^{(k)}$ with

$$\begin{aligned}\sigma_{l,1}^{(k)} &= \Big(\partial u_l T(\boldsymbol{x} + \boldsymbol{u}^{(k)})\Big)\Big(\partial u_1 T(\boldsymbol{x} + \boldsymbol{u}^{(k)})\Big)\\ \sigma_{l,2}^{(k)} &= \Big(\partial u_l T(\boldsymbol{x} + \boldsymbol{u}^{(k)})\Big)\Big(\partial u_2 T(\boldsymbol{x} + \boldsymbol{u}^{(k)})\Big).\end{aligned}$$

The discrete version of equation (6.25) is as follows

$$\boldsymbol{N}^h(\boldsymbol{u}^{h,(k)})\boldsymbol{u}^{h,(k+1)} = \boldsymbol{B}^h(\boldsymbol{u}^{h,(k)}) \tag{6.27}$$

where

$$\begin{aligned}\boldsymbol{N}^h(\boldsymbol{u}^{(k)}) &= \begin{bmatrix} -r\mathcal{L} + \sigma_{11}^h(\boldsymbol{u}^{h,(k)}) & \sigma_{12}^h(\boldsymbol{u}^{h,(k)}) \\ \sigma_{21}^h(\boldsymbol{u}^{h,(k)}) & -r\mathcal{L} + \sigma_{22}^h(\boldsymbol{u}^{h,(k)}) \end{bmatrix},\\ \boldsymbol{B}^h(\boldsymbol{u}^{(k)}) &= \begin{bmatrix} -G_1^h + f_1^h(u_1^{(k)}, u_2^{(k)}) + \sigma_{11}^h(\boldsymbol{u}^{(k)})u_1^{h,(k)} + \sigma_{12}^h(\boldsymbol{u}^{h,(k)})u_2^{h,(k)} \\ -G_2^h + f_2^h(u_1^{(k)}, u_2^{(k)}) + \sigma_{21}^h(\boldsymbol{u}^{(k)})u_1^{h,(k)} + \sigma_{22}^h(\boldsymbol{u}^{h,(k)})u_2^{h,(k)} \end{bmatrix},\end{aligned}$$

$\mathcal{L}$ is the discrete version of the Laplace operator Δ and G_l^h is the discrete version of

$$\nabla \cdot \boldsymbol{\mu}_l + r\nabla \cdot \boldsymbol{q}_l,\ l = 1, 2.$$

Third, we solve the system of equations (6.27) using a weighted pointwise Gauss Seidel method

$$\boldsymbol{u}^{h,(k+1)} = (1-\omega)\boldsymbol{u}^{h,(k)} + \omega\Big(\boldsymbol{N}^h(\boldsymbol{u}^{(k)})\Big)^{(-1)}\boldsymbol{B}^h(\boldsymbol{u}^{(k)})$$

where $\omega \in (0, 2)$ and we choose $\omega = 0.9725$.

The iterative algorithm to solve (6.22) is now summarised as follows. We linearise the force term $\boldsymbol{f}$ using a first order approximation from Taylor expansion in (6.26) to

Algorithm 9 Augmented Lagrangian Method for Gaussian Curvature Image Registration.

```
1. Initialise $\boldsymbol{\mu}_1 = \boldsymbol{\mu}_2 = \boldsymbol{0}, \boldsymbol{u}(\boldsymbol{x}) = 0, \gamma, r$.
2. For $k = 0, 1, ..., IMAX$
   (a) Step 1: Solve (6.23-6.24) for $(\boldsymbol{q}_1^{(k+1)}, \boldsymbol{q}_2^{(k+1)})$ with $(u_1, u_2) = (u_1^{(k)}, u_2^{(k)})$.
   (b) Step 2: Solve (6.25) for $(u_1^{(k+1)}, u_2^{(k+1)})$ with $(\boldsymbol{q}_1, \boldsymbol{q}_2) = (\boldsymbol{q}_1^{(k+1)}, \boldsymbol{q}_2^{(k+1)})$.
   (c) Step 3: Update Lagrange multipliers.
       $\boldsymbol{\mu}_1^{(k+1)} = \boldsymbol{\mu}_1^{(k)} + r(\boldsymbol{q}_1^{(k+1)} - \nabla u_1^{(k+1)})$, $\boldsymbol{\mu}_2^{(k+1)} = \boldsymbol{\mu}_2^{(k)} + r(\boldsymbol{q}_2^{(k+1)} - \nabla u_2^{(k+1)})$
3. End for.
```

have a simple and stable numerical scheme. Noted that

$$\begin{aligned}\sigma_{l,1}^{(k)} &= \partial_{u_1} f_l(u_1^{(k)}, u_2^{(k)}) \\ &= \partial_{u_1}\Big[\Big(T(\boldsymbol{x}+\boldsymbol{u}^{(k)}) - R(\boldsymbol{x})\Big)\partial_{u_l} T(\boldsymbol{x}+\boldsymbol{u}^{(k)})\Big] \\ &= \partial_{u_1} T(\boldsymbol{x}+\boldsymbol{u}^{(k)})\partial_{u_l} T(\boldsymbol{x}+\boldsymbol{u}^{(k)}) + \Big(T(\boldsymbol{x}+\boldsymbol{u}^{(k)}) - R(\boldsymbol{x})\Big)\partial_{u_1}\Big(\partial_{u_l} T(\boldsymbol{x}+\boldsymbol{u}^{(k)})\Big) \\ &\approx \partial_{u_1} T(\boldsymbol{x}+\boldsymbol{u}^{(k)})\partial_{u_l} T(\boldsymbol{x}+\boldsymbol{u}^{(k)})\end{aligned}$$

because the image difference $T(\boldsymbol{x}+\boldsymbol{u}^{(k)}) - R(\boldsymbol{x})$ becomes small for well registered images. The second order derivative of $T(\boldsymbol{x}+\boldsymbol{u}^{(k)})$ represented by $\partial_{u_1}\Big(\partial_{u_l} T(\boldsymbol{x}+\boldsymbol{u}^{(k)})\Big)$ is a problematic and difficult part of $\sigma_{l,1}^{(k)}$. The term is very sensitive to noise and it requires high computational cost to estimate properly. If the discrete image gradient $\partial_{u_l} T(\boldsymbol{x}+\boldsymbol{u}^{(k)})$ does not vanish at one point, the matrix system in (6.27) is strictly or irreducibly diagonally dominant. This guarantee the existence of a unique solution of each linearised system and global convergence of Gauss Seidel iteration [17].

6.4 Numerical Results

We use two numerical experiments to examine the efficiency and robustness of Algorithm 9 on a variety of deformations. To judge the quality of the alignment we calculate the relative reduction of the similarity measure

$$\varepsilon = \frac{\mathcal{D}(T, R, \boldsymbol{u}^{(*)})}{\mathcal{D}(T, R, \boldsymbol{u}^{(0)})}$$

and the minimum value of the determinant of the Jacobian matrix J of the transformation, denoted by $\mathcal{F}$

$$J = \left[\begin{array}{cc} 1+u_{1,x} & u_{1,y} \\ u_{2,x} & 1+u_{2,y} \end{array}\right], \ \mathcal{F} = \min\left(\det(J)\right). \tag{6.28}$$

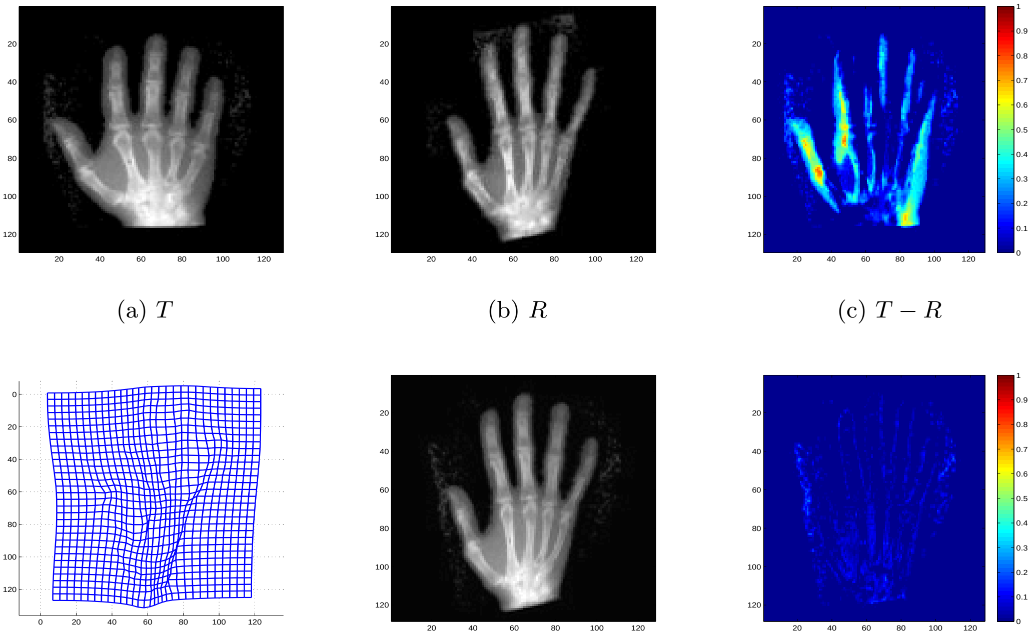
We can observe that when $\mathcal{F} > 0$, the deformed grid is free from folding and cracking.

All experiments were run on a single level. Experimentally, we found that $r \in [0.02, 2]$ works well for several types of image. As for the stopping criterion, we use $tol = 0.001$ for the residual of the Euler-Lagrange equations (6.23)-(6.25) and the maximum number of iterations is 30. Experiments were carried out using Matlab R2014b with Intel(R) core (TM) i7-2600 processor and 16G RAM.

6.4.1 Test 1: A Pair of Smooth X-ray Images

Images for Test 1 are taken from [68] where X-ray images of two hands of different individuals need to be aligned. The size of the images is 128 × 128 and the recovered transformation is expected to be smooth. The scaled version of the transformation and the transformed template image are given in Figures 6.3 (d) and (e) respectively.

The transformation is smooth and the model is able to solve such a problem. For



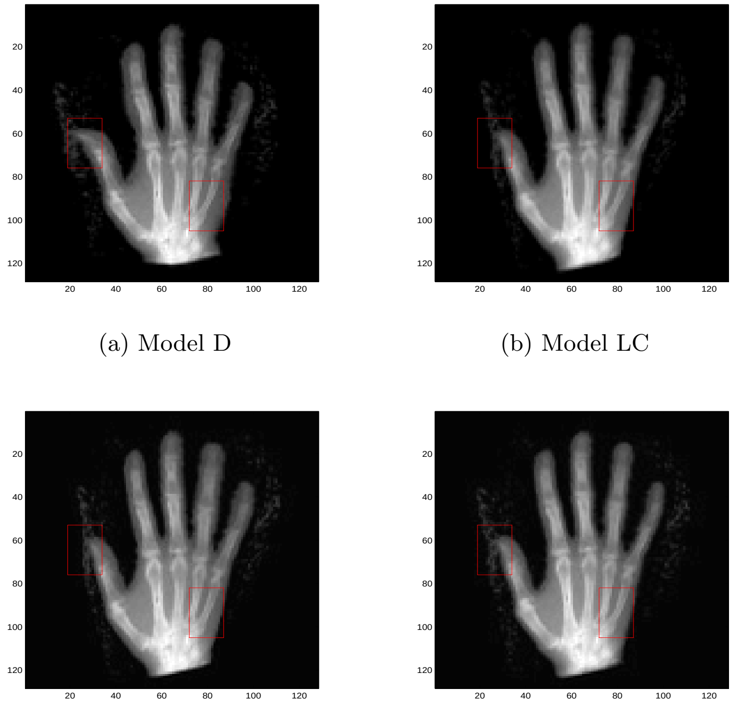
(a) T (b) R (c) $T - R$

(d) $\boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x})$ (e) $T(\boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x}))$ (f) $T(\boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x})) - R$

Figure 6.3: Test 1 (X-ray of hand). Illustration of the effectiveness of Gaussian curvature with smooth problems. On the top row, from left to right: (a) template, (b) reference and (c) the difference before registration. On the bottom row, from left to right: (d) the transformation applied to a regular grid, (e) the transformed template image and (f) the difference after registration. As can be seen from the result (e) and the small difference after registration (f), Gaussian curvature is able to solve smooth problems.

comparison, the transformed template images for the diffeormorphic demon method, linear, mean and Gaussian curvatures are shown in Figures 6.4 (a), (b), (c) and (d) respectively. We can observe that there are some differences of these images inside the red boxes where only Gaussian curvature delivers the best result of the features inside the boxes. Enlargements of the red boxes in Figure 6.4 are shown in Figure 6.5 for all models, with the best result given by the Gaussian curvature for both parts of the hand.

We summarise the results for Test 1 in Table 6.1 where ML and SL stand for multi and single level respectively. For all models, γ is chosen as small as possible such that $\mathcal{F} > 0$;, here for Model D, $\gamma = \frac{\sigma_i^2}{\sigma_x^2}$. We can see that the fastest model is the diffeormorphic demon, followed by linear and mean curvature. The current implementation for Gaussian curvature is on single level and the model uses the augmented Lagrangian method which has four dual variables and four Lagrange multipliers terms. Thus, it requires more computational time than the other models.



