

# Evaluation of Optimal Control-based Deformable Registration Model

By  
Naleli Jubert Matjelo



A dissertation submitted to the Department of Electrical Engineering,  
University of Cape Town, in fulfillment of the requirements  
for the degree of Master of Science in Engineering.

Supervisors: Prof. Martin Braae (UCT), A/Prof. Fred Nicolls (UCT)  
and Dr. Neil Muller (iThemba LABS)

Cape Town, May 19, 2014

# Declaration

I declare that this dissertation is my own, unaided work. It is being submitted for the degree of Master of Science in Engineering in the University of Cape Town. It has not been submitted before for any degree or examination in any other university.

Signature of Author.....

Cape Town

May 19, 2014

# Abstract

The deformable image registration is central to many challenges in medical imaging applications. The basic idea of the deformable image registration problem is to find an approximation of a reasonable deformation which transforms one image to match another based on a chosen similarity measure. A reasonable deformation can be thought of as one that is physically realizable. A number of models, guaranteeing reasonable deformations, have been proposed and implemented with success under various similarity measures. One such model is based on the grid deformation method (GDM) and is the method of interest in this thesis. This work focuses on the evaluation of an optimal control-based model for solving the deformable image registration problem which is formulated using GDM. This model is compared with other four well-known variational-based deformable image registration models: *elastic*, *fluid*, *diffusion* and *curvature* models. Using similarity and deformation quality measures as performance indices, the non-dominated sorting genetic algorithm (NSGA-II) is applied to approximate the Pareto fronts for each model to facilitate proper evaluation. The Pareto fronts are also visualized using level diagrams analysis.

# Acknowledgements

The success of this work has been the product of an unwearied support from the people who dared to sacrifice their times for the sake of this work presented in this document. Having said that I would like to proceed and thank these people who supported this work:

- Prof. M. Braae for the guidance, encouragement and technical support he gave as my Supervisor in every bit of this work.
- Dr. F. Nicolls for the technical support and guidance he offered as my UCT Co-Supervisor for the success of this work.
- My supervisor at iThemba LABS, Dr. N. Muller for his technical support and guidance.
- iThemba LABS for their financial assistance and provision of data to facilitate this work.
- My colleagues for the motivation they had installed in me throughout the course of this work.
- My family and friends for their love and support.

# Contents

<b>Declaration</b>	<b>i</b>
<b>Abstract</b>	<b>ii</b>
<b>Acknowledgements</b>	<b>iii</b>
<b>Nomenclature</b>	<b>xii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Problem Statement . . . . .	3
1.2 Purpose of Study . . . . .	3
1.3 Objectives of Study . . . . .	4
1.4 Scope and Limitations . . . . .	4
1.5 Thesis Outline . . . . .	5
<b>2 Affine Pre-Registration</b>	<b>6</b>
2.1 The Feature Space . . . . .	7
2.2 The Search Space . . . . .	7
2.2.1 Global (Parametric) Transformations . . . . .	8
2.2.1.1 Rigid Transformation . . . . .	8
2.2.1.2 Affine Transformation . . . . .	8

<i>CONTENTS</i>	v
2.2.1.3 Perspective Transformation . . . . .	9
2.2.2 Local (Nonparametric) Transformations . . . . .	10
2.3 The Similarity Measure . . . . .	10
2.3.1 Sum of Absolute Differences (SAD) . . . . .	11
2.3.2 Sum of Squared Differences (SSD) . . . . .	11
2.3.3 Correlation Coefficient (CC) . . . . .	11
2.3.4 Mutual Information (MI) . . . . .	12
2.4 The Search Strategy . . . . .	13
2.4.1 Principal Axes Transformation (PAT) . . . . .	15
2.4.2 Gradient-based Methods . . . . .	17
2.4.2.1 Gradient Descent (GD) . . . . .	18
2.4.2.2 Gauss Newton (GN) . . . . .	18
2.4.2.3 Levenberg-Marquardt (LM) . . . . .	19
2.4.3 Nelder-Mead Simplex (NMS) Method . . . . .	21
2.4.4 $\alpha$ - Branch and Bound ( $\alpha$ BB) . . . . .	24
2.4.5 Evolutionary Programming . . . . .	25
2.5 The Affine Pre-Registration System . . . . .	26
<b>3 Variational-based Models</b>	<b>29</b>
3.1 Elastic Model . . . . .	31
3.1.1 Formulation and Implementation . . . . .	32
3.2 Fluid Model . . . . .	34
3.2.1 Formulation and Implementation . . . . .	35
3.3 Diffusion Model . . . . .	37
3.3.1 Formulation and Implementation . . . . .	37

<i>CONTENTS</i>	vi
3.4 Curvature Model . . . . .	39
3.4.1 Formulation and Implementation . . . . .	39
<b>4 Optimal Control-based Model</b>	<b>41</b>
4.1 Grid Deformation Method . . . . .	41
4.1.1 The Deformation Method: Version One . . . . .	42
4.1.2 The Deformation Method: Version Two . . . . .	43
4.1.3 The Deformation Method: Version Three . . . . .	44
4.2 Optimal Control Formulation . . . . .	45
4.2.1 Problem Statement . . . . .	45
4.2.2 The Optimality System . . . . .	46
4.3 Solution to the Optimality System . . . . .	48
4.3.1 The System of Poisson Equations . . . . .	48
4.3.2 The System of Linear Equations . . . . .	51
<b>5 Evaluation in Fitness Space</b>	<b>56</b>
5.1 Motivation . . . . .	58
5.2 Pareto Front Approximation . . . . .	59
5.2.1 Pareto Front . . . . .	59
5.2.2 Approximation by NSGA-II . . . . .	59
5.3 Fitness (Objective) Functions . . . . .	62
5.3.1 Similarity Measure . . . . .	63
5.3.2 Deformation Quality Measure . . . . .	63
5.4 Level Diagrams Analysis of Pareto Fronts . . . . .	66
<b>6 Evaluation Results</b>	<b>68</b>

<i>CONTENTS</i>	vii
6.1 Experiment One . . . . .	69
6.1.1 Experimental Setup . . . . .	70
6.1.2 Results and Interpretations . . . . .	70
6.1.2.1 Comparison with Curvature Model . . . . .	71
6.1.2.2 Comparison with Elastic Model . . . . .	72
6.1.2.3 Comparison with Fluid Model . . . . .	73
6.1.2.4 Comparison with Diffusion Model . . . . .	73
6.1.2.5 Summary Table . . . . .	74
6.1.3 Visualization using Level Diagrams . . . . .	74
6.2 Experiment Two . . . . .	75
6.2.1 Experimental Setup . . . . .	75
6.2.2 Results and Interpretations . . . . .	76
6.2.3 Visualization Using Level Diagrams . . . . .	77
<b>7 Conclusion</b>	<b>79</b>
7.1 Future Work . . . . .	80

# List of Figures

2.1	Principal axes with the angle $\theta$ , and the major and minor axes corresponding to the largest and smallest eigenvalues of the covariance matrix. . . . .	16
2.2	The registration of two CT slices (top row) and two synthetic images (bottom row) using PAT. The first column shows the reference images $R(\mathbf{x})$ , the second column shows the template images $T(\mathbf{x})$ and the third column shows the transformed or registered template image $T(\mathbf{Ax})$ . . . . .	17
2.3	The registration of two CT slices (top row) and two synthetic images (bottom row) using LM. The first column shows the reference images $R(\mathbf{x})$ , the second column shows the template images $T(\mathbf{x})$ and the third column shows the transformed or registered template image $T(\mathbf{Ax})$ . . . . .	20
2.4	The registration of two CT slices (top row) and two synthetic images (bottom row) using the NMS method. The first column shows the reference images $R(\mathbf{x})$ , the second column shows the template images $T(\mathbf{x})$ and the third column shows the transformed or registered template image $T(\mathbf{Ax})$ . . . . .	23
2.5	The registration of two CT slices (top row) and two synthetic images (bottom row) using our pre-registration system. The first column shows the reference image $R(\mathbf{x})$ , the second column shows the template image $T(\mathbf{x})$ and the third column shows the transformed or registered template image $T(\mathbf{Ax})$ . . . . .	27
3.1	The registration of two CT slices (top row) and two synthetic images (bottom row) using the elastic model. The first column shows the reference images $R(\mathbf{x})$ , the second column shows the template images $T(\mathbf{x})$ , the third column shows transformed or registered template image $T(\phi(\mathbf{x}))$ and the last column shows the resulting deformation $\phi(\mathbf{x})$ . . . . .	34