(a) Model D

(b) Model LC

(c) Model MC

(d) Gaussian curvature

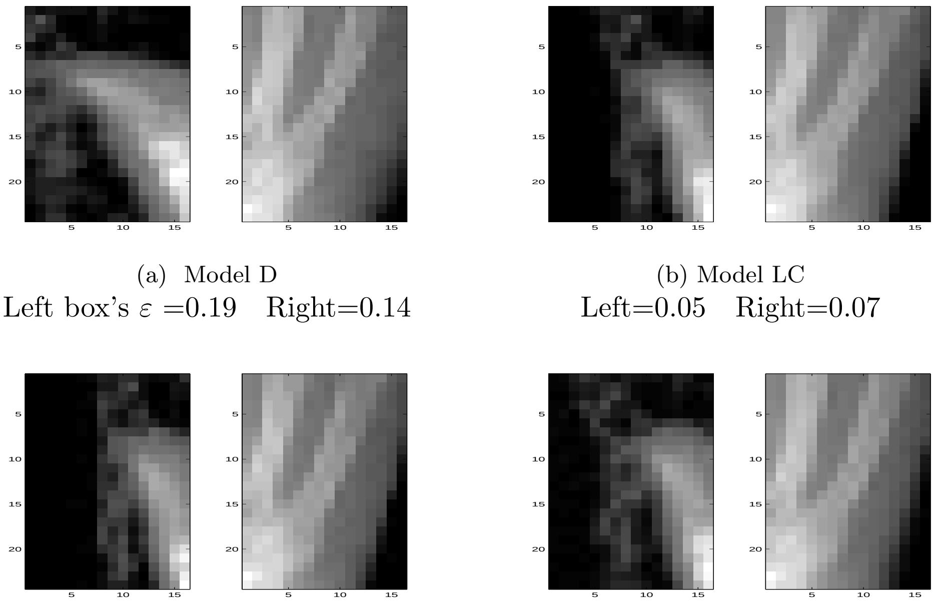
Figure 6.4: Test 1 (X-ray of hand). Comparison of Gaussian curvature with competing methods. The transformed template image using (a) Model D, (b) Model LC, (c) Model MC and (d) Gaussian curvature. Note the difference of these three images inside the red boxes.

Measure	Model D		Model LC		Model MC	GC
	ML	SL	ML	SL	SL	SL
γ	1.6	1.6	0.1	0.5	0.0001	0.0001
Time (s)	15.19	186.48	84.33	12.98	275.3	953.15
ε	0.1389	0.1229	0.0720	0.3780	0.0964	0.0582
$\mathcal{F}$	0.0600	0.1082	0.3894	0.1973	0.6390	0.3264

Table 6.1: Quantitative measurements for all models for Test 1. ML and SL stand for multi and single level respectively. γ is chosen as small as possible such that $\mathcal{F} > 0$ for all methods. $\mathcal{F} > 0$ indicates the deformation consists of no folding and cracking of the deformed grid. We can see that the smallest value of ε is given by Gaussian curvature (GC).

6.4.2 Test 2: A Pair of Brain MR Images

We take as Test 2 a pair of medical images of size 256 × 256 from the Internet Brain Segmentation Repository (IBSR) `https://www.nitrc.org/project/ibsr` where 20 normal MR brain images and their manual segmentations are provided. We choose a particular pair of individuals with different sizes of ventricle to illustrate a large deformation problem. Figure 6.6 shows the test images and the registration results



(a) Model D
Left box's ε =0.19   Right=0.14

(b) Model LC
Left=0.05   Right=0.07

(c) Model MC
Left box's ε =0.05   Right=0.07

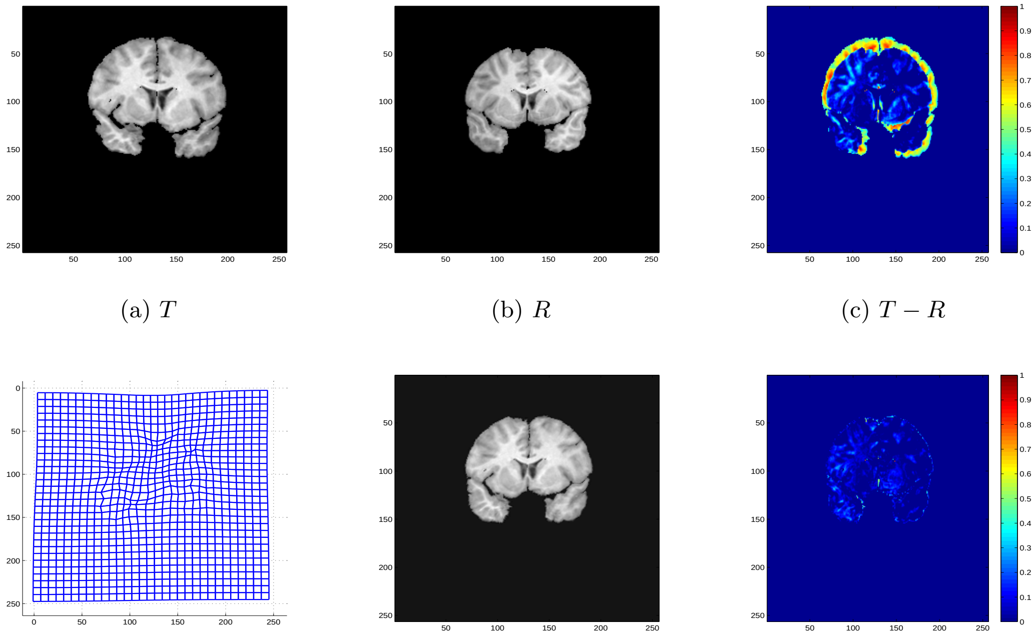
(d) Gaussian curvature
Left=0.05   Right=0.05

Figure 6.5: Test 1 (X-ray of hand). Comparison of transformed templates in zoomed-in boxes and their local ε values: (a) Model D, (b) Model LC, (c) Model MC and (d) Gaussian curvature. Gaussian curvature has the smallest ε value.

using the Gaussian curvature model. We can see that the model is able to solve real medical problems involving large deformations, which is particularly important for atlas construction in medical applications. Figure 6.7 shows the transformed template images for all four methods. We can see that Gaussian curvature gives the best result inside the red boxes in comparison with the diffeomorphic demon, the linear and mean curvature models as depicted in Figure 6.7 (d). Enlargements of the red boxes in Figure 6.7 are shown in Figure 6.8 where we can observe that Gaussian curvature gives better alignment for both parts of the brain.

Measure	Model D		Model LC		Model MC	GC
	ML	SL	ML	SL	SL	SL
γ	1.2	1.4	0.16	2.0	0.0001	0.0001
Time (s)	23.89	209.00	275.04	35.70	830.22	1053.7
ε	0.2004	0.7580	0.1128	0.4283	0.1998	0.1062
$\mathcal{F}$	0.0277	0.0387	0.3157	0.0148	0.8240	0.0138

Table 6.2: Quantitative measurements for all models for Test 2. ML and SL stand for multi and single level respectively. γ is chosen to be as small as possible such that $\mathcal{F} > 0$ for all models. $\mathcal{F} > 0$ indicates the deformation consists of no folding and cracking of the deformed grid. We can see that the smallest value of ε is given by Gaussian curvature (GC).



(a) T

(b) R

(c) $T - R$

(d) $\boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x})$

(e) $T(\boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x}))$

(f) $T(\boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x})) - R$

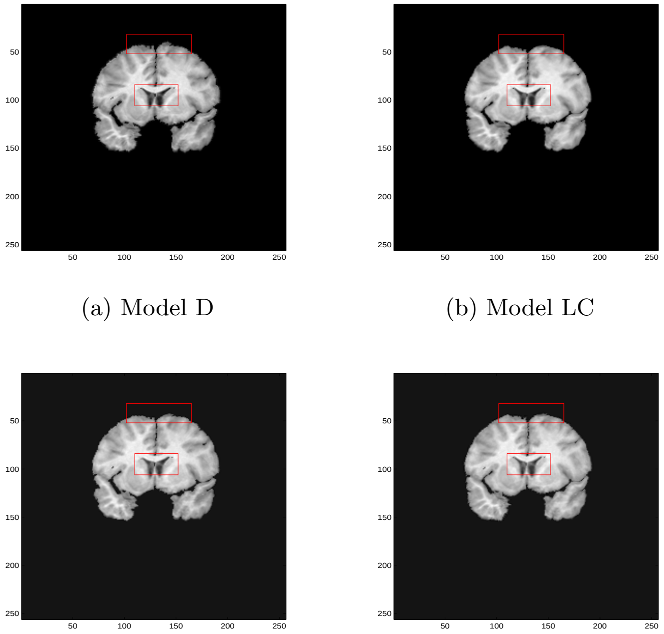
Figure 6.6: Test 2: A pair of Brain MR images. Illustration of the effectiveness of Gaussian curvature with real medical images. On the top row, from left to right: (a) template, (b) reference and (c) the difference before registration. On the bottom row, from left to right: (d) the transformation applied to a regular grid, (e) the transformed template image and (f) the difference after registration. As can be seen from the result (e) and the small difference after registration (f), Gaussian curvature can be applied to real medical images and is able to obtain good results.

The values of the quantitative measurements for Test 2 are recorded in Table 6.2 where the lowest values of ε are given by the Gaussian curvature model, indicating higher similarity between the transformed template result and the reference image. However, our propose model required more time than the other models since our model consists of more variables than the others.

6.5 Discussion

Gaussian curvature is proposed as a novel regularisation term for variational formulation based image registration. We have presented an efficient numerical scheme using the augmented Lagrangian method to solve the model. All of the experimental results indicate that Gaussian curvature obtains improved results over the mean curvature, linear curvature and demon methods for mono-modal image registration. The model can be extended to multi-modality image registration by changing the distance measure.

γ is the regularisation parameter and it controls the smoothness of the deformation field. We run experiments with various values of γ and fixing $r = 0.02$ for Test 1



(a) Model D

(b) Model LC

(c) Model MC

(d) Gaussian curvature

Figure 6.7: Test 2: A pair of Brain MR images. Comparison of Gaussian curvature with competing methods. The transformed template image using (a) Model D, (b) Model LC, (c) Model MC, and (d) Gaussian curvature. Notice the differences of these three images inside the red boxes. Considerably more accurate results are obtained, particularly within these significant regions, by employment of the Gaussian curvature model.

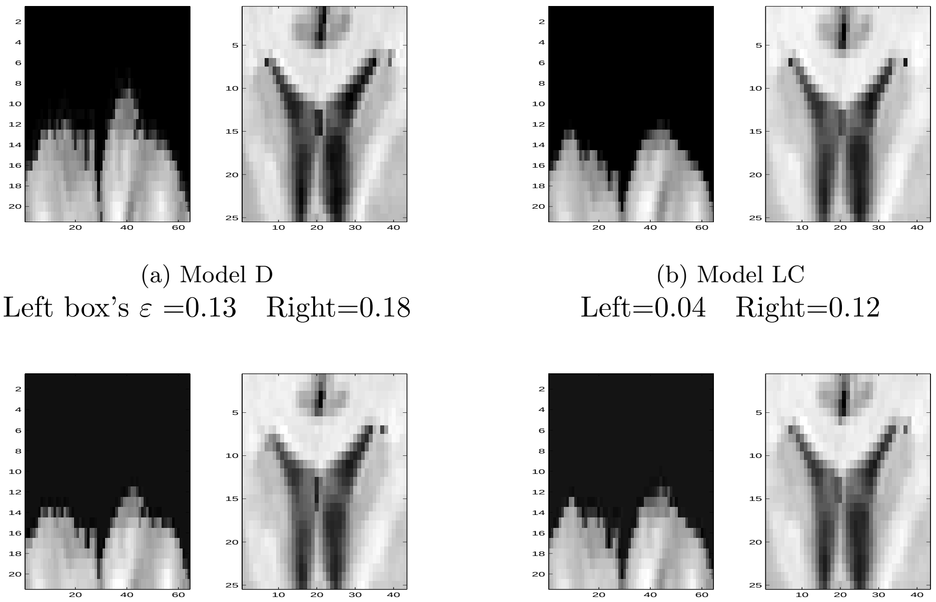
as shown in Figure 6.9. From the figure, with an optimal value of r, decreasing γ will decrease the value of $\mathcal{F}$ and ε until a value $\gamma = \gamma^*$ (in Figure 6.9, $\gamma^* = 0.0001$). Decreasing γ, $\gamma < \gamma^*$ has no affect on $\mathcal{F}$ and ε. We also observe how the functional $\mathcal{J}$ in equation (6.13) evolves during the iterations. The result is shown in Figure 6.9 (c) for Test 1 using $\gamma = 0.0001$ and $r = 0.02$. The functional $\mathcal{J}$ and the fitting term $\mathcal{D}$ are decreasing and the regularisation term $\mathcal{S}^{\text{GC}}$ is increasing, indicating the convergence of the model.

One of the main aspects in solving the Gaussian curvature model using ALM is the parameter r. The parameter stabilises the minimisation problem by introducing a quadratic energy on the distance between ∇u_l and $\boldsymbol{q}_l$. A bigger value of r will bring ∇u_l and $\boldsymbol{q}_l$ close together and produce a higher value of dissimilarity between R and T, as shown in Figure 6.10. It also controls the smoothness of the deformation field.