3.2 The registration of two CT slices (top row) and two synthetic images (bottom row) using the fluid model. The first column shows the reference images  $R(\mathbf{x})$ , the second column shows the template images  $T(\mathbf{x})$ , the third column shows transformed or registered template image  $T(\phi(\mathbf{x}))$  and the last column shows the resulting deformation  $\phi(\mathbf{x})$ . . . . . 36

3.3 The registration of two CT slices (top row) and two synthetic images (bottom row) using the diffusion model. The first column shows the reference images  $R(\mathbf{x})$ , the second column shows the template images  $T(\mathbf{x})$ , the third column shows transformed or registered template image  $T(\phi(\mathbf{x}))$  and the last column shows the resulting deformation  $\phi(\mathbf{x})$ . . . . . 38

3.4 The registration of two CT slices (top row) and two synthetic images (bottom row) using the curvature model. The first column shows the reference images  $R(\mathbf{x})$ , the second column shows the template images  $T(\mathbf{x})$ , the third column shows transformed or registered template image  $T(\phi(\mathbf{x}))$  and the last column shows the resulting deformation  $\phi(\mathbf{x})$ . . . . . 40

4.1 Mapping an arbitrary quadrilateral element in the physical space onto the regular quadrilateral in the natural space to simplify integration. . . . . 50

4.2 Registration of two synthetic image using the optimal control-based model implemented with GFEM. The first column shows the reference images  $R(\mathbf{x})$ , the second column shows the template images  $T(\mathbf{x})$ , the third column shows transformed or registered template image  $T(\phi(\mathbf{x}))$  and the last column shows the resulting deformation  $\phi(\mathbf{x})$ . . . . . 51

4.3 Registration of two synthetic image using the Optimal Control-based model implemented with LSFEM. The first column shows the reference images  $R(\mathbf{x})$ , the second column shows the template images  $T(\mathbf{x})$ , the third column shows transformed or registered template image  $T(\phi(\mathbf{x}))$  and the last column shows the resulting deformation  $\phi(\mathbf{x})$ . . . . . 55

5.1 The flow diagram of NSGA-II. . . . . 60

5.2 Approximation of Pareto Front generated by NSGA-II. . . . . 62

5.3 A regular grid with quadrilateral elements (left) and its deformed version (right). 64

5.4 The Level Diagram of the Pareto Front example in Fig. 5.2. . . . . 67

6.1	Example of a CT slice (left) and a DRR (right). . . . .	69
6.2	Two dimensional side views of Pareto fronts approximation. The top row with sub-figures labeled (a), (b) and (c) shows the Pareto fronts generated without including the pre-registration step. The bottom row with sub-figures (d), (e) and (f) shows the Pareto fronts generated with the pre-registration step included. . . . .	71
6.3	The Level Diagrams for the Pareto Fronts in Fig. 6.2 above. . . . .	75
6.4	Two dimensional side views of Pareto fronts approximation. . . . .	76
6.5	The level diagram analysis for the Pareto fronts in Fig. 6.4 above. . . . .	77
6.6	The registration of DRRs by curvature model. Top row show two DRRs produced by cross sectional projection. Bottom row shows two DRRs produced by side view projection. . . . .	78

# List of Tables

- 2.1 Classification of Optimization methods based on search strategy and scope. 14
  
- 6.1 Comparison of variational-based models with the optimal control-based model based on sensitivity to pre-registration, similarity and deformation quality measures. . . . . 74
  
- 6.2 The NSGA-II minimization output relating to the curvature model. . . . . 78

# Nomenclature

## Abbreviations

<b>aDVF</b>	—	Automatically calculated Displacement Vector Field.
$\alpha$ <b>BB</b>	—	Branch and Bound method with convexification parameter $\alpha$ .
$\gamma$ <b>BB</b>	—	Branch and Bound method with convexification parameter $\gamma$ .
<b>BB</b>	—	Branch and Bound.
<b>CT</b>	—	Computer Tomography.
<b>CG</b>	—	Conjugate Gradient method.
<b>CC</b>	—	Cross Correlation.
<b>DRRs</b>	—	Digitally Reconstructed Radiographs.
<b>EAs</b>	—	Evolutionary Algorithms.
<b>EP</b>	—	Evolutionary Programming.
<b>FDM</b>	—	Finite Difference Method.
<b>FEM</b>	—	Finite Element Method.
<b>FFD</b>	—	Free-Form Deformation.
<b>GFEM</b>	—	Galerkin Finite Element Method.
<b>GN</b>	—	Gauss Newton method.
<b>GAs</b>	—	Genetic Algorithms.
<b>GD</b>	—	Gradient Descent method.
<b>LSFEM</b>	—	Least Squares Finite Element Method.
<b>LM</b>	—	Levenberg-Marquardt method.
<b>MRI</b>	—	Magnetic Resonance Imaging.
<b>mDVF</b>	—	Measured Displacement Vector Field.
<b>MAs</b>	—	Memetic Algorithms.
<b>MI</b>	—	Mutual Information.
<b>NMS</b>	—	Nelder-Mead Simplex method.
<b>NSGA</b>	—	Non-dominated Sorting Genetic Algorithm.

<b>OF</b>	—	Optical Flow.
<b>PDE</b>	—	Partial Differential Equation.
<b>PRs</b>	—	Portal Radiographs.
<b>P</b>	—	Powell method.
<b>PAT</b>	—	Principal Axes Transformation.
<b>QN</b>	—	Quasi-Newton method.
<b>SBX</b>	—	Simulated Binary Crossover.
<b>SOR</b>	—	Successive Over-Relaxation.
<b>SAD</b>	—	Sum of Absolute Differences.
<b>SSD</b>	—	Sum of Squared Differences.

### List of Symbols

$T(\mathbf{x})$	—	Template image.
$R(\mathbf{x})$	—	Reference image.
$\phi(\mathbf{x})$	—	Deformation.
$T(\phi(\mathbf{x}))$	—	Transformed template image.
$\Omega$	—	Image domain.
$\Gamma$	—	Boundary of image domain.
$\lambda, \mu$	—	<i>Lamé-constants</i> for elastic materials.
$L^2$	—	Lebesgue norm.
$H^1$	—	Sobolev norm.
$\mathbf{u}(\mathbf{x})$	—	Displacement field.
$f(\mathbf{x})$	—	Monitor function (controls area).
$g(\mathbf{x})$	—	Control function for local rotations.

# Chapter 1

## Introduction

Image registration is defined as a process of determining an optimal transformation that maps the template image points to the corresponding points in the reference image [1]. These images could be of the same or different objects and imaging modalities, and possibly captured at different times and pose. Generally speaking image registration can be divided into two major categories depending on the types of transformations concerned. The first category is the rigid (global) image registration whose search space is limited to global transformations like affine and perspective transformations. Such transformations can only map straight lines to other straight lines. The second category is the nonrigid (deformable) image registration whose search space extends to include nonlinear transformations which can map straight lines onto curves.

With recent developments in imaging technology and increases in computing power, image registration has found its use in many areas of inquiry including in the fields of astronomy, health sciences and engineering. Medical imaging is one such field which makes extensive use of image registration in solving daily health problems. Image registration is an essential component in medical imaging due to the fact that it serves as one of the prerequisites for advanced processing and analysis including disease diagnosis, monitoring and treatment. Using image registration in these applications requires prior knowledge of the nature of tissue or object involved, such as its softness. This gives an idea as to which methods of image registration will be most appropriate based on the transformations they permit.

Rigid image registration is appropriate in applications where the involved tissue or object deformation can be reasonably approximated by global transformations and such applica-

tions include brain tumor diagnosis. However global image registration in itself can not be used satisfactorily in applications involving large nonlinear deformation such as in breast and prostate cancer diagnosis. For such applications deformable image registration is the most suited for the job.