In Figure 6.10, we use Test 1, fix $\gamma = 0.008$ and vary the value of $r, r \in [0.002, 2]$. Define

$$n_1 = \frac{1}{|\Omega|} \text{ mean } (\boldsymbol{q}_1 - \nabla u_1, \boldsymbol{q}_2 - \nabla u_2) \tag{6.29}$$

and n_2 as the average residual of equations (6.23), (6.24) and (6.25). We can observe



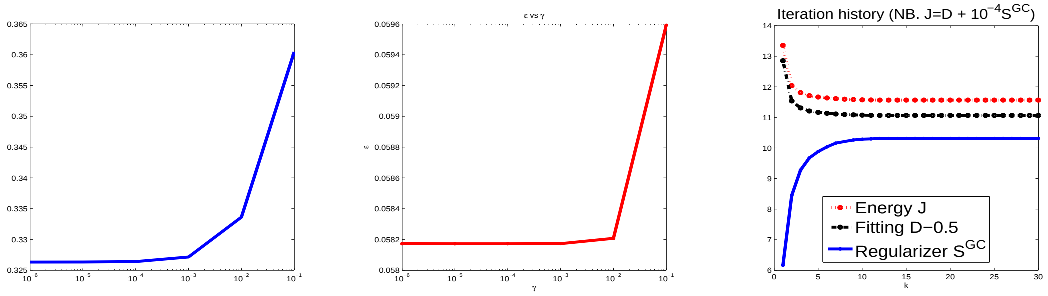
(a) Model D
Left box's ε =0.13   Right=0.18

(b) Model LC
Left=0.04   Right=0.12

(c) Model MC
Left box's ε =0.07   Right=0.20

(d) Gaussian curvature
Left=0.03   Right=0.11

Figure 6.8: Test 2: A pair of Brain MR images. Comparison of transformed templates in zoomed-in boxes and their local ε values: (a) Model D, (b) Model LC, (c) Model MC and (d) Gaussian curvature. Again Gaussian curvature has the smallest ε value.



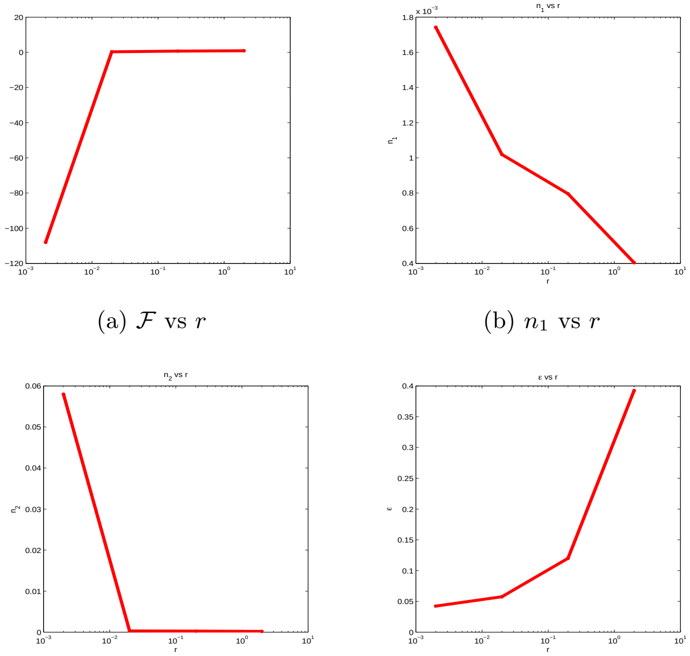
(a) $\mathcal{F}$ vs γ

(b) ε vs γ

(c) Iteration history

Figure 6.9: The effects on the values of $\mathcal{F}$ and ε for various values of γ are shown in (a) and (b). We obtain these figures using $r = 0.02$ for Test 1 and it confirms that γ controls the smoothness of the deformation field. The iteration history for Test 1 is shown in (c). Since the functional $\mathcal{J}$ is decreasing, the convergence of the proposed model is confirmed.

that with a very small r, the residual increases and produces mesh folding even when the value of ε is small. Thus, we required an optimal value of r such that ε is small and $\mathcal{F} > 0$. We start with a large value of r, for example $r = 2$, and check n_2. If n_2 is close to zero we reduce r by a factor of 10. Otherwise we increase r. This procedure was done on a coarse grid (16×16), with a small number of iterations. Thus, the computational



(a) $\mathcal{F}$ vs r

(b) n_1 vs r

(c) n_2 vs r

(d) ε vs r

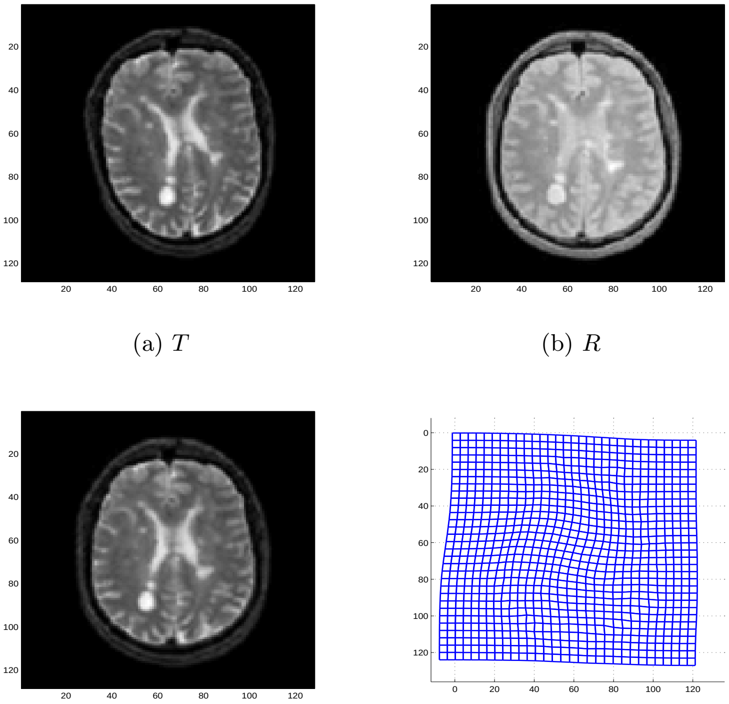
Figure 6.10: The effects on the value of $\mathcal{F}, n_1, n_2$ and ε for various values of r. In (a), $\mathcal{F}$ decreases with decreasing value of r. We should use the value of r, such that $\mathcal{F} > 0$, to avoid mesh folding. In (b), we can see that increasing the value of r will decrease the difference between $\boldsymbol{q}_1, \boldsymbol{q}_2$ and $\nabla u_1, \nabla u_2$. From (c), with a large value of r, we have smaller residual indicated by n_2. In (d), although small $r = 0.002$, gives a very small ε, but since $\mathcal{F} < 0$ for this value of r, we choose the optimal value of r to be $r = 0.02$.

cost is low. For Tests 1 and 2, we obtain $r = 0.02$ through this procedure.

The linear curvature model is based on the approximation of the mean curvature. The mean curvature model for image registration is highly nonlinear making the model difficult to solve. In contrast, we use the Gaussian curvature term without any approximation with an efficient numerical solver for the model. The diffeormorphic demon model is equivalent to the second order gradient descent on the SSD as shown in [74]. The model is limited to mono-modality images and it is not directly applicable to multi-modality images. The proposed model however can be easily modified to work with multi-modality images by changing the distance measure $\mathcal{D}(T, R, \boldsymbol{\varphi})$ from the SSD with mutual information or the normalised gradient field. We show results for multi-modality images in Figure 6.11 for the Gaussian curvature model with mutual information as the distance measure.

6.6 Conclusion

We have introduced a novel regularisation term for non-parametric image registration based on the Gaussian curvature of the surface induced by the displacement field.



(a) T (b) R

(c) $T(\boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x}))$ (d) $\boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x})$

Figure 6.11: Results of Gaussian curvature image registration for multi-modality images. The model is able to register multi-modality images with mutual information as the distance measure.

The model can be effectively solved using the augmented Lagrangian and alternating minimisation methods. For comparison, we used three models: the linear curvature [25], the mean curvature [19] and the demon algorithm [95] for mono-modality images. Numerical experiments show that the proposed model delivers better results than the competing models.

Chapter 7

An Improved Model for Joint Segmentation and Registration

We have so far presented a decomposition model combining parametric and non-parametric image registration in Chapters 4 and 5, as the first part of our work. For the second part, we proposed a novel non-parametric image registration model based on Gaussian curvature in Chapter 6. Image segmentation and registration are two of the most challenging tasks in medical imaging. They are closely related because registration results will be affected by segmentation and vice versa. In this chapter, we present the third part of our main work which makes use of image registration and image segmentation. We present an improved model for joint segmentation and registration based on active contour without edges. The proposed model is motivated by [58] and linear curvature [24]. Numerical results show that the new model outperform s the existing model for the registration and segmentation of one or multiple objects in the image. The proposed model also improves registration results when the features inside the object are to be segmented, posing different kinds of deformations to the object itself.

7.1 Introduction

Image segmentation aims to separate objects or features in the image that have similar characteristics into different classes or sub-regions, via detection and visualisation of the contours of the objects in the images. Meanwhile, image registration is the process of finding a geometric transformation between images such that the template (target) images are aligned with the reference (source) images. In some fields, such as medical image processing, these two depend on each other and should be treated simultaneously in a joint framework. One important application of such combinations can be found in [31] and similar papers where atlases are constructed from magnetic resonance (MR) scans to analyse and understand brain tumour development The task of constructing the atlases requires alignment of the brain tumour MR scans to a common coordinate system and the automatic segmentation of the scans. According to [22], 25% of published work in medical imaging literature are joint segmentation and registration methods. In this chapter, we describe a new model for simultaneous segmentation and registration based on variational formulations. The idea of joining the tasks of segmentation and registration has been utilised by Guyader and Vese in [58] using a level set representation which aligns the contour of the template image and simultaneously segments the reference image. The method relates both problems using a segmentation model based on the active contour model without edges which is solved in terms of the displacement field. This chapter describes an improved segmentation and registration approach related to the model in [58].

The first work on variational model for joint region based segmentation and registration was proposed for rigid registration by Yezzi and his colleagues in [108]. Later, the work on segmentation and rigid registration was extended to non-rigid deformation in [91] where the model improves segmentation and registration for CT and MR images. The authors pointed out that the success of the model is dependent on how well the segmentation results manage to represent the images. Thus, the model is highly dependent on the segmentation results. These two works [108, 91] produce segmented images for both the reference and template images and the mapping from the template image to the reference image. In [99], the authors proposed a registration and segmentation model for multi-modality images using cross cumulative residual entropy as a distance measure for registration. To model the deformation, the authors [99] used a parametric model based on cubic B-spline and for segmentation, the piecewise constant Chan-Vese (CV) model [12]. However the model requires segmentation of the reference image and the work can be considered as registration driven by segmentation.

The Guyader and Vese (GV-JSR) model [58] that has coupled segmentation and registration uses the nonlinear elastic model to register the segmented template and reference images. The model manages to produce topology-preserving segmentation where the initial contour from the template image is deformed to the contour of the reference image without merging or breaking which is difficult to achieve by other segmentation methods. However, the model is limited to well-defined objects or features that have clear boundaries but without fine details. Another limitation of the model is that it cannot cope with multiple objects detected in one image. Here, our proposal is to improve the model for overcoming these two drawbacks. We will describe in the following an improved segmentation and registration approach related to the model in [58].

The contribution of this chapter is twofold. First, to improve the GV-JSR model for cases where the objects are with fine details, we add a weighted Heaviside sum of the squared difference (SSD) term in the GV-JSR model. Second, for better registration, invariant to the affine registration and which allows large deformation, we use the linear curvature model [25, 24] to replace the nonlinear elastic term in the GV-JSR model. In this way, there is no need for a pre-registration step to cater for affine linear transformation [25]. Beside the ability to recover affine linear transformation, the linear curvature model for registration also produces more smooth transformation than a nonlinear elastic model. It is well known that low order regularisation terms, such as nonlinear elasticity are less effective than high order ones such as linear curvature in producing smooth transformations [68, 67]. To the best of our knowledge, only diffusion, linear and nonlinear elastic models for non-parametric image registration have been used in the task of joining segmentation and registration.

The outline of this chapter is as follows. In Section 7.2 we review the task of joining segmentation and registration. In Section 7.3, we introduce our proposed new joint segmentation and registration (NJSR) model which improves the original GV-JSR model. We show in Section 7.4 some numerical tests including comparisons. Finally, we present our conclusion in Section 7.5.

7.2 Review of Joint Segmentation and Registration

Variational models for image processing have received lots of attention in the medical imaging community due to the robustness and accuracy of the models. This includes joining the tasks of segmentation and registration. In this section, we provide a brief review of variational formulation models for joint segmentation and registration. Before we proceed, we introduce the same notation as in Chapters 2 and 3. Let the reference and template images, $R, T : \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}$, as given compactly support functions and denote by $\boldsymbol{\varphi} = \boldsymbol{\varphi}(\boldsymbol{x}) : \Omega \rightarrow \mathbb{R}^2$, the unknown transformation aiming for $T(\boldsymbol{\varphi}(\boldsymbol{x})) \approx R(\boldsymbol{x})$ with $\boldsymbol{x} = (x_1, x_2)$. In the non-parametric (variational approach) image registration, the transformation is written as $\boldsymbol{\varphi}(\boldsymbol{x}) = \boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x})$. This transformation enables us to focus on the unknown displacement vector $\boldsymbol{u}(\boldsymbol{x}) = (u_1(\boldsymbol{x}), u_2(\boldsymbol{x}))$. Here $\boldsymbol{u}(\boldsymbol{x})$ is searched over admissible functions in the set $\mathcal{U}$, a linear subspace of a Hilbert space with Euclidean scalar product. Let Γ be a closed curve which distinguishes foreground and background for a two phase segmentation model. We use the level set of a Lipschitz function $\phi_0 : \Omega \rightarrow \mathbb{R}$ to represent Γ as its zero level set such that

$$\begin{cases} \Gamma = \partial\Omega_1 = \{(x_1, x_2) \in \Omega | \phi_0(x_1, x_2) = 0\}, \\ \text{inside}(\Gamma) = \Omega_1 = \{(x_1, x_2) \in \Omega | \phi_0(x_1, x_2) > 0\}, \\ \text{outside}(\Gamma) = \Omega_2 = \{(x_1, x_2) \in \Omega | \phi_0(x_1, x_2) < 0\}. \end{cases} \tag{7.1}$$

Introducing a regularised Heaviside function H_ϵ,

$$H_\epsilon(z) = \frac{1}{2}\left(1 + \frac{2}{\pi} \arctan \frac{z}{\epsilon}\right)$$

and its corresponding Delta function

$$\delta_\epsilon(z) = \frac{\mathrm{d}H_\epsilon(z)}{\mathrm{d}z} = \frac{\epsilon}{\pi(\epsilon^2 + z^2)}.$$

7.2.1 The Unal-Slabaugh Model [91]

The region based energy functional for joint segmentation and registration [91] can be written as

$$\begin{aligned}\min_{\phi_0(\boldsymbol{x}),\boldsymbol{u}(\boldsymbol{x})} \mathcal{J}(\phi_0(\boldsymbol{x}),\boldsymbol{u}(\boldsymbol{x})) = &\int_\Omega (T(\boldsymbol{x})-a_1)^2 H_\epsilon(\phi_0(\boldsymbol{x}))\,\mathrm{d}\boldsymbol{x} + \int_\Omega (T(\boldsymbol{x})-a_2)^2(1-H_\epsilon(\phi_0(\boldsymbol{x})))\,\mathrm{d}\boldsymbol{x}\\ &+ \int_\Omega \delta_\epsilon(\phi_0(\boldsymbol{x}))|\nabla\phi_0(\boldsymbol{x})|\,\mathrm{d}\boldsymbol{x}\\ &+ \int_\Omega (R(\boldsymbol{x})-c_1)^2 H_\epsilon(\phi_0(\boldsymbol{x}+\boldsymbol{u}(\boldsymbol{x})))\,\mathrm{d}\boldsymbol{x}\\ &+ \int_\Omega (R(\boldsymbol{x})-c_2)^2(1-H_\epsilon(\phi_0(\boldsymbol{x}+\boldsymbol{u}(\boldsymbol{x}))))\,\mathrm{d}\boldsymbol{x} + \gamma\int_\Omega |\nabla\boldsymbol{u}(\boldsymbol{x})|^2\,\mathrm{d}\boldsymbol{x}\end{aligned} \tag{7.2}$$

where $\phi_0(\boldsymbol{x})$ is the level set function as in equation (7.1) and $\boldsymbol{u}(\boldsymbol{x})$ is the displacement field. a_1 and a_2 are the average intensities inside and outside the zero level set $\phi_0(\boldsymbol{x})$ for the template image, and similarly for c_1 and c_2. Unal and Slabaugh used the H^1 seminorm to regularise the displacement field

$$\mathcal{S}^{\mathrm{diff}}(\boldsymbol{u}(\boldsymbol{x})) = \int_\Omega |\nabla\boldsymbol{u}(\boldsymbol{x})|^2\,\mathrm{d}\boldsymbol{x} = \int_\Omega |\nabla u_1(\boldsymbol{x})|^2 + |\nabla u_2(\boldsymbol{x})|^2\,\mathrm{d}\boldsymbol{x} \tag{7.3}$$

in (7.2) which is a diffusion regularisation term for non-parametric image registration. The model in (7.2) is solved using an explicit numerical scheme and will produce the displacement field and segmented images for both the template and reference.

7.2.2 The Schumacher et al. Model [85]

In the previous section, we reviewed the task of joining segmentation and registration using the diffusion model to regularise the displacement field $\boldsymbol{u}(\boldsymbol{x})$. In this section, we will review the task which uses the linear elastic model to control the smoothness of the displacement field $\boldsymbol{u}(\boldsymbol{x})$. Schumacher et al. [85] presents improved work in image registration using manual segmentation where the segmentation of the template images are given. Even though the author did not clearly mention that their work is joint segmentation and registration, their presented model uses active contours to generate the segmented image (mask) of the reference image and the displacement field by solving the problem

$$\min_{\boldsymbol{u}(\boldsymbol{x})} \mathcal{D}^{\mathrm{SSD}^{\mathrm{mix}}}(R,T,M_R,M_T,\boldsymbol{u}(\boldsymbol{x})) + \gamma\mathcal{S}^{\mathrm{LE}}(\boldsymbol{u}(\boldsymbol{x})) \tag{7.4}$$

where

$$\begin{aligned}\mathcal{D}^{\mathrm{SSD}^{\mathrm{mix}}}(R,T,M_R,M_T,\boldsymbol{u}) = \frac{1}{2}\int_\Omega &[T(\boldsymbol{x}+\boldsymbol{u}(\boldsymbol{x}))-R(\boldsymbol{x})]\\ &\cdot[M_T(\boldsymbol{x}+\boldsymbol{u}(\boldsymbol{x}))-M_R(\boldsymbol{x})]\cdot[M_A(\boldsymbol{x})]^2\,\mathrm{d}\boldsymbol{x}.\end{aligned} \tag{7.5}$$

M_T and M_R are the masks for the template and reference images, respectively. M_A is the third mask for suppressing an unwanted area. The definition of the M_T, M_R and M_A are as follows. Let $B_T^{(i)}, B_R^{(i)} \in \Omega, i = 1, 2, ..., m$ where m is the number of regions. It is assumed that $B_T^{(i)} \cap B_R^{(j)} = \varnothing$ for $i \neq j$. Let $b_i \in \mathbb{R}^+$ denote the weighting factor for $B_T^{(i)}$ and $B_R^{(i)}$, respectively. Define

$$M_T(\boldsymbol{x}) = \begin{cases} b_i, & \boldsymbol{x} \in B_T^{(i)}; \\ 1, & \text{otherwise} \end{cases},$$
$$M_R(\boldsymbol{x}) = \begin{cases} b_i + 1, & \boldsymbol{x} \in B_R^{(i)}; \\ 1, & \text{otherwise} \end{cases}, \tag{7.6}$$
$$M_A(\boldsymbol{x}) = \begin{cases} 0, & \boldsymbol{x} \in B_R^{(i)} \wedge b_i = 0; \\ 1, & \text{otherwise} \end{cases}.$$

Thus, $T(\boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x})) - R(\boldsymbol{x})$ is multiplied by

$$\begin{cases} -1, & \boldsymbol{x} \in B_T^{(i)} \wedge \boldsymbol{x} \in B_R^{(i)}; \\ b_i, & \boldsymbol{x} \notin B_T^{(i)} \wedge \boldsymbol{x} \in B_R^{(i)}; \\ -b_i, & \boldsymbol{x} \in B_T^{(i)} \wedge \boldsymbol{x} \notin B_R^{(i)}. \end{cases} \tag{7.7}$$

The authors use a linear elastic model for the regularisation term in (7.4) which is given by

$$\mathcal{S}^{\text{LE}}(\boldsymbol{u}(\boldsymbol{x})) = \int_\Omega \frac{\mu}{4} \sum_{l,m=1}^{2} \Big(\partial_{x_l} u_m + \partial_{x_m} u_l\Big)^2 + \frac{\lambda}{2}\Big(\text{div}\, \boldsymbol{u}\Big)^2 \mathrm{d}\boldsymbol{x} \tag{7.8}$$

where μ and λ are the Lame constants. The mask of the reference image can be generated automatically using the mask of the template image and a snake based segmentation scheme. See [85] for more details.

7.2.3 The GV-JSR Model [57]

A nonlinear elastic model for image registration was introduced in [107] to model large and smooth transformation for image registration. This motivates the authors in [58] to use it in the GV-JSR model. The GV-JSR model uses the initial given segmentation of the template image to find the geometric transformation of the template image and the segmentation of the reference image. In this section, we will review it before highlighting the disadvantages of the model. The segmentation of the template image is represented by the zero level line of $\phi_0(\boldsymbol{x})$. The target contour Γ which separates the foreground and background in the reference image is represented as the zero level line of $\phi_0(\boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x}))$ where $\boldsymbol{u}(\boldsymbol{x})$ is the displacement field. The joint functional for segmentation and registration in [57] is given by

$$\begin{aligned}\min_{c_1,c_2,\boldsymbol{u}(\boldsymbol{x})} \mathcal{J}(c_1,c_2,\boldsymbol{u}(\boldsymbol{x})) = \lambda_1 &\int_\Omega |R(\boldsymbol{x})-c_1|^2 H_\epsilon(\phi_0(\boldsymbol{x}+\boldsymbol{u}(\boldsymbol{x})))\,\mathrm{d}\boldsymbol{x}\\ &+\lambda_2\int_\Omega |R(\boldsymbol{x})-c_2|^2(1-H_\epsilon(\phi_0(\boldsymbol{x}+\boldsymbol{u}(\boldsymbol{x}))))\,\mathrm{d}\boldsymbol{x}\\ &+\alpha\boldsymbol{\mathcal{S}}^{\mathrm{NLE}}(\boldsymbol{p})+\alpha\beta\|\boldsymbol{p}-\nabla\boldsymbol{u}(\boldsymbol{x})\|^2\end{aligned} \tag{7.9}$$

where c_1 and c_2 are the average intensities inside and outside the curve Γ in the reference image which is represented by the zero level line as in equation (7.1). The variable $\boldsymbol{p}$ is the dual variable for $\nabla\boldsymbol{u}(\boldsymbol{x})$ for simplicity and to reduce the nonlinearity in the regularisation term. It is given by

$$\boldsymbol{p}=\begin{pmatrix} p_{11} & p_{12}\\ p_{21} & p_{22}\end{pmatrix}\approx\nabla\boldsymbol{u}(\boldsymbol{x})=\begin{pmatrix}\frac{\partial u_1}{\partial x_1} & \frac{\partial u_1}{\partial x_2}\\ \frac{\partial u_2}{\partial x_1} & \frac{\partial u_2}{\partial x_2}\end{pmatrix}. \tag{7.10}$$

The regularisation term in (7.9), denoted by $\boldsymbol{\mathcal{S}}^{\mathrm{NLE}}$, is the nonlinear elastic regularisation term for image registration based on Yanovsky et al. [107, 105, 62] which is given by

$$\begin{aligned}\boldsymbol{\mathcal{S}}^{\mathrm{NLE}}(\boldsymbol{p})=\int_\Omega &\frac{\lambda}{8}\left(2(p_{11}+p_{22})+p_{11}^2+p_{12}^2+p_{21}^2+p_{22}^2\right)^2\\ &+\frac{\mu}{4}\Big((2p_{11}+p_{11}^2+p_{21}^2)^2\\ &+(2p_{22}+p_{12}^2+p_{22}^2)^2+2(p_{12}+p_{21}+p_{11}p_{22}+p_{21}p_{22})^2\Big)\,\mathrm{d}\boldsymbol{x}\end{aligned} \tag{7.11}$$

where μ and λ are the Lame constants and the model uses the Dirichlet boundary condition.

The GV-JSR model [57, 58] is incorporated with the regridding step, thus it manages to recover large deformation. The idea of regridding was proposed by Christensen et al. [15] to model large deformation. The regridding step is as follows. The determinant of the Jacobian matrix of the transformation is calculated during the registration process to make sure there is no folding or cracking in the deformation field. If the minimum value of the determinant falls below a certain threshold, the last displacement field is stored and the template image is initialised using the last displacement field. Then, the displacement field is set to zero and the process continues until convergence. In [10], the authors extend the regridding concept and show how the method can be applied in the case of other regularisation terms such as diffusion, linear curvature and linear elastic with several types of boundary conditions. For example, to solve the famous large deformation problem where we want to align a letter C with a dot (refer to [67] for more details) the model requires two regridding steps. So, it is natural for any regularisation based models to recover large deformation as long as the regridding step is incorporated in the model.

One of the main advantages of the GV-JSR model is the ability to produce topology-preserving segmentation where the initial contour from the template image is deformed to the contour of the reference image without merging or breaking. The contour of the reference image is the deformed version of the contour of the template image using the found smooth transformation. It is deformed without separation of the initial contour from the template image which is difficult to achieve with the standard level set implementation of the active contour [58].