Deformable image registration aims to find a physically realizable deformation that optimizes a chosen similarity measure between two images. That is, the deformable image registration problem can be thought of as a bi-objective minimization problem whereby two key objectives are *similarity measure* and *deformation quality measure*.

Similarity measure provides a quantitative measure of correspondence between two image features and it is upon this quantifiable correspondence measure that optimization problems can be formulated such that alignment of features is maximized. *sum of squared differences* (SSD) is one of the popular similarity measures in the literature [2, 3, 4] and it is adopted for evaluation in this work.

The deformation quality measure is perhaps the most vital component in the deformable image registration process since without it arbitrary deformations may lead to grid tangling and other undesirable solutions. The necessity of good quality deformations has led to development of various kinds of deformable models including variational-based ones like *elastic*, *fluid*, *diffusion* and *curvature* models [3, 4, 5, 6, 7]. Each of these variational-based models is equipped with a unique regularizer which penalizes physically unrealizable deformations.

These regularization terms result in transformations which do not optimize the similarity measure alone; in fact if the weighting on the regularizer is too small, the model will be unstable. On the other hand if it is too large the regularity will be too strong and the resulting transformation will not accurately optimize the similarity measure [1]. Closely related to variational-based models is an optimal control approach to the deformable image registration problem, whose formulation is based on grid deformation method (GDM) [1, 8, 9]. GDM is able to generate a grid with a desired grid density distribution that is free from grid tangling without the need for any regularizer. This is achieved through a positive monitor function as a control input. More details on the variational-based model and the optimal control-based model will be discussed in Chapter 3 and Chapter 4 respectively.

## 1.1 Problem Statement

The current proton therapy program at iThemba LABS is designed around treating lesions predominantly in the head and neck region. One of the useful side effects is that the anatomy in this region is very rigid and registration of the target volume can rely on comparatively simple image registration techniques. It would be desirable, however, to be able to treat a wider range of configurations, especially treating various lesions in the abdominal region such as the prostate. Since the body structure is less rigid, registering the patient position to the CT scan data is much harder. This requires investigating various deformable image registration models to decide when two shapes are sufficiently similar.

Although there are many models in the literature for tackling the deformable image registration problem relatively less work has been done on the subject of their evaluation. One model may perform best for a particular application and worse in other applications. The performances of these models usually differ due to the differences in their formulation and without sufficient evaluation it is not easy to decide which model is best suited for a particular application. There is a need to carry out more evaluations and most importantly quantitative evaluations which give more useful insight on the models through defining proper objectives as performance markers to facilitate efficient evaluation.

Evaluation of the adequacy of deformable image registration models for a given application is required not only to allow the appropriate selection of the most suitable model but also to tune their parameters for optimal performance. Based on the chosen objectives the models are evaluated, and from the evaluation results the strengths and weakness of each model are investigated. For instance some models penalize global transformations hence are likely to perform poorly in the absence of good initialization relative to others which do not penalize global transformations. So the sensitivity of the models to initialization needs to be investigated through the evaluation, thus arming the designer with enough information to make well-informed decisions.

## 1.2 Purpose of Study

With regard to the problem outlined in the problem statement section above, the purpose of this study is to carry out an evaluation of the optimal control-based model, using as references for comparison purposes four variational-based models. In so doing, these other four variational-based models are evaluated as well. This work serves as a contribution, in

some way, to an ongoing study on the evaluation of deformable image registration models.

### 1.3 Objectives of Study

The objectives of this study include the development of a fast and reliable pre-registration system that offers better initialization for the deformable registration models. This will also help investigate the sensitivity of each model to initialization provided by the pre-registration system. The implementation of all deformable models is on the same framework of Finite Elements as a way of avoiding differences in approximations brought by different frameworks. The decision is made as to which evaluation scheme to use to allow for efficient and rigorous comparison between the optimal control-based model and the variational-based models. In order to facilitate proper evaluation reasonable objective functions are proposed to serve as performance indices. A method for visualizing the results of evaluation is chosen and used to show how this evaluation can guide a designer or user when choosing a model appropriate for a particular application.

### 1.4 Scope and Limitations

In this study the evaluation does not cover all models used for solving the deformable image registration problem. Rather, it covers only four variational models: elastic, fluid, diffusion and curvature models as well as the optimal control-based model. There are different optimal control-based models for solving the deformable image registration problem, but we focus only on the model formulated based on the grid deformation method (GDM). There are three versions of GDM, two of which are static while the third is dynamic. The version of GDM used in the formulation of the optimal control-based model is the first of the static versions.

In the evaluation procedure we have not used convergence time as one of the performance measures so the results of this study are invariant to the differences in the computational intensities of model implementations. SSD is the only similarity measure used for carrying out the entire evaluation process. This is because there are no experiments conducted which involve multimodality so SSD is justified as a reasonable similarity measure. All the experiments for this evaluation use computer tomography (CT) slices and digitally reconstructed radiographs (DRRs) only.

## 1.5 Thesis Outline

The rest of this document proceeds in the following fashion: Chapter 2 addresses the basics of image registration with emphasis on rigid image registration and some of its solution methods. Some of these methods are integrated to form a reliable pre-registration system to be used later in evaluation. Chapter 3 covers the variational-based models and their implementation procedure while Chapter 4 presents the optimal control-based model and its implementation procedure. Chapter 5 covers the evaluation method used, and proposes the objective functions for evaluation as well as the method to be used for interpreting the evaluation results. Chapter 6 describes the experiments over which the models are tested and presents the evaluation results as well as their interpretations. Chapter 7 concludes the study and makes comments on possible future work.

# Chapter 2

## Affine Pre-Registration

Iterative solution methods for the deformable image registration problem suffer from convergence problems whenever good initial guesses are not available [2]. These convergence problems arise mainly due to the fact that many deformable image registration models penalize the affine (global) transformations in their formulation, as is demonstrated in the later chapters. This problem can be overcome by providing good initial guesses so that those mismatches described by affine transformations are mostly reduced between the two images. One way of providing good initialization is by using a reliable affine (global) image registration method in a pre-registration step, as mentioned in [2]. For that reason we need to assemble a reliable pre-registration system that will be used in deformable image registration. This requires exploration of available options for solving the affine image registration problem. This chapter therefore starts by covering the basic concepts of image registration in general. More focus is later turned towards the affine image registration which is essential in pre-registration.

Given two images, a template image  $T(\mathbf{x})$  and a reference image  $R(\mathbf{x})$  with  $\mathbf{x} \in \Omega \subset \mathbb{R}^m$ , image registration aims to obtain a physically realizable transformation  $\phi(\mathbf{x})$  from  $T(\mathbf{x})$  to  $T(\phi(\mathbf{x}))$  such that the similarity between  $R(\mathbf{x})$  and  $T(\phi(\mathbf{x}))$  is minimized [2]. The superscript  $m$  indicates the dimensionality of the image space  $\Omega$ , thus  $m = 2$  for two dimensional (2D) images. As mentioned in [10], image registration is in general composed of four distinct components which are: *the feature space, the search space, the similarity measure* and *the search strategy*. These components are addressed in the sections below.

## 2.1 The Feature Space

The feature space is concerned with that which is observable in the image. Thus it is constituted by observables such as raw pixel intensities, statistical properties, contours, corners and landmarks in the image. Features like edges and corners are secondary to pixel intensities since they are extracted out of pixel intensity distributions in the image. The similarity measure necessitates the feature space in the sense that the feature space defines the observables within the images that can be used for comparisons purposes. Image registration methods can be classified based on the nature of the feature space used in the construction of the similarity measure. Methods which use raw pixel intensities are referred to as *intensity-based* methods while those which use secondary features like contours and landmarks are referred to as *feature-based* methods. Unlike the intensity-based methods, which are computationally intensive and slow because of the necessity to process all pixels, the feature-based methods tend to be a lot faster since processing is not done over the entire image domain  $\Omega$  or every pixel. However the feature-based methods can result into undesirable results due to not processing all image pixels; one example of this can be found in [11] with the landmark-based image registration of the human hand. This problem can be solved by increasing the number of landmarks with a uniform distribution.