Topology preservation is important for several applications in medical imaging such as in computational brain anatomy. The GV-JSR model manages to preserve the topology of the initial contour without incorporation of soft or hard constraints in the model. Based on our tests however, we found that the model is only suitable for single objects in a well defined image with relatively large structures. The registration process is only driven by the forces on the boundary of the outer structures of the objects and does produce an incomplete deformation field for the inner structures of the objects. We also found that the GV-JSR model is unable to deliver good results when the template image consists of more than one object as shown using Test 4 in §4 where the template image consists of two homogeneous objects and both of them pose different kinds of deformation where one of them has larger deformation compared to the other.

7.3 The Proposed New Joint Segmentation and Registration (NJSR) Model

To deal with the two cases where the GV-JSR model fails to register, we propose to include two new terms in the functional (7.9). The first term is a SSD term of the form

$$\mathcal{D}^{\text{SSD}}(T,R,\boldsymbol{u}(\boldsymbol{x})) = \frac{1}{2}\int_\Omega (T(\boldsymbol{x}+\boldsymbol{u}(\boldsymbol{x})) - R(\boldsymbol{x}))^2 \,\mathrm{d}\boldsymbol{x} \tag{7.12}$$

which is weighted by the parameter λ_3 and the term $H_\epsilon(\phi_0(\boldsymbol{x}+\boldsymbol{u}(\boldsymbol{x})))$ and the second term is the linear curvature term to regularise the deformation field in the NJSR model. The sum of the squared difference in (7.12) is the optimal similarity measure of mono-modality images where the intensity values of the same objects in the reference and template images are equal. Thus our new NJSR model is the following

$$\begin{aligned}\min_{c_1,c_2,\boldsymbol{u}(\boldsymbol{x})} \mathcal{J}(c_1,c_2,\boldsymbol{u}(\boldsymbol{x})) = {} & \lambda_1\int_\Omega |R(\boldsymbol{x})-c_1|^2 H_\epsilon(\phi_0(\boldsymbol{x}+\boldsymbol{u}(\boldsymbol{x})))\,\mathrm{d}\boldsymbol{x} \\ & + \lambda_2\int_\Omega |R(\boldsymbol{x})-c_2|^2(1-H_\epsilon(\phi_0(\boldsymbol{x}+\boldsymbol{u}(\boldsymbol{x}))))\,\mathrm{d}\boldsymbol{x} \\ & + \mathcal{D}^{\text{SSDH}}(T,R,\phi_0(\boldsymbol{x}),\boldsymbol{u}(\boldsymbol{x})) + \alpha\mathcal{S}^{\text{LC}}(\boldsymbol{u})\end{aligned} \tag{7.13}$$

where

$$\mathcal{D}^{\text{SSDH}}(T,R,\phi_0(\boldsymbol{x}),\boldsymbol{u}(\boldsymbol{x})) = \lambda_3\int_\Omega (T((\boldsymbol{x}+\boldsymbol{u}(\boldsymbol{x}))-R(\boldsymbol{x})))^2 H_\epsilon(\phi_0(\boldsymbol{x}+\boldsymbol{u}(\boldsymbol{x})))\,\mathrm{d}\boldsymbol{x} \tag{7.14}$$

and

$$\mathcal{S}^{\text{LC}}(\boldsymbol{u}) = \int_\Omega (\Delta u_1)^2 + (\Delta u_2)^2 \,\mathrm{d}\boldsymbol{x}. \tag{7.15}$$

Now, we adopt the level set formulation, $\phi_0(\boldsymbol{x})$ to find the boundary Γ. c_1 and c_2 in (7.13) are the average intensity values inside and outside the boundary Γ in the reference image. To update c_1 and c_2, minimise equation (7.13) to obtain,

$$\begin{aligned} c_1 &= \frac{\int R(\boldsymbol{x}) H_\epsilon(\phi_0(\boldsymbol{x}+\boldsymbol{u}(\boldsymbol{x})))\,\mathrm{d}\boldsymbol{x}}{\int H_\epsilon(\phi_0(\boldsymbol{x}+\boldsymbol{u}(\boldsymbol{x})))\,\mathrm{d}\boldsymbol{x}}, \\ c_2 &= \frac{\int R(\boldsymbol{x})(1-H_\epsilon(\phi_0(\boldsymbol{x}+\boldsymbol{u}(\boldsymbol{x}))))\,\mathrm{d}\boldsymbol{x}}{\int 1-H_\epsilon(\phi_0(\boldsymbol{x}+\boldsymbol{u}(\boldsymbol{x})))\,\mathrm{d}\boldsymbol{x}}. \end{aligned} \tag{7.16}$$

To update $\boldsymbol{u}(\boldsymbol{x})$, we can solve the functional in (7.13) by either the optimise then discretise approach (i.e. the Euler Lagrange equation to be discretised by a numerical method) or the discretise then optimise approach (i.e. the discretised functional to be optimised). From either of these approaches, we obtain a nonlinear system of equations, which we solve iteratively to yield the final solution. Below, we adopt the latter approach i.e. the functional in (7.13) is solved with respect to the displacement field $\boldsymbol{u}(\boldsymbol{x})$ using a discretise then optimise approach based on the quasi-Newton method in a multilevel framework for faster implementation.

Similarly to Chapter 2, the grid points are located at the centre of the cell

$$\Omega^h = \{\boldsymbol{x}_{i,j} = (x_{1,i}, x_{2,j}) = ((i-0.5)h, (j-0.5)h) | 1 \le i, j \le N\}, \tag{7.17}$$

where the domain Ω^h is split into $N \times N$ cells of size $h \times h$. We shall re-use the notation T, R for discrete images of size $N \times N$. We re-define the solution vector

$$\boldsymbol{U} = \begin{bmatrix} \boldsymbol{u}_1 \\ \boldsymbol{u}_2 \end{bmatrix}_{2N^2\times 1}, \boldsymbol{x} = \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \end{bmatrix}_{2N^2\times 1}, \tag{7.18}$$

where

$$\boldsymbol{u}_1 = \begin{bmatrix} u_{1,1,1} & u_{1,2,1} & \cdots & u_{1,N,1} & u_{1,1,1} & \cdots & u_{1,N,1} & u_{1,1,2} & \cdots & u_{1,N,N} \end{bmatrix}^T,$$

$$\boldsymbol{u}_2 = \begin{bmatrix} u_{2,1,1} & u_{2,2,1} & \cdots & u_{2,N,1} & u_{2,1,1} & \cdots & u_{2,N,1} & u_{2,1,2} & \cdots & u_{2,N,N} \end{bmatrix}^T$$

and $\boldsymbol{x}_1, \boldsymbol{x}_2$ are similarly defined.

The discretised form of the functional in (7.13), by a finite difference method is

$$\begin{aligned}\min_{c_1,c_2,\boldsymbol{U}} \mathcal{J}^h(c_1,c_2,\boldsymbol{U}) = \lambda_1 &\sum_{i,j=1}^{N} |R(\boldsymbol{x}_{i,j}) - c_1|^2 H_\epsilon(\phi_0(\boldsymbol{x}_{i,j} + \boldsymbol{u}(\boldsymbol{x}_{i,j}))) \\ + \lambda_2 &\sum_{i,j=1}^{N} |R(\boldsymbol{x}_{i,j}) - c_2|^2 (1 - H_\epsilon(\phi_0(\boldsymbol{x}_{i,j} + \boldsymbol{u}(\boldsymbol{x}_{i,j})))) \\ + \lambda_3 &\sum_{i,j=1}^{N} (T((\boldsymbol{x}_{i,j} + \boldsymbol{u}(\boldsymbol{x}_{i,j})) - R(\boldsymbol{x}_{i,j})))^2 H_\epsilon(\phi_0(\boldsymbol{x}_{i,j} + \boldsymbol{u}(\boldsymbol{x}_{i,j}))) \\ + \frac{\alpha}{2} &\sum_{l=1}^{2} \sum_{i,j=1}^{N} \Big(-4u_l(\boldsymbol{x}_{i,j}) + u_l(\boldsymbol{x}_{i+1,j}) + u_l(\boldsymbol{x}_{i-1,j}) + u_l(\boldsymbol{x}_{i,j+1}) + u_l(\boldsymbol{x}_{i,j-1}) \Big)^2 .\end{aligned} \tag{7.19}$$

Here, we are using homogeneous Neumann boundary conditions approximated by one side differences

$$u_l(\boldsymbol{x}_{i,1}) = u_l(\boldsymbol{x}_{i,2}), u_l(\boldsymbol{x}_{1,j}) = u_l(\boldsymbol{x}_{2,j}), u_l(\boldsymbol{x}_{i,N-1}) = u_l(\boldsymbol{x}_{i,N}), u_l(\boldsymbol{x}_{N-1,j}) = u_l(\boldsymbol{x}_{N,j}), l = 1, 2. \tag{7.20}$$

Starting with a zero initial guess,

$$\boldsymbol{U} = \boldsymbol{0}, \tag{7.21}$$

we solve

$$\boldsymbol{H}\delta\boldsymbol{U} = -\boldsymbol{G} \tag{7.22}$$

for $\delta\boldsymbol{U}$ and update $\boldsymbol{U} \leftarrow \boldsymbol{U} + \tau\delta\boldsymbol{U}$ with τ as the Armijo line search parameter [102]. $\boldsymbol{H}$ and $\boldsymbol{G}$ are the Hessian and gradient matrix for the functional $\mathcal{J}^h$ in equation (7.19) with respect to the displacement vector $\boldsymbol{U}$. The algorithm for the proposed model is given in Algorithm 10 where we obtain multilevel representation of the reference and template images denoted by $T^{\text{level}}, R^{\text{level}}$ using standard coarsening in the implementation. We also obtain multilevel representation of the surface $\phi_0(\boldsymbol{x})$ which represents the contour Γ of the template image. The coarsest and finest levels of images are denoted by Minlevel

Algorithm 10 The NJSR model for joint segmentation and registration.

```
1. Initialisation:
       $R, T, \alpha, \lambda_1, \lambda_2, \lambda_3, \boldsymbol{U} = \boldsymbol{0}, \phi_0(\boldsymbol{x})$.
2. For level = Minlevel, ..., Maxlevel
    (a) Solve registration problem on this level using quasi-Newton method,
            $\boldsymbol{U}^{\text{level}} \leftarrow \text{Register}(T^{\text{level}}, R^{\text{level}}, \phi_0^{\text{level}}, \boldsymbol{U}^{\text{level},0})$.        (7.23)
    (b) If level < Maxlevel, interpolate $\boldsymbol{U}^{\text{level}}$ to the next finer level.
3. End for.
```

and Maxlevel respectively. We start with zero initial guess for the displacement field on the Minlevel. After registration on each level, the deformation field $\boldsymbol{U}^{\text{level}}$ is interpolated to the next finer level (level = level + 1) using bilinear interpolation. These recursive procedures are perform iteratively until we reach level = Maxlevel.

7.4 Numerical Results

We use four sets of images for testing the GV-JSR model and the NJSR model (Algorithm 10) on a variety of images and deformations. To judge the quality of the registration we calculate the relative reduction of the similarity measure

$$\varepsilon = \frac{\mathcal{D}^{\text{SSD}}(T, R, \boldsymbol{u}(\boldsymbol{x})^{(*)})}{\mathcal{D}^{\text{SSD}}(T, R, \boldsymbol{u}(\boldsymbol{x})^{(0)})}. \tag{7.24}$$

In all of the tests, we do not use the regridding step for fair comparison and the value of the regularisation parameters are chosen such that the minimum value of the determinant of the Jacobian matrix J of the transformation, denoted as $\mathcal{F}$

$$J = \begin{bmatrix} 1 + \frac{\partial u_1}{\partial x_1} & \frac{\partial u_1}{\partial x_2} \\ \frac{\partial u_2}{\partial x_1} & 1 + \frac{\partial u_1}{\partial x_2} \end{bmatrix}, \quad \mathcal{F} = \min(\det(J)), \tag{7.25}$$

is greater than zero. This indicates that the deformed grid obtained from the displacement field is free from folding and cracking. Details of the tests are:

- Test 1(One Feature with GV-JSR Model). Test 1 consists of two X-ray images of a human hand from [68] to illustrate the type of images where the GV-JSR model is able to segment and register. The images in Test 1 consist of one object with relatively large structure.

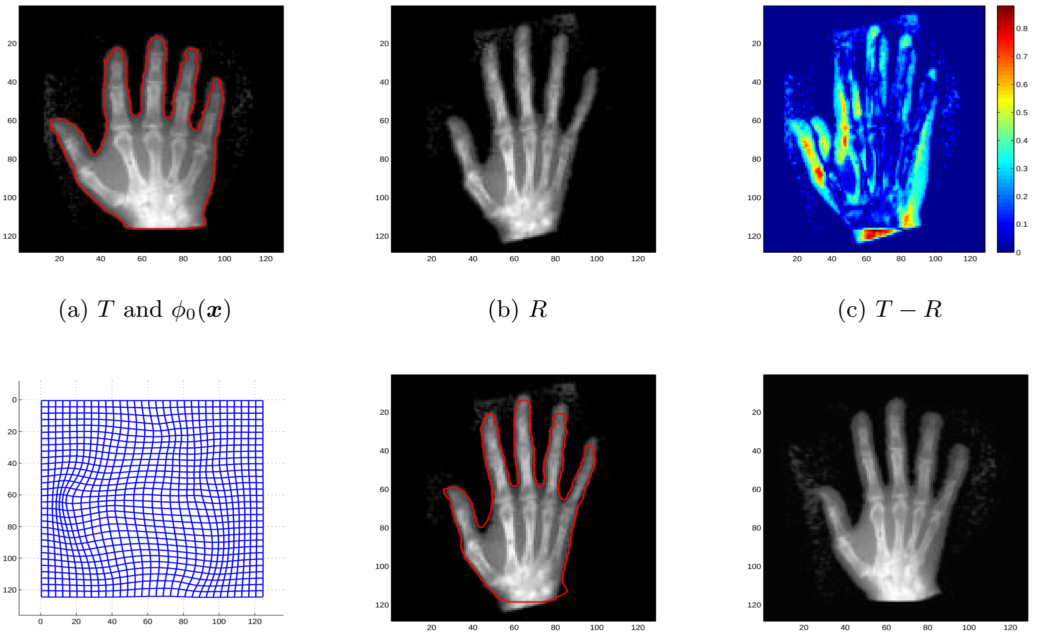
- Test 2 (Global Deformation with GV-JSR Model). The images for Test 2 come from [46] where the GV-JSR model manages to deliver good results because the features inside the objects in the template image pose the same deformation with the boundary of the object to be segmented.

- Test 3 (Local Deformation with GV-JSR and NJSR Models). Test 3 is used to illustrate images where the GV-JSR model fails to provide the deformation field between the reference and template images where the data set is from [40]. In this test, the features inside the contour pose different kinds of deformation with the contour. Since the GV-JSR model is based on the boundary mapping, we obtain no alignment for the features inside the contour Γ. Note that the outer structure is nicely registered whereas the inner structure is poorly registered. We show that our proposed model, NJSR, is able to solve Test 3 which involves different kinds of deformation for the boundary (contour) of the object and the features inside the contour.

- Test 4 (Case of More than One Object). The second class of problems where the GV-JSR model fails to deliver good results is shown in Test 4 where two objects are detected in the template image. These two objects have different kinds of deformations where one of them has larger deformation compared to other one.

In all tests we use $\lambda_1 = \lambda_2 = 250, \lambda = 0.5, \mu = 0.005$ for the GV-JSR model in a single level implementation which are the best parameters value for the model. We solve the GV-JSR model based on the numerical solver provided in [57] without the regridding step. Meanwhile, for the NJSR model, we use $\lambda_1 = \lambda_2 = \lambda_3 = 1$ and $\alpha = 0.25$ and $\alpha = 1$ for Tests 3 and 4, respectively.

7.4.1 Test 1: One Feature with GV-JSR Model



(a) T and $\phi_0(\boldsymbol{x})$

(b) R

(c) $T - R$

(d) $\boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x}), \mathcal{F} = 0.4790$

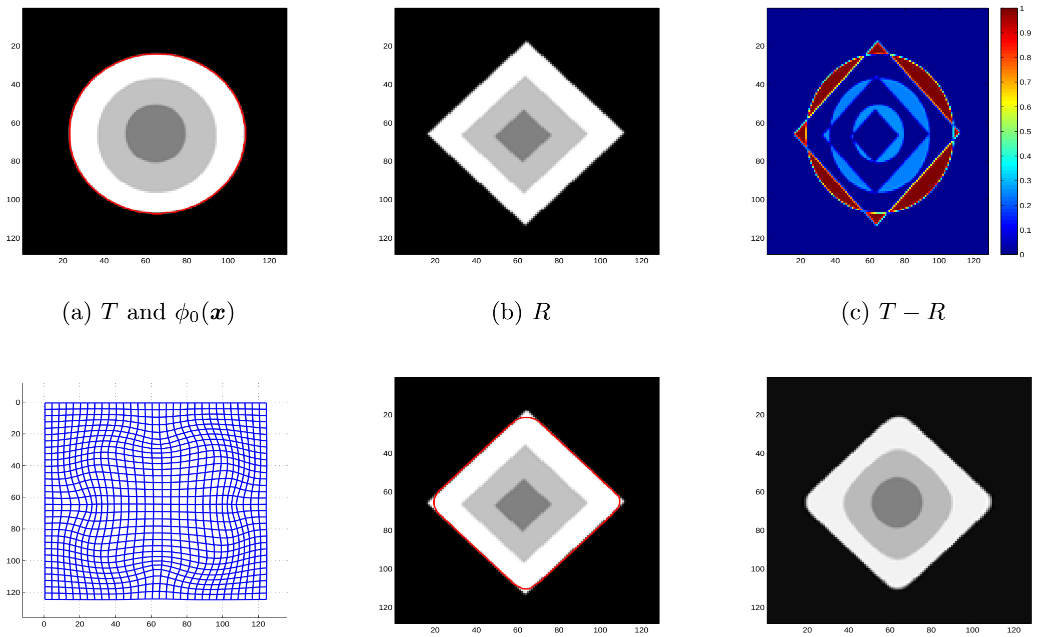
(e) R and $\phi_0(\boldsymbol{x} + \boldsymbol{u})$

(f) $T(\boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x})), \varepsilon = 0.2343$

Figure 7.1: Test 1: GV-JSR model. Illustration of the type of images where the GV-JSR model delivers good results where the object to be segmented in the template image is relatively large. The results obtained in this test are for $\alpha = \beta = 25$.

Images for Test 1 are the same as [68] where X-ray images of two hands of different individuals need to be aligned. The size of the images is 128×128 and the recovered transformation is expected to be smooth. For this test we take $\alpha = \beta = 25$. We show the results of Test 1 obtained by the GV-JSR model in Figure 7.1. The template image and the zero level set of Γ are shown in red in Figure 7.1 (a). The resulting deformation field is shown in Figure 7.1 (d) with the value of $\mathcal{F} = 0.4790$. The zero level of $\phi_0(\boldsymbol{x}+\boldsymbol{u})$ is shown in red with the reference image in Figure 7.1 (e). The model uses Dirichlet boundary conditions which explains why the lower part of the hand is not aligned as shown in Figure 7.1 (f) with the value of $\varepsilon = 0.2343$. In this test, the object inside Γ exhibits the same deformation as Γ, so the GV-JSR model manages to deliver an acceptable level of results.

7.4.2 Test 2: Global Deformation with GV-JSR Model



(a) T and $\phi_0(\boldsymbol{x})$

(b) R

(c) $T - R$

(d) $\boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x}), \mathcal{F} = 0.7424$

(e) R and $\phi_0(\boldsymbol{x} + \boldsymbol{u})$

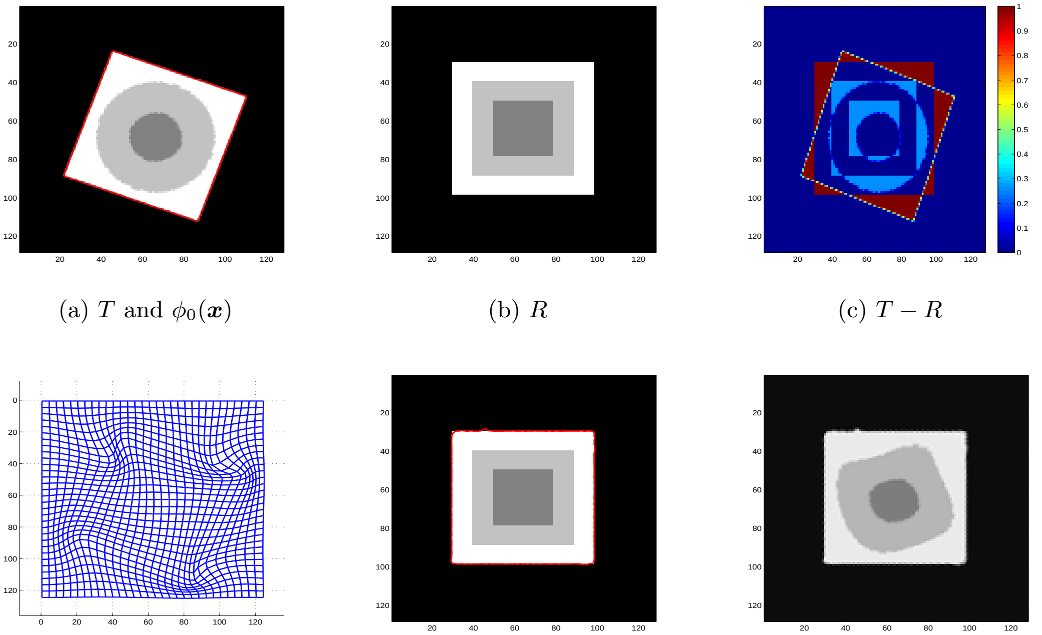
(f) $T(\boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x})), \varepsilon = 0.0518$

Figure 7.2: Test 2: GV-JSR model. Illustration of the second class of problems where the GV-JSR model manages to provide good results where the deformation of the features inside the object to be segmented pose the same deformation as the object itself.

Synthetic images for Test 2 from [46] are used to illustrate cases where the features inside the object have the same deformation as the boundary of the object. The results of Test 2 using the GV-JSR model with $\alpha = \beta = 25$, are shown in Figure 7.2. The template image and the zero level set of Γ in red are shown in Figure 7.2 (a). The resulting deformation field is shown in Figure 7.2 (d) with $\mathcal{F} = 0.7424$. The zero level set of $\phi_0(\boldsymbol{x}+\boldsymbol{u})$ is shown in red with the reference image in Figure 7.2 (e). The resulting transformed template image using the deformation in (d) is shown in Figure 7.2 (f) with $\varepsilon = 0.0518$. In this problem, the object inside Γ exhibits the same deformation as Γ, thus the GV-JSR model manages to deliver an acceptable level of results.

7.4.3 Test 3: Local Deformation with GV-JSR and NJSR Models

In Test 3, we use the images in Figure 7.3 to illustrate where the GV-JSR model with $\alpha = 5$ and $\beta = 25$ fails to deliver good results. In the figure, we can observe that



(a) T and $\phi_0(\boldsymbol{x})$

(b) R

(c) $T - R$

(d) $\boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x}), \mathcal{F} = 0.3319$

(e) R and $\phi_0(\boldsymbol{x} + \boldsymbol{u})$

(f) $T(\boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x})), \varepsilon = 0.0509$

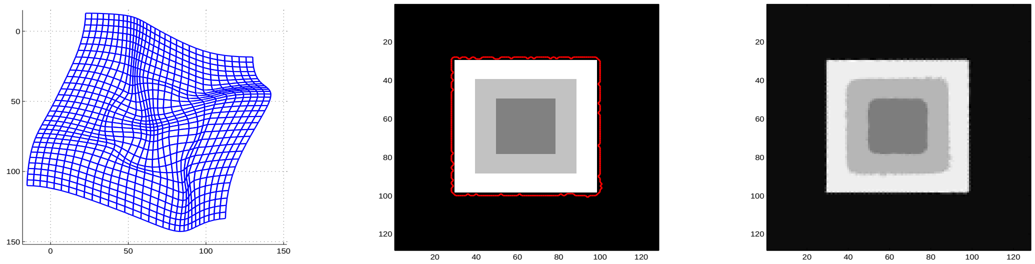
Figure 7.3: Test 3: GV-JSR model. Illustration of the type of image which has different deformation for the boundary Γ and the features inside Γ. The GV-JSR model fails to align the features inside Γ but manages to align the outer most square in the template image. In this test we are using $\alpha = 5$ and $\beta = 25$.

the deformation inside Γ is different from the deformation of Γ. We can see in Figure 7.3 (f), the resulting transformed template image contains a huge difference with the reference image in (b) for the inner squares. However, the model manages to align the outermost square. In the figure, we have $\mathcal{F} = 0.3319$ and $\varepsilon = 0.0509$.

We resolve the issues in Test 3 by using the NJSR model, and the resulting images are depicted in Figure 7.4. In this figure, we obtain the segmentation of the reference image as shown in Figure 7.4 (b). Since the NJSR model uses the linear curvature model for registration which contains an affine linear transformation, it manages to recover the rotation part of the deformation without affine pre-registration step as shown in Figure 7.4 (a) with $\mathcal{F} = 0.3004$. The resulting transformed template image, shown in Figure 7.4 (c), has better alignment with the reference image in Figure 7.3 (b) compared to the one obtained by the GV-JSR model in Figure 7.3 (f). In this test, we have $\varepsilon = 0.0062$ which is lower than the one obtain from the GV-JSR model in Figure 7.3 (f).