## 2.2 The Search Space

The search space is more focused on the kinds of transformation models considered in image registration process. In general these transformation models can be classified as either global (parametric) or local (nonparametric) transformation models based on how they apply across the whole image domain  $\Omega$ . Regardless of whether global or local, the transformation  $\phi(\mathbf{x})$  is intended to align the two images with respect to their common features. The nature of distortions in images has influence on the choice of the search space, although the nature of the distortions is sometimes not known *a priori*. Both global and local transformations are elaborated in the subsections below.

## 2.2.1 Global (Parametric) Transformations

A global transformation is one which applies uniformly to every feature in the image domain, and such transformations include *translation, rotation, scaling, shear, projection* and their combinations such as *rigid, affine* and *perspective* transformation models. Global transformations are limited to mapping straight lines to straight lines but not curves. Due to their uniformity the global transformations can be factored and parametrized into matrix form with few parameters. This makes the rigid image registration problem relatively easier to solve than the nonrigid registration problem.

### 2.2.1.1 Rigid Transformation

The rigid transformation is constituted by translation and rotation transformations only, so object shape and angles are preserved. In 2D the rigid transformation can be described by the following equations which transform a coordinate pair  $\mathbf{x} = (x, y)$  to a new coordinate pair  $\phi(\mathbf{x}) = (x', y')$ :

$$\begin{aligned} x' &= x \cos \theta - y \sin \theta + a_3 \\ y' &= x \sin \theta + y \cos \theta + a_6 \end{aligned} \quad (2.1)$$

where  $\theta$  is the rotation angle while  $a_3$  and  $a_6$  are the translation parameters. These equations can be put in matrix form as follows:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & a_3 \\ \sin \theta & \cos \theta & a_6 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}. \quad (2.2)$$

Letting  $\tilde{\mathbf{x}} = \begin{bmatrix} x & y & 1 \end{bmatrix}^T$  and  $\mathbf{A}$  be the transformation matrix above, we can rewrite Equation (2.1) above even more compactly as  $\phi(\tilde{\mathbf{x}}) = \mathbf{A}\tilde{\mathbf{x}}$ . Combining the rigid transformation above with an isotropic scale  $\alpha$  gives rise to a *similarity* transformation. A more general transformation than the similarity transformation is an *affine* transformation which is elaborated below.

### 2.2.1.2 Affine Transformation

The affine transformation is restricted to mapping straight lines onto other straight lines and parallelism of lines is conserved. In contrast to rigid transformations, an affine transformation does not preserve angles. It is composed of the combination of rotation  $\mathbf{R}_t$ ,

anisotropic scale  $\mathbf{S}_c$ , shear  $\mathbf{S}_h$  and translation  $\mathbf{T}_r$  in any order. These transformations are shown below:

$$\mathbf{R}_t = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{T}_r = \begin{bmatrix} 1 & 0 & a_3 \\ 0 & 1 & a_6 \\ 0 & 0 & 1 \end{bmatrix}, \quad (2.3)$$

$$\mathbf{S}_c = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{S}_h = \begin{bmatrix} 1 & s_h & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Parametrically the affine transformation matrix  $\mathbf{A}$  can be represented in terms of the fundamental transformation matrices in Equation (2.3) as shown below:

$$\mathbf{A} = \mathbf{T}_r \mathbf{R}_t \mathbf{S}_c \mathbf{S}_h = \begin{bmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ 0 & 0 & 1 \end{bmatrix}, \quad (2.4)$$

where  $a_1, \dots, a_6$  result from computing the matrix product of these fundamental matrices. Now the affine transformation can be represented as  $\phi(\tilde{\mathbf{x}}) = \mathbf{A}\tilde{\mathbf{x}}$ . The problem of affine image registration can then be thought of as seeking the parameters  $a_i$ ,  $i = 1, \dots, 6$ , which minimize a chosen dissimilarity measure between two images. The only constraint that is enforced on the parameters  $a_i$  is that they should not lead to a singular transformation matrix  $\mathbf{A}$ .

### 2.2.1.3 Perspective Transformation

Unlike in the affine transformation model parallelism of lines is not preserved in the perspective transformation model. The perspective transformation model adds two projection parameters  $a_7$  and  $a_8$  to the original affine transformation model and gives the following more general transformation matrix  $\mathbf{A}$ :

$$\mathbf{A} = \begin{bmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ a_7 & a_8 & 1 \end{bmatrix}. \quad (2.5)$$

### 2.2.2 Local (Nonparametric) Transformations

A local transformation is one which does not necessarily apply uniformly throughout all image domain. This means local transformations include nonlinear transformations which cannot be parametrized into matrix form but rather have at least as many parameters as there are features in the image. This is because every feature in the image has its own transformation, which can be different from every other transformation applied to other features. Contrary to global transformations, local transformations can map straight lines onto curves so are nonlinear. These local transformations form the search space considered in nonrigid or deformable image registration. Unlike global transformations, which always keep the transformed objects intact, local transformations allow deformations which can result in objects splitting. These kind of deformations are not favorable in deformable image registration and are part of what renders the problem ill-posed [12, 13]. For this reason constraints or regularizers are necessary to prevent certain unwanted local transformations from occurring. These regularizers are explained in Chapter 3.

The local transformation  $\phi(\mathbf{x})$  can be represented as follows:

$$\phi(\mathbf{x}) = \mathbf{x} \pm \mathbf{u}(\mathbf{x}), \quad (2.6)$$

where  $\mathbf{u}(\mathbf{x})$  is the displacement field which quantifies the shift of every point  $\mathbf{x}$  in the image domain  $\Omega$ . In general  $\mathbf{u}(\mathbf{x})$  can be any function, but for deformable image registration it is desired that  $\mathbf{u}(\mathbf{x})$  be a bijection so that it is both continuous and invertible. The inversion is important in deformable image registration because it allows the template image to be matched to the reference image and for the reference image to be matched to the template image.

## 2.3 The Similarity Measure

Having decided the features of interest, the image registration process proceeds by aligning features that are common to both images. The modalities involved in image registration influence the choice of features for image registration. Some common imaging modalities include X-ray, computer tomography (CT), magnetic resonance imaging (MRI) and ultrasound. For instance, statistical properties of images may be more appropriate for multimodal applications than the raw pixel intensities. The choice of similarity measure is reliant upon the modalities of images involved in the registration process. The similarity

measure is focused on quantifying the differences between the geometrically transformed template image and the reference image. It is the measure of how well  $T(\phi(\mathbf{x}))$  matches with  $R(\mathbf{x})$ .

The most intuitive and most popular similarity measure in the literature is the sum of squared differences (SSD). It is cheap to compute but is only appropriate to monomodal applications. Other similarity measures include sum of absolute differences (SAD), correlation coefficient (CC) and mutual information (MI). These similarity measures are described briefly in the sections below.

### 2.3.1 Sum of Absolute Differences (SAD)

The SAD measures how similar two images are to each other based on the sum of absolute values of their pixel difference. When images match exactly the SAD will be exactly zero. The SAD is given as:

$$\text{SAD} = \int_{\Omega} |T(\phi(\mathbf{x})) - R(\mathbf{x})| d\mathbf{x}. \quad (2.7)$$

This similarity measure presents some difficulties when first order necessary conditions for optimality are derived since computing its derivative is not easy.

### 2.3.2 Sum of Squared Differences (SSD)

The SSD is one of the most popular similarity measures in the literature and is very efficient in monomodal applications. It is based on the  $L^2$ -norm of the discrepancies between the transformed template image  $T(\phi(\mathbf{x}))$  and the reference image  $R(\mathbf{x})$  as shown below:

$$\text{SSD} = \frac{1}{2} \|T(\phi(\mathbf{x})) - R(\mathbf{x})\|_{L^2}^2 = \frac{1}{2} \int_{\Omega} (T(\phi(\mathbf{x})) - R(\mathbf{x}))^2 d\mathbf{x}. \quad (2.8)$$

Despite it being of higher order relative to SAD this similarity measure is easily differentiable since it has no discontinuities in its formulation.