7.4.4 Test 4: Case of More than One Object

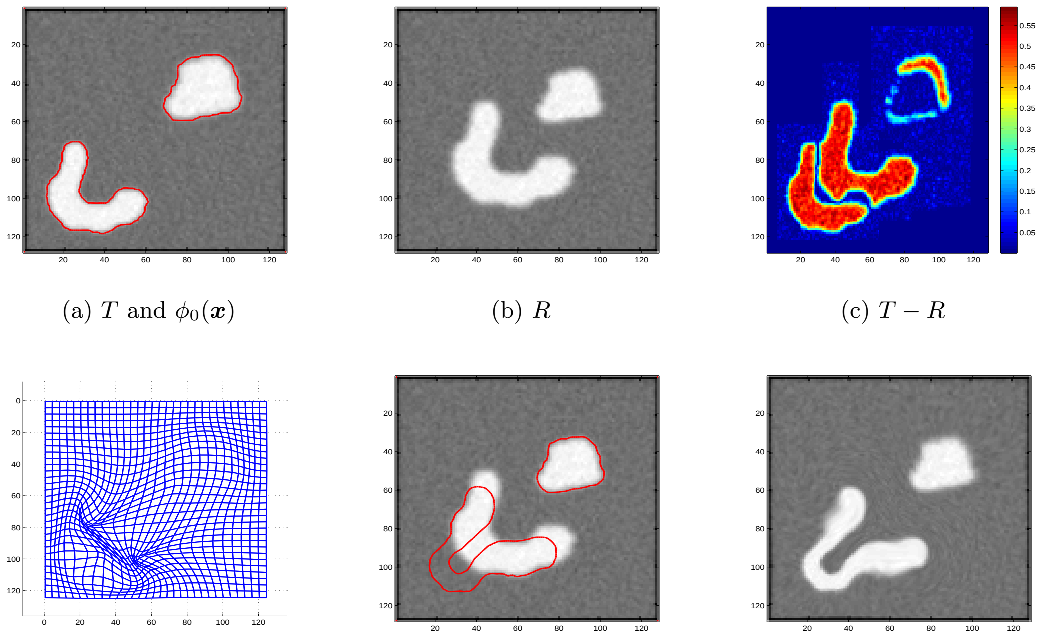
In Test 4, we have two objects in the template image, as shown in Figure 7.5 (a), and they have different kinds of deformation where the big one at the left hand side has



(a) $\boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x}), \mathcal{F} = 0.3004$ (b) R and $\phi_0(\boldsymbol{x} + \boldsymbol{u})$ (c) $T(\boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x})), \varepsilon = 0.0062$

Figure 7.4: Test 3: NJSR model. We have better results using the NJSR model for Test 3 where the circles in T are deformed to squares as in R. We also have smaller value of $\varepsilon = 0.0062$ for the NJSR model than $\varepsilon = 0.0509$ which is obtained from the GV-JSR model.

scaling and translation. Meanwhile, the smaller object at the top right only requires translation. The reference image for this particular test is shown in Figure 7.5 (b) and



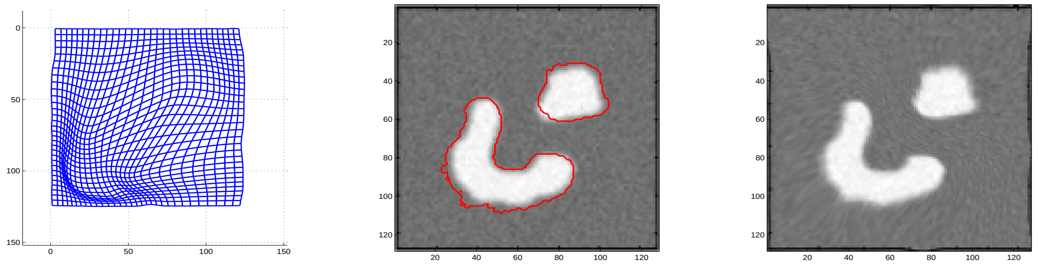
(a) T and $\phi_0(\boldsymbol{x})$ (b) R (c) $T - R$

(d) $\boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x}), \mathcal{F} = 0.0100$ (e) R and $\phi_0(\boldsymbol{x} + \boldsymbol{u})$ (f) $T(\boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x})), \varepsilon = 0.3382$

Figure 7.5: Test 4: GV-JSR model. The model fails when there is more than one feature detected in the template image which is indicated by a large value of $\varepsilon = 0.3382$.

it was chosen before discretisation of the problem. We can also interchange between the template and reference images. The results using the GV-JSR model with $\alpha = 5$ and $\beta = 25$ are shown in Figures 7.5 (d), (e) and (f). We can observe that the model did not manage to deliver good alignment between the transformed template shown in (f) and the reference image shown in (b). The deformation field applied on the regular grid is shown in Figure 7.5 (d) with $\mathcal{F} = 0.0100$ and $\varepsilon = 0.3382$.

We resolve the issues in Test 4 using the NJSR model and the resulting images are depicted in Figure 7.6. In this figure, we obtained better segmentation of the reference image as shown in Figure 7.6 (e). Since the NJSR model uses the linear curvature model for registration and Neumann boundary conditions, it manages to recover large deformation without regridding as shown in Figure 7.6 (d). The resulting transformed template image, shown in Figure 7.6 (f), has better alignment with the reference image in Figure 7.6 (c) compared to the one obtained by the GV-JSR model in Figure 7.5 (f). In this test we have $\mathcal{F} = 0.1330$ and $\varepsilon = 0.0797$. The small value of ε obtained from the NJSR model indicates higher similarity between the reference and the transformed template image compare to the one obtained from GV-JSR model in Figure 7.5 (f).



(a) $\boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x}), \mathcal{F} = 0.1330$ (b) R and $\phi_0(\boldsymbol{x} + \boldsymbol{u})$ (c) $T(\boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x})), \varepsilon = 0.0797$

Figure 7.6: Test 4: NJSR model. Our proposed model manages to segment and register this particular kind of problem where we have more than one homogeneous object in the template image. We obtain a smaller value of $\varepsilon = 0.0239$ from the NJSR model than $\varepsilon = 0.3382$ which is obtained from the GV-JSR model.

7.5 Conclusion

We have presented an improved model for joint segmentation and registration in a variational formulation. The proposed model consists of two new terms which extend the original Guyader and Vese (GV-JSR) model's applicability. The first term is a weighted SSD with a regularised Heaviside of the zero level set function to quantify the different deformations exhibited by the features inside of the contour of the template image. The second term is the linear curvature term to control the smoothness of the deformation field which is superior to the non-linear elastic term in the old GV-JSR model. The new NJSR model is particularly effective when more than one object is detected in the image.

Chapter 8

Conclusion and Future Research

This thesis has showed the author's work on the mathematical modelling of three effective models for image registration. Numerical methods for solving these models are also presented in this thesis to show the effectiveness of the proposed models. The first model given is the decomposition model combining parametric and non-parametric deformation for mono and multi-modalities images. The second model demonstrated in this thesis is the Gaussian curvature model for non-parametric image registration, in Chapter 6. The presented model is a novel approach using Gaussian curvature for mono-modality images. We demonstrated an augmented Lagrangian method to solve the model. The third model given in this thesis is an improved model for the task of joint segmentation and registration, in Chapter 7.

8.1 Conclusion

First, in Chapter 4, we have proposed a decomposition model combining parametric and non-parametric image registration models. As such, we choose cubic B-spline and linear curvature models for the parametric and non-parametric parts respectively. To choose the regularisation parameter, we proposed a continuation approach based on the determinant of the Jacobian matrix of the transformation. As a result, we found that the decomposition model performs better than the individual models for mono-modality images.

Second, in Chapter 5, we extended the decomposition model, in Chapter 4, to multi-modality images where images come from different imaging machines. Thus, the sum of squared difference of the intensity values is no longer valid in the minimisation problem. We test mutual information and the normalised gradient field as the new similarity measures with the decomposition model. We found that the normalised gradient field works better than mutual information in some cases. However, both models are at a disadvantage for cases where there is a strong bias field in the images.

Third, we present a novel method for non-parametric image registration using Gaussian curvature. A new effective numerical solver for the model is presented which is based on the augmented Lagrangian method. From our numerical experiments, we concluded that Gaussian curvature outperforms three competing models, which are linear and mean curvature and diffeomorphic demon models, in terms of robustness and accuracy.

Finally, in Chapter 7, we combine the tasks of segmentation and registration in a joint framework. We present an improved model for joint segmentation and registration based on an active contour without edges and linear curvature model for image registration. Numerical results showed that the new model outperforms the existing model for joint segmentation and registration of one and multiple objects in the image. The model also improves registration results when the features inside the object pose different kinds of deformations of the object itself.

8.2 Future Directions

There are many different future directions that can be taken from the work presented in this thesis. We mention some of them here:

- In Chapter 4 we proposed a decomposition model using cubic B-spline and linear curvature. We may extend the model to include landmark registration based models.

- We extended the decomposition model in Chapter 5 to multi-modality images. Possible future research would be to apply the model using different distance measures and incorporating soft or hard constraints to deal with bias field images.

- We proposed a novel regularisation term for image registration using Gaussian curvature. Although we are able to obtain good results, it would interesting to solve registration with image denoising using Gaussian curvature for both problems.

- The model presented in Chapter 6 may be easily extended to three dimensional images. At the same time, there are possibilities for developing other novel numerical methods to solve the model.

- A possible future direction for Chapter 7 may be to develop a selective segmentation based model for joint segmentation and registration.

Bibliography

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

[2] R. Bajcsy and S. Kovačič. Multiresolution elastic matching. *Comput. Vision Graph.*, 46(1):1–21, 1989.

[3] M. Bertero, T. A. Poggio, and V. Torre. Ill-posed problems in early vision. *Proceedings of the IEEE*, 76(8):869–889, 1988.

[4] R. S. Borden. *A course in advanced calculus.* Courier Corporation, 2012.

[5] W. L. Briggs and S. F. McCormick. *A multigrid tutorial.* Society for Industrial and Applied Mathematics, 2000.

[6] C. Broit. *Optimal registration of deformed images.* PhD thesis, University of Pennsylvania, 1981.

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

[8] M. Burger, J. Modersitzki, and D. Tenbrinck. Mathematical methods in biomedical imaging. *GAMMMitteilungen*, 37(2):154–183, 2014.

[9] P. Cachier, X. Pennec, and N. Ayache. Fast Non Rigid Matching by Gradient Descent: Study and Improvements of the "Demons" Algorithm. Technical Report RR-3706, 1999.

[10] N. D. Cahill, J. A. Noble, and D. J. Hawkes. Fourier methods for nonparametric image registration. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[11] N. D. Cahill, J. A. Noble, and D. J. Hawkes. Demons algorithms for fluid and curvature registration. In *IEEE International Symposium on Biomedical Imaging: From Nano to Macro, ISBI '09.*, pages 730–733, 2009.

[12] T. Chan and L. Vese. *An active contour model without edges*, pages 141–151. Springer, 1999.

[13] K. Chen. *Matrix preconditioning techniques and applications*. Number 19. Cambridge University Press, 2005.

[14] G. E. Christensen, S. C. Joshi, and M. I. Miller. Volumetric transformation of brain anatomy. *IEEE T. Med. Imaging*, 16(6):864–877, 1997.

[15] G. E. Christensen, R. D. Rabbitt, and M. I. Miller. Deformable templates using large deformation kinematics. *IEEE T. Med. Imaging*, 5(10):1435–1447, 1996.

[16] N. Chumchob and K. Chen. A robust affine image registration method. *Int. J. Numer. Anal. Mod.*, 6(2):311–334, 2009.

[17] N. Chumchob and K. Chen. A robust multigrid approach for variational image registration models. *J. Comput. Appl. Math.*, 236(5):653–674, 2011.

[18] N. Chumchob and K. Chen. Improved variational image registration model and a fast algorithm for its numerical approximation. *Numer. Meth. Part. D. E.*, 28(6):1966–1995, 2012.

[19] N. Chumchob, K. Chen, and C. Brito-Loeza. A fourth-order variational image registration model and its fast multigrid algorithm. *Multiscale Model. Simul.*, 9(1):89–128, 2011.

[20] W. R. Crum, T. Hartkens, and D. L. Hill. Non-rigid image registration: theory and practice. *Brit. J. Radiol.*, pages S140–S153, 2004.

[21] M. Elsey and S. Esedoglu. Analogue of the total variation denoising model in the context of geometry processing. *Multiscale Model. Simul.*, 7(4):1549–1573, 2009.

[22] M. Erdt, S. Steger, and G. Sakas. Regmentation: A new view of image segmentation and registration. *Journal of Radiation Oncology Informatics*, 4(1):1–23, 2012.

[23] B. Fischer and J. Modersitzki. Fast inversion of matrices arising in image processing. *Numer. Algorithms*, 22(1):1–11, 1999.

[24] B. Fischer and J. Modersitzki. Curvature based registration with applications to mr-mammography. In *Computational Science ICCS 2002*, pages 202–206. Springer, 2002.

[25] B. Fischer and J. Modersitzki. Curvature based image registration. *J. Math. Imaging Vis.*, 18(1):81–85, 2003.

[26] B. Fischer and J. Modersitzki. A unified approach to fast image registration and a new curvature based registration technique. *Linear Algebra Appl.*, 380(0):107–124, 2004.

[27] D. S. Fritsch, E. L. Chaney, A. Boxwala, M. J. McAuliffe, S. Raghavan, A. Thall, and J. R. Earnhart. Core-based portal image registration for automatic radiotherapy treatment verification. *International Journal of Radiation Oncology Biology Physics*, 33(5):1287–1300, 1995.

[28] C. Frohn-Schauf, S. Henn, and K. Witsch. Multigrid based total variation image registration. *Comput. Vis. Sci.*, 11(2):101–113, 2008.

[29] J. C. Gee and R. K. Bajcsy. Elastic matching: Continuum mechanical and probabilistic analysis. In *Brain Wraping*, pages 193–198. Academic Press, 2000.

[30] E. Giusti. *Minimal Surfaces and Functions of Bounded Variation.* Birkhauser, 1984.

[31] A. Gooya, K. M. Pohl, M. Bilello, G. Biros, and C. Davatzikos. *Joint segmentation and deformable registration of brain scans guided by a tumor growth model*, pages 532–540. Springer, 2011.

[32] A. A. Goshtasby. *2-D and 3-D Image Registration: for Medical, Remote Sensing, and Industrial Applications.* Wiley-Interscience, 2005.