### 2.3.3 Correlation Coefficient (CC)

As the name suggests, the CC quantifies the similarity between the template image and the reference image based on how well they correlate with each other. It measures the statistical dependence or information redundancy between the image intensities of corresponding

pixels in both images [14]. This similarity measure is computed as follows:

$$\text{CC} = \frac{\int_{\Omega} (T(\phi(\mathbf{x})) - \bar{T})(R(\mathbf{x}) - \bar{R})d\mathbf{x}}{\sqrt{\int_{\Omega} (T(\phi(\mathbf{x})) - \bar{T})^2 d\mathbf{x} \int_{\Omega} (R(\mathbf{x}) - \bar{R})^2 d\mathbf{x}}}, \quad (2.9)$$

with  $\bar{T}$  and  $\bar{R}$  being the mean intensity values of the template image and the reference image respectively. When the discrepancies between the two images is minimized the CC will be maximized. This similarity measure is very appropriate in multimodal applications. No limiting constraints are imposed on the image content of the modalities involved [14].

### 2.3.4 Mutual Information (MI)

Mutual information is an information theory measure of the statistical dependence between two random variables or the amount of information that one variable contains about the other. It can be qualitatively considered as a measure of how well one image explains the other [15, 16]. Similar to CC, MI gets maximized when the two images match correctly. The formulation of MI between the transformed template image  $T(\phi(\mathbf{x}))$  and the reference image  $R(\mathbf{x})$  is constituted by the entropy of the template image  $H(T(\phi(\mathbf{x})))$ , the entropy of the reference image  $H(R(\mathbf{x}))$ , and the joint entropy  $H(T(\phi(\mathbf{x})), R(\mathbf{x}))$ , defined as follows:

$$\begin{aligned} H(T(\phi(\mathbf{x}))) &= -\sum_t P_T(t) \log_2(P_T(t)), \\ H(R(\mathbf{x})) &= -\sum_r P_R(r) \log_2(P_R(r)), \\ H(T(\phi(\mathbf{x})), R(\mathbf{x})) &= -\sum_t \sum_r P_{TR}(t, r) \log_2(P_{TR}(t, r)), \end{aligned} \quad (2.10)$$

where  $P_T(t)$ ,  $P_R(r)$  are the marginal distributions of the image intensities of the template image and the reference image, and  $P_{TR}(t, r)$  is their joint probability. Given Equation (2.10), MI is defined as follows:

$$\text{MI}(T(\phi(\mathbf{x})), R(\mathbf{x})) = H(T(\phi(\mathbf{x}))) + H(R(\mathbf{x})) - H(T(\phi(\mathbf{x})), R(\mathbf{x})). \quad (2.11)$$

There are other similarity measures, such as those constructed out of image moments, but these are not discussed here.

## 2.4 The Search Strategy

The search strategy is concerned with the method used for obtaining the transformation  $\phi(\mathbf{x})$  which accurately matches the template image  $T(\phi(\mathbf{x}))$  to the reference image  $R(\mathbf{x})$ . The most popular strategy for finding this transformation is optimization [2, 17]. In optimization a cost functional is formulated based mainly on the similarity measure, and this functional is then minimized with respect to the transformation parameters so that the optimal transformation is obtained. As mentioned in [2] the general form of the cost functional is given by:

$$J[\phi] = D[R, T, \phi] + \alpha S[\phi], \quad \alpha > 0, \quad (2.12)$$

where  $D[R, T, \phi]$  is the similarity measure with  $\phi$  as the transformation,  $\alpha$  is the regularization constant and  $S[\phi]$  is the regularizer or smoothness measure. In global image registration the regularization  $S[\phi]$  can be neglected since the only regularization that is required is for the transformation matrix  $\mathbf{A}$  to be non-singular, and this can be achieved without adding the regularization term. Also since the transformation parameters can be factored into the matrix, the cost functional can be restated such that it varies explicitly with the transformation matrix elements  $a_i$ . The resulting cost functional for global image registration problem is then given as:

$$J[\mathbf{a}] = D[T, R, \mathbf{a}], \quad (2.13)$$

where the vector containing the elements of  $\mathbf{A}$  is denoted by  $\mathbf{a}$ , that is  $\mathbf{a} = [a_1 \dots a_n]^T$  with  $n = 6$  and  $n = 8$  for affine and perspective transformation models respectively. The optimization problem can then be presented as follows:

$$\min_{\mathbf{a}} J[\mathbf{a}]. \quad (2.14)$$

Since we are considering minimization problems CC and MI need to be inverted in order to conform to the minimization methods. There are many optimization methods that are used to solve this problem and such methods are divided into *local optimization methods* or *global optimization methods* based on whether they are designed to find local optimal solutions or global optimal solutions. Local optimization methods are often faster than global optimization methods but they are prone to getting trapped in local minima. Also based on how they search for the optimal solution, these methods can also be classified into *deterministic search methods* [18] and *stochastic search methods* [19].

Deterministic search methods such as the *gradient-based* methods and *direction-based* methods use the following iterative search procedure:

$$\mathbf{a}^{k+1} = \mathbf{a}^k + \alpha^k \mathbf{d}^k, \quad (2.15)$$

where  $\mathbf{a}^k$ ,  $\mathbf{d}^k$  and  $\alpha^k$  denote the current solution vector, the search direction vector and the scaling factor obtained at each iteration  $k$  respectively. Given the current solution  $\mathbf{a}^k$  and the search direction  $\mathbf{d}^k$  at iteration  $k$ , the scaling factor  $\alpha^k$  is chosen such that  $J[\mathbf{a}^k + \alpha^k \mathbf{d}^k] \leq J[\mathbf{a}^k]$ . There are many deterministic methods in the literature [17, 20] and each of them is identified by how it obtains the search direction  $\mathbf{d}^k$  and the scaling factor  $\alpha^k$ . The two major groups of deterministic optimization methods are *gradient-based* methods and *direction-based* methods. The gradient-based methods include *gradient descent* (GD) method, *Newton method*, *Gauss Newton* (GN) method, *quasi-Newton* (QN) method, *Levenberg-Marquardt* (LM) method and *conjugate gradient* (CG) method [17, 20]. The direction-based methods include *Nelder-Mead simplex* (NMS) method, *Powell* (P) method and *differential evolution* (DE). In this chapter we cover NMS, GD, GN and LM methods.

Global optimization methods include *branch and bound* (BB) methods [21], *evolutionary algorithms* (EAs), *genetic algorithms* (GAs) and *memetic algorithms* (MAs) [19, 22, 23]. BB methods are deterministic, and they guarantee convergence to a global optimal solution in a finite number of iterations. EAs and GAs are stochastic search methods while MAs adopt both the deterministic and the stochastic search strategies. In this chapter we cover  $\alpha$ -branch and bound ( $\alpha$ BB) and evolutionary programming (EP), although they are not included in the development of the pre-registration system due to being quite slow for real-time applications. Table 2.1 shows a summary of these optimizations methods classified based on their type as well as their search strategy.

Table 2.1: Classification of Optimization methods based on search strategy and scope.

Optimization Method	Global/Local	Deterministic/Stochastic
GD, GN, QN, LM & CG	Local	Deterministic
NMS, PM, DE	Local	Deterministic
BB methods (e.g. $\alpha$ BB, $\gamma$ BB)	Global	Deterministic
EAs and GA	Global	Stochastic
MA	Global	Deterministic & Stochastic

There are other methods for obtaining a required transformation to match two images and some are the *principal axis method* (PAT), *Fourier-based correlation*, the *Procrustes algorithm* and *mutual information-based techniques* [10]. In this chapter we cover only

PAT and use it in the implementation of the pre-registration system. Details of other methods can be found in [10]. The next section elaborates more on PAT and the sections that follow focus on the optimization approach.

### 2.4.1 Principal Axes Transformation (PAT)

PAT is an approximate registration approach using statistical features, the image centers and an eigendecomposition of the covariance matrix derived from the input images [2]. As mentioned in [10] the feature space acted upon by PAT consists of features of images, such as edges and corners. The search space consists of global translations and rotations while the similarity measure is the variance of the projection of the feature's location vector onto the principal axis [10].

Given the template image and the reference image a centroid can be computed for each image and the difference between centroids can be used to translate the template image to match the reference image. The center coordinates  $(\bar{x}_T, \bar{y}_T)$  for the template image  $T(\mathbf{x})$  and the center coordinates  $(\bar{x}_R, \bar{y}_R)$  for reference image  $R(\mathbf{x})$  are given by the following:

$$\begin{aligned}\bar{x}_T &= \frac{\sum_{x,y} xT(x,y)}{\sum_{x,y} T(x,y)}, & \bar{y}_T &= \frac{\sum_{x,y} yT(x,y)}{\sum_{x,y} T(x,y)}, \\ \bar{x}_R &= \frac{\sum_{x,y} xR(x,y)}{\sum_{x,y} R(x,y)}, & \bar{y}_R &= \frac{\sum_{x,y} yR(x,y)}{\sum_{x,y} R(x,y)}.\end{aligned}\tag{2.16}$$

Now the centers of the images are aligned by translating the template image using the following translation equations:

$$a_3 = \bar{x}_R - \bar{x}_T, \quad a_6 = \bar{y}_R - \bar{y}_T .\tag{2.17}$$

The centroids are then used to form covariance matrices for the template image  $C_T$  and the reference image  $C_R$ . The eigenvalue decompositions of the matrices give rotation information necessary to transform the template image to align with the reference image. In the 2D case, the distribution of image data forms an elliptic shape and the eigenvector matrix from the eigenvalue decomposition of  $C_T$  or  $C_R$  is a rotation matrix whose angle of rotation  $\theta$  is the angle made by the major axis of an ellipse with the horizontal axis of the coordinate space. This is shown in Fig. 2.1.

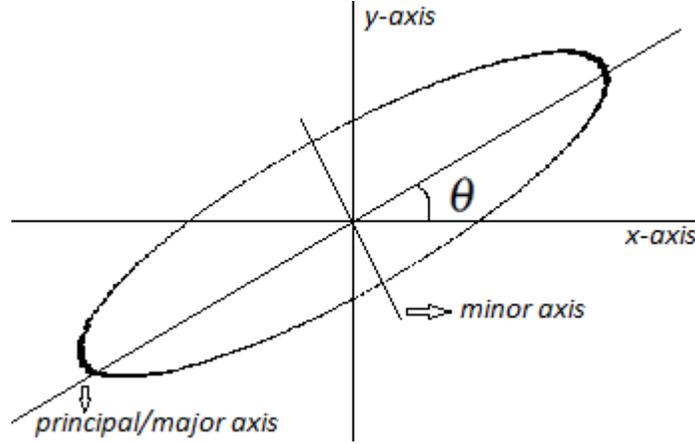


Fig. 2.1: Principal axes with the angle  $\theta$ , and the major and minor axes corresponding to the largest and smallest eigenvalues of the covariance matrix.

The difference between the angles made by the major axes can be used to align the two images. The covariance matrices  $C_T$  and  $C_R$  are given as follows:

$$C_T = \begin{bmatrix} c_{11}^T & c_{12}^T \\ c_{21}^T & c_{22}^T \end{bmatrix}, \quad C_R = \begin{bmatrix} c_{11}^R & c_{12}^R \\ c_{21}^R & c_{22}^R \end{bmatrix}, \quad (2.18)$$

where parameters  $c_{ij}^T$  and  $c_{ij}^R$  are given by:

$$\begin{aligned} c_{11}^T &= \sum_{x,y} (x - \bar{x}_T)^2 T(x,y), & c_{11}^R &= \sum_{x,y} (x - \bar{x}_R)^2 R(x,y), \\ c_{22}^T &= \sum_{x,y} (y - \bar{y}_T)^2 T(x,y), & c_{22}^R &= \sum_{x,y} (y - \bar{y}_R)^2 R(x,y), \\ c_{12}^T &= \sum_{x,y} (x - \bar{x}_T)(y - \bar{y}_T) T(x,y), & c_{12}^R &= \sum_{x,y} (x - \bar{x}_R)(y - \bar{y}_R) R(x,y), \\ c_{21}^T &= c_{12}^T, & c_{21}^R &= c_{12}^R. \end{aligned} \quad (2.19)$$

The smallest and largest eigenvalues of the covariance matrix indicate the direction of the major and the minor axes of the ellipse, respectively. Fig. 2.2 shows the results of PAT applied to  $T(\mathbf{x})$ . This method of registration is sensitive to missing or additional data. That is, if the two images differ not only in translations and rotations but also data is missing in one but not the other, the method often performs poorly. Regardless the method quickly and cheaply obtains an approximation which can serve as a good initialization for other accurate methods such as the affine and perspective registration models via an optimization approach. The optimization methods are covered in the following sections.

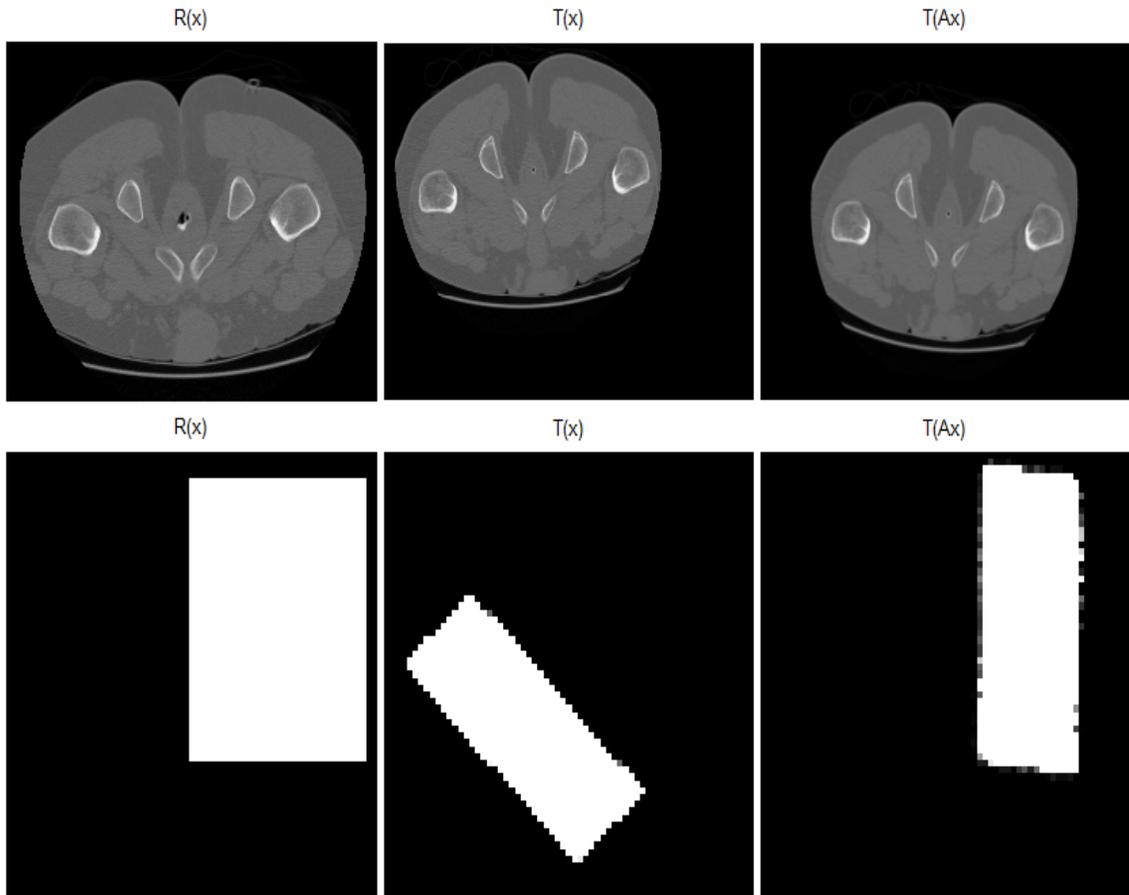


Fig. 2.2: The registration of two CT slices (top row) and two synthetic images (bottom row) using PAT. The first column shows the reference images  $R(\mathbf{x})$ , the second column shows the template images  $T(\mathbf{x})$  and the third column shows the transformed or registered template image  $T(\mathbf{Ax})$ .