[33] E. Haber and J. Modersitzki. *Beyond Mutual Information: A Simple and Robust Alternative*, chapter 72, pages 350–354. Informatik aktuell. Springer Berlin Heidelberg, 2005.

[34] E. Haber and J. Modersitzki. *Intensity Gradient Based Registration and Fusion of Multi-modal Images*, pages 726–733. Springer Berlin Heidelberg, 2006.

[35] W. Hackbusch. *Multi-Grid Methods and Applications.* Springer-Verlag, 1985.

[36] J. V. Hajnal, D. J. Hawkes, and D. L. G. Hill. *Medical image registration.* Biomedical engineering series. CRC Press, 2001.

[37] M. P. Heinrich, M. Jenkinson, M. Bhushan, T. Matin, F. V. Gleeson, S. M. Brady, and J. A. Schnabel. Mind: modality independent neighbourhood descriptor for multi-modal deformable registration. *Med. Image Anal.*, 16(7):1423–35, 2012.

[38] S. Henn. A multigrid method for a fourth-order diffusion equation with application to image processing. *SIAM J. Sci. Comput.*, 27(3):831–849, 2005.

[39] S. Henn. A full curvature based algorithm for image registration. *J. Math. Imaging Vis.*, 24(2):195–208, 2006.

[40] S. Henn. A translation and rotation invariant gaussnewton like scheme for image registration. *BIT*, 46(2):325–344, 2006.

[41] S. Henn and K. Witsch. A variational image registration approach based on curvature scale space. In *Scale Space and PDE Methods in Computer Vision*, pages 143–154. Springer Berlin Heidelberg, 2005.

[42] D. Hill, D. Hawkes, J. Crossman, M. Gleeson, T. Cox, E. Bracey, A. Strong, and P. Graves. Registration of mr and ct images for skull base surgery using point-like anatomical features. *Brit. J. Radiol.*, 1991.

[43] D. L. Hill, P. G. Batchelor, M. Holden, and D. J. Hawkes. Medical image registration. *Phys. Med. Biol.*, 46(3):R1–45, 2001.

[44] E. Hodneland, A. Lundervold, J. Rrvik, and A. Z. Munthe-Kaas. Normalized gradient fields for nonlinear motion correction of dce-mri time series. *Comput. Med. Imag. Grap.*, 38(3):202–210, 2014.

[45] M. Holden. A review of geometric transformations for nonrigid body registration. *IEEE T. Med. Imaging*, 27(1):111–28, 2008.

[46] L. Hömke. A multigrid method for anisotropic pdes in elastic image registration. *Numerical Linear Algebra with Applications*, 13(2):215–229, 2006.

[47] L. Hömke, C. Frohn-Schauf, S. Henn, and K. Witsch. Total variation based image registration. In *Image Processing Based on Partial Differential Equations*, pages 343–361. Springer, 2007.

[48] B. K. P. Horn and B. G. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

[49] M. Ibrahim and K. Chen. A composition model combining parametric transformation and non-parametric deformation for effective image registration. In *SIAM Conference on Imaging Science*, 2014.

[50] A. Iserles. *A first course in the numerical analysis of differential equations*. Number 44. Cambridge University Press, 2009.

[51] C. T. Kelley. *Iterative methods for optimization*. SIAM, 1999.

[52] A. Klein, J. Andersson, B. A. Ardekani, J. Ashburner, B. B. Avants, M.-C. Chiang, G. E. Christensen, D. L. Collins, J. C. Gee, P. Hellier, J. H. Song, M. Jenkinson, C. Lepage, D. Rueckert, P. M. Thompson, T. Vercauteren, R. P. Woods, J. J. Mann, and R. V. Parsey. Evaluation of 14 nonlinear deformation algorithms applied to human brain mri registration. *Neuroimage*, 46(3):786–802, 2009.

[53] D. J. Kroon and C. H. Slump. Mri modalitiy transformation in demon registration. In *IEEE International Symposium on Biomedical Imaging: From Nano to Macro, ISBI '09.*, pages 963–966, 2009.

[54] J. Kybic. Fast parametric elastic image registration. *IEEE T. Image Process.*, 12(11), 2003.

[55] J. Kybic, P. Thevenaz, A. Nirkko, and M. Unser. Unwarping of unidirectionally distorted epi images. *IEEE T. Med. Imaging*, 19(2):80–93, 2000.

[56] K. C. Lam and L. M. Lui. Landmark- and intensity-based registration with large deformations via quasi-conformal maps. *SIAM J. Imaging Sci.*, 7:2364–2392, 2014.

[57] C. Le Guyader and L. Vese. *A Combined Segmentation and Registration Framework with a Nonlinear Elasticity Smoother*, pages 600–611. Springer Berlin Heidelberg, 2009.

[58] C. Le Guyader and L. Vese. A combined segmentation and registration framework with a nonlinear elasticity smoother. *Computer Vision and Image Understanding*, 115(12):1689–1709, 2011.

[59] S. Lee, G. Wolberg, and S. Y. Shin. Scattered data interpolation with multilevel b-splines. *IEEE T. Vis. Comput. Gr.*, 3:228–244, 1997.

[60] S.-H. Lee and J. K. Seo. Noise removal with gauss curvature-driven diffusion. *Trans. Img. Proc.*, 14(7):904–909, 2005.

[61] Q. Li, G. Wang, J. Liu, and S. Chen. Robust scale-invariant feature matching for remote sensing image registration. *Geoscience and Remote Sensing Letters, IEEE*, 6(2):287–291, 2009.

[62] T. Lin, C. Le Guyader, I. Dinov, P. Thompson, A. Toga, and L. Vese. Gene expression data to mouse atlas registration using a nonlinear elasticity smoother and landmark points constraints. *J. Sci. Comput.*, 50(3):586–609, 2012.

[63] H. Lombaert, L. Grady, X. Pennec, N. Ayache, and F. Cheriet. Spectral demons image registration via global spectral correspondence. In *Computer Vision ECCV 2012*, pages 30–34. Springer Berlin Heidelberg, 2012.

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

[65] J. A. Maintz and M. A. Viergever. A survey of medical image registration. *Med. Image Anal.*, 2(1):1–36, 1998.

[66] V. Mani and D. Rivazhagan. Survey of medical image registration. *Journal of Biomedical Engineering and Technology*, 1(2):8–25, 2013.

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

[68] J. Modersitzki. *Flexible Algorithms for Image Registration.* SIAM publications, 2009.

[69] M. Mora, J. Andrade, and C. Tauber. Regularized multilevel b-spline registration: Application to cardiac motion estimation. In *IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pages 512–515, 2010.

[70] D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on pure and applied mathematics*, 42(5):577–685, 1989.

[71] A. Myronenko and S. Xubo. Intensity-based image registration by minimizing residual complexity. *IEEE Transactions on Medical Imaging*, 29(11):1882–1891, 2010.

[72] J. Nocedal and S. J. Wright. *Numerical optimization.* Springer, 2006.

[73] S. Osher and J. A. Sethian. Fronts propagating with curvature-dependent speed: algorithms based on hamilton-jacobi formulations. *Journal of computational physics*, 79(1):12–49, 1988.

[74] X. Pennec, P. Cachier, and N. Ayache. Understanding the “demon’s algorithm”: 3D non-rigid registration by gradient descent. In *Proc. MICCAI’99*, pages 597–605. Springer-Verlag, 1999.

[75] J. P. Pluim, J. B. Maintz, and M. A. Viergever. Mutual information based registration of medical images: a survey. *IEEE T. Med. Imaging*, 22(8):986–1004, 2003.

[76] G. Porter. Cctv images as evidence. *Australian Journal of Forensic Sciences*, 41(1):11–25, 2009.

[77] A. Rangarajan, H. Chui, and J. S. Duncan. Rigid point feature registration using mutual information. *Medical Image Analysis*, 3(4):425–440, 1999.

[78] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1-4):259–268, 1992.

[79] W. Rudin. *Principles of mathematical analysis*, volume 3. McGraw-Hill New York, 1964.

[80] D. Rueckert, P. Aljabar, R. A. Heckemann, J. V. Hajnal, and A. Hammers. Diffeomorphic registration using b-splines. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2006*, pages 702–709. Springer, 2006.

[81] D. Rueckert and J. Schnabel. Medical image registration. In *Biomedical Image Processing*, pages 131–154. Springer, 2011.

[82] D. Rueckert, L. I. Sonoda, C. Hayes, D. L. Hill, M. O. Leach, and D. J. Hawkes. Nonrigid registration using free-form deformations: application to breast mr images. *IEEE T. Med. Imaging*, 18(8):712–21, 1999.

[83] J. Russell and R. Cohn. *Runge's Phenomenon.* Book on Demand, 2012.

[84] Y. Saad. *Iterative Methods for Sparse Linear Systems.* Society for Industrial and Applied Mathematics, 2003.

[85] H. Schumacher, A. Franz, and B. Fischer. Weighted medical image registration with automatic mask generation. In *Medical Imaging*, pages 61442B–61442B. International Society for Optics and Photonics, 2006.

[86] R. Shekhar and V. Zagrodsky. Mutual information-based rigid and nonrigid registration of ultrasound volumes. *IEEE Transactions on Medical Imaging*, 21(1):9–22, 2002.

[87] G. D. Smith. *Numerical solution of partial differential equations: finite difference methods.* Oxford university press, 1985.

[88] J. P. Thirion. Image matching as a diffusion process: an analogy with maxwell's demons. *Med. Image Anal.*, 2(3):243–260, 1998.

[89] L. N. Trefethen. *Approximation theory and approximation practice.* SIAM, 2013.

[90] U. Trottenberg and A. Schuller. *Multigrid.* Academic Press, 2001.

[91] G. Unal and G. Slabaugh. Coupled pdes for non-rigid registration and segmentation. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 168–175, 2005.

[92] B. Van Ginneken, B. M. ter Haar Romeny, and M. A. Viergever. Computer-aided diagnosis in chest radiography: a survey. *IEEE Transactions on Medical Imaging*, 20(12):1228–1241, 2001.

[93] T. Vercauteren, X. Pennec, E. Malis, A. Perchant, and N. Ayache. Insight into efficient image registration techniques and the demons algorithm. In *Information Processing in Medical Imaging*, pages 495–506. Springer, 2007.

[94] T. Vercauteren, X. Pennec, A. Perchant, and N. Ayache. Non-parametric diffeomorphic image registration with the demons algorithm. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2007*, pages 319–326. Springer, 2007.

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

[96] P. Viola and W. M. Wells III. Alignment by maximization of mutual information. *Int. J. Comput. Vision*, 24(2):137–154, 1997.

[97] C. R. Vogel. *Computational Methods for Inverse Problems.* Society for Industrial and Applied Mathematics, 2002.

[98] G. Wahba. *Spline Models for Observational Data.* Society for Industrial and Applied Mathematics, 1990.

[99] F. Wang and B. C. Vemuri. *Simultaneous registration and segmentation of anatomical structures from brain MRI*, pages 17–25. Springer, 2005.

[100] H. Wang, L. Dong, J. O'Daniel, R. Mohan, A. S. Garden, K. K. Ang, D. A. Kuban, M. Bonnen, J. Y. Chang, and R. Cheung. Validation of an accelerated 'demons' algorithm for deformable image registration in radiation therapy. *Phys. Med. Biol.*, 50(12):2887–905, 2005.

[101] J. Weickert, B. T. H. Romeny, and M. A. Viergever. Efficient and reliable schemes for nonlinear diffusion filtering. *IEEE Transactions on Image Processing*, 7(3):398–410, 1998.

[102] S. J. Wright and J. Nocedal. *Numerical optimization*, volume 2. Springer New York, 1999.

[103] Z. Xie and G. E. Farin. Image registration using hierarchical b-splines. *IEEE T. Vis. Comput. Gr.*, 10(1):85–94, 2004.

[104] F. Yamaguchi. *Curves and surfaces in computer aided geometric design.* Springer, 1988.

[105] I. Yanovsky, C. Le Guyader, A. Leow, W. Toga, Arthur, M. Thompson, Paul, and L. Vese. Unbiased Volumetric Registration via Nonlinear Elastic Regularization. In *2nd MICCAI Workshop on Mathematical Foundations of Computational Anatomy*, pages 1–12, 2008.

[106] I. Yanovsky, P. Thompson, S. Osher, and A. Leow. Topology preserving log-unbiased nonlinear image registration: Theory and implementation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, June 2007.

[107] I. Yanovsky, P. M. Thompson, S. Osher, and A. D. Leow. Large deformation unbiased diffeomorphic nonlinear image registration: Theory and implementation. *UCLA CAM Report 06-716*, 2006.

[108] A. Yezzi, L. Zollei, and T. Kapur. A variational framework for joint segmentation and registration. In *IEEE Workshop on Mathematical Methods in Biomedical Image Analysis*, pages 44–51, 2001.

[109] W. Zhu and T. Chan. Image denoising using mean curvature of image surface. *SIAM J. Imaging Sci.*, 5(1):1–32, 2012.

[110] W. Zhu, X.-C. Tai, and T. Chan. A fast algorithm for a mean curvature based image denoising model using augmented lagrangian method. In *Efficient Algorithms for Global Optimization Methods in Computer Vision*, pages 104–118. Springer Berlin Heidelberg, 2014.