Looking at the registration results by PAT on Fig. 2.2 it can be observed that the registered template image  $T(\mathbf{Ax})$  is approximately aligned with the reference  $R(\mathbf{x})$  in terms of rotation and translation, but the scaling has not improved from  $T(\mathbf{x})$  to  $T(\mathbf{Ax})$ . These results are therefore in line with the theoretical predictions elaborated above. With this method the normalized dissimilarity measure  $\frac{\|T(\mathbf{Ax}) - R(\mathbf{x})\|_{L^2}^2}{\|T(\mathbf{x}) - R(\mathbf{x})\|_{L^2}^2}$  has been reduced from the value 1 down to 0.68106 in roughly zero seconds (CPU time) for CT slices and 1 down to 0.43798 in roughly zero seconds for synthetic images.

## 2.4.2 Gradient-based Methods

In this section we cover three gradient-based methods: GD, GN and LM. We also show how LM relates to GD and GN.

### 2.4.2.1 Gradient Descent (GD)

The gradient descent method takes the negative of the gradient of the cost function as its search direction. The gradient is found by taking the first order derivative of the cost functional which results in the following:

$$\mathbf{g}(\mathbf{a}) = \mathbf{J}(\mathbf{a})^T [T(\mathbf{Ax}) - R(\mathbf{x})], \quad (2.20)$$

where  $\mathbf{J}(\mathbf{a})$  is the Jacobian matrix given by:

$$\mathbf{J}(\mathbf{a}) = \frac{\partial \mathbf{Ax}}{\partial \mathbf{a}} \cdot \nabla T(\mathbf{Ax}). \quad (2.21)$$

At iteration  $k$  the GD method employs the search direction as:

$$\mathbf{d}^k = -\mathbf{g}(\mathbf{a}^k),$$

and  $\alpha^k$  is obtained using either the exact line search methods [2, 20, 24, 25] which solve the minimization problem:

$$\alpha^k = \underset{\alpha}{\operatorname{argmin}} \quad \mathbf{J}(\mathbf{a}^k + \alpha \mathbf{d}^k), \quad (2.22)$$

or the inexact line search methods. The exact line search methods are expensive as they require many cost functional evaluations thus the inexact line search methods are widely used [17]. In [17, 25] the scaling factor at iteration  $k$  is defined by the following exponentially decreasing function:

$$\alpha^k = \frac{l}{(k+A)^p}, \quad l > 0, \quad A \geq 1, \quad 0 < p \leq 1. \quad (2.23)$$

The GD method has the property that it converges faster when far away from the optimal solution, but it becomes slow to converge when near to the optimal solution. This makes GD inferior to other methods when close to the optimal solution.

### 2.4.2.2 Gauss Newton (GN)

The Newton method makes use of both the gradient  $\mathbf{g}(\mathbf{a})$  and the Hessian  $\mathbf{H}(\mathbf{a})$  of the cost functional in order to obtain the search direction [2]. The resulting search direction is

given as:

$$\mathbf{d}^k = -\mathbf{H}(\mathbf{a}^k)^{-1} \mathbf{g}(\mathbf{a}^k). \quad (2.24)$$

The method converges very quickly when it is close to the optimal solution. However, it is computationally demanding due to the need to compute the exact Hessian matrix. As mentioned in [2] this method may be not suitable in registering two images for practical applications because computing higher derivatives is time consuming and numerically unstable. GN is a modification of the Newton method which addresses the Hessian problem found in Newton method. In the GN the Hessian is approximated using the Jacobian matrix rather than computing the exact Hessian. That is, the approximate Hessian used is

$$\mathbf{H}(\mathbf{a}) \approx \mathbf{J}(\mathbf{a})^T \mathbf{J}(\mathbf{a}). \quad (2.25)$$

This formulation of GN makes the assumption that the resulting Hessian is well-conditioned and non-singular, which is not always the case. One effective way of conditioning the approximated Hessian in GN is by using the Cholesky factorization with a predefined threshold which enforces a certain degree of diagonal dominance in the matrix [20]. Based on the properties of GD and GN, there seems to be a need for a hybrid of these two methods, one which behaves like GD when far from the solution and switches when closer to behave like GN. One improvement of GN which behaves like such a hybrid of GN and GD is the LM method.

### 2.4.2.3 Levenberg-Marquardt (LM)

The LM method is an improvement of GN which modifies the Hessian approximation  $\mathbf{J}(\mathbf{a})^T \mathbf{J}(\mathbf{a})$  by adding either a multiple of the identity matrix  $\mathbf{I}$  to give

$$\mathbf{H}(\mathbf{a}) \approx \mathbf{J}(\mathbf{a})^T \mathbf{J}(\mathbf{a}) + \mu \mathbf{I} \quad (2.26)$$

or by adding the diagonal of  $\mathbf{J}(\mathbf{a})^T \mathbf{J}(\mathbf{a})$  to give

$$\mathbf{H}(\mathbf{a}) \approx \mathbf{J}(\mathbf{a})^T \mathbf{J}(\mathbf{a}) + \mu \text{diag}(\mathbf{J}(\mathbf{a})^T \mathbf{J}(\mathbf{a})) \quad (2.27)$$

where  $\mu > 0$  is the LM parameter adjusted to guarantee that the search direction is a descent direction. The addition of the diagonal dominant matrix helps to improve the diagonal dominance of the entire Hessian approximation thus making it more positive definite. The parameter  $\mu$  gives weight to the matrix  $\mathbf{I}$  and serves an important purpose in the con-

vergence of this method. This parameter is set to be large initially so that the algorithm mimics the GD and approach the minimum quickly and it is reduced as the minimum is approached to mimic pure GN thus avoiding the zig-zag problems associated with GN. Even though these methods work so well in solving the problem they are sensitive to poor initialization due to the fact that this minimization problem is nonconvex and presents local minima as traps that catch these methods.

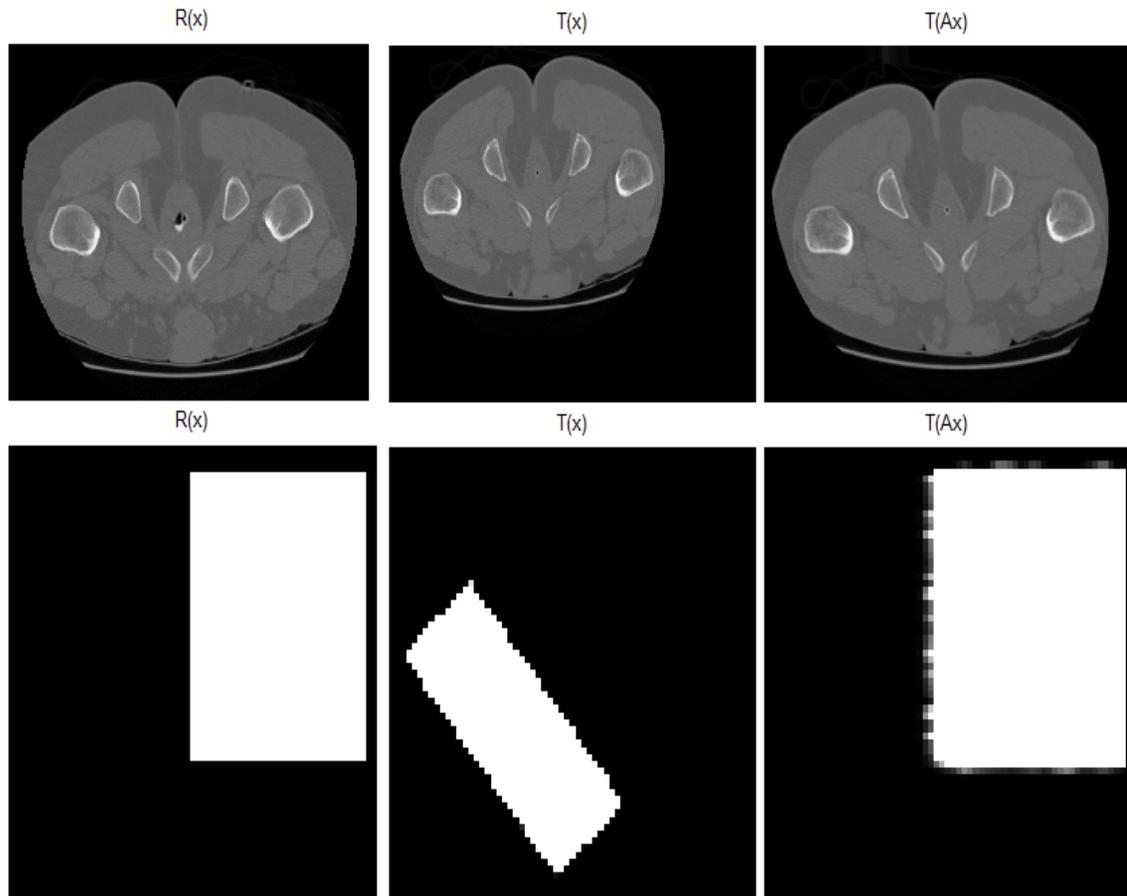


Fig. 2.3: The registration of two CT slices (top row) and two synthetic images (bottom row) using LM. The first column shows the reference images  $R(\mathbf{x})$ , the second column shows the template images  $T(\mathbf{x})$  and the third column shows the transformed or registered template image  $T(A\mathbf{x})$ .

Fig. 2.3 shows the results of using LM to register the same two CT slices and two synthetic images as those used with PAT earlier. Contrary to PAT, LM was able to correct not only global rotation and translation but also anisotropic scale and shear as well. Given that the search space allowed in the implementation is affine transformations it was expected that misalignment that is described by these four basic transformations would be corrected provided that a reasonable initialization is available. From several tests made on LM

it was observed that initialization has a big impact on the convergence to an acceptable solution. In the test shown in Fig. 2.3 the normalized dissimilarity measure  $\frac{\|T(\mathbf{Ax}) - R(\mathbf{x})\|_{L_2}^2}{\|T(\mathbf{x}) - R(\mathbf{x})\|_{L_2}^2}$  was reduced from 1 to 0.16374 in 11 seconds (CPU time) with CT slices and from 1 to 0.09730 in 9 seconds (CPU time) with synthetic images. This is a significant improvement from the figures obtained with PAT previously with regard to convergence, however it took more time to complete.

### 2.4.3 Nelder-Mead Simplex (NMS) Method

The NMS method is a popular direct search method for multidimensional unconstrained minimization problems. It maintains at each step a non-degenerate simplex, a geometric figure in  $n$ -dimensions of nonzero volume which is the convex hull of  $n + 1$  vertices [26]. There are four scalar parameters essential to the workings of the NMS method and these are:

- the coefficient of reflection  $\rho > 0$ ,
- the coefficient of expansion  $\chi > 1$ ,  $\chi > \rho$ ,
- the coefficient of contraction  $0 < \gamma < 1$  and
- the coefficient of shrinkage  $0 < \sigma < 1$ .

As mentioned in [26] the nearly universal choices used in the standard NMS method are  $\rho = 1$ ,  $\chi = 2$ ,  $\gamma = \frac{1}{2}$  and  $\sigma = \frac{1}{2}$ . More detail on the Nelder-Mead simplex method can be found in [26]. For purposes of solving the affine image registration minimization problem presented above the vertex  $\mathbf{a} = [a_1 \dots a_6]^T$  represents a 6-dimensional vector constructed from elements of the affine transformation matrix in Equation (2.4). Also we consider  $f(\mathbf{a}) = \frac{1}{2} \|T(\mathbf{Ax}) - R(\mathbf{x})\|_{L_2}^2$ . One iteration of the NMS method proceeds as outlined in [26]:

---

**Algorithm 2.1** One iteration of the NMS method.

---

**1. Ordering vertices:** order the  $n + 1$  vertices to satisfy  $f(\mathbf{a}_1) \leq f(\mathbf{a}_2) \leq \dots \leq f(\mathbf{a}_{n+1})$  with  $(\mathbf{a}_i)_{i=1,2,\dots,n+1}$  as the solution vectors to an optimization problem defined by the cost function  $f$ . Then  $\mathbf{a}_1$  is the best vertex while  $\mathbf{a}_{n+1}$  is the worst vertex.

**2. Reflection:** compute the reflection point  $\mathbf{a}_r$  from

$$\mathbf{a}_r = \bar{\mathbf{a}} + \rho(\bar{\mathbf{a}} - \mathbf{a}_{n+1}) = (1 + \rho)\bar{\mathbf{a}} - \rho\mathbf{a}_{n+1}, \quad (2.28)$$

where  $\bar{\mathbf{a}} = \frac{1}{n} \sum_{i=1}^n \mathbf{a}_i$  is the centroid of the  $n$  best point excluding vertex  $\mathbf{a}_{n+1}$ . If  $f(\mathbf{a}_1) \leq f(\mathbf{a}_r) < f(\mathbf{a}_n)$  then accept the reflected point  $\mathbf{a}_r$  and terminate the iteration.

**3. Expansion:** if  $f(\mathbf{a}_r) < f(\mathbf{a}_1)$  then calculate the expansion point  $\mathbf{a}_e$  as follows:

$$\mathbf{a}_e = \bar{\mathbf{a}} + \chi(\mathbf{a}_r - \bar{\mathbf{a}}) = (1 + \rho\chi)\bar{\mathbf{a}} - \rho\chi\mathbf{a}_{n+1}. \quad (2.29)$$

If  $f(\mathbf{a}_e) < f(\mathbf{a}_r)$  then accept  $\mathbf{a}_e$  and terminate the iteration, otherwise accept  $\mathbf{a}_r$  and terminate the iteration.

**4. Contraction:** if  $f(\mathbf{a}_r) \geq f(\mathbf{a}_n)$ , perform a contraction between  $\bar{\mathbf{a}}$  and the better of  $\mathbf{a}_{n+1}$  and  $\mathbf{a}_r$ .

(a) *Outside:* if  $f(\mathbf{a}_n) \leq f(\mathbf{a}_r) < f(\mathbf{a}_{n+1})$  then perform an outside contraction:

$$\mathbf{a}_c = \bar{\mathbf{a}} + \gamma(\mathbf{a}_r - \bar{\mathbf{a}}) = (1 + \rho\gamma)\bar{\mathbf{a}} - \rho\gamma\mathbf{a}_{n+1}. \quad (2.30)$$

If  $f(\mathbf{a}_c) \leq f(\mathbf{a}_r)$  then accept  $\mathbf{a}_c$  and terminate the iteration, otherwise go to step 5.

(b) *Inside:* if  $f(\mathbf{a}_r) \geq f(\mathbf{a}_{n+1})$  then perform an inside contraction:

$$\mathbf{a}_{cc} = \bar{\mathbf{a}} - \gamma(\bar{\mathbf{a}} - \mathbf{a}_{n+1}) = (1 - \gamma)\bar{\mathbf{a}} + \gamma\mathbf{a}_{n+1}. \quad (2.31)$$

If  $f(\mathbf{a}_{cc}) < f(\mathbf{a}_{n+1})$  accept  $\mathbf{a}_{cc}$  and terminate the iteration otherwise go to step 5.

**5. Shrinkage:** shrink the simplex as follows:

$$\mathbf{v}_i = \mathbf{a}_1 + \sigma(\mathbf{a}_i - \mathbf{a}_1), \quad (2.32)$$

where  $i = 2, \dots, n + 1$ . The unordered vertices of the simplex at the next iteration consist of  $\mathbf{a}_1, \mathbf{v}_2, \dots, \mathbf{v}_{n+1}$ .

---

Fig. 2.4 shows the results of applying the Nelder-Mead simplex method to CT slices and synthetic images for registration.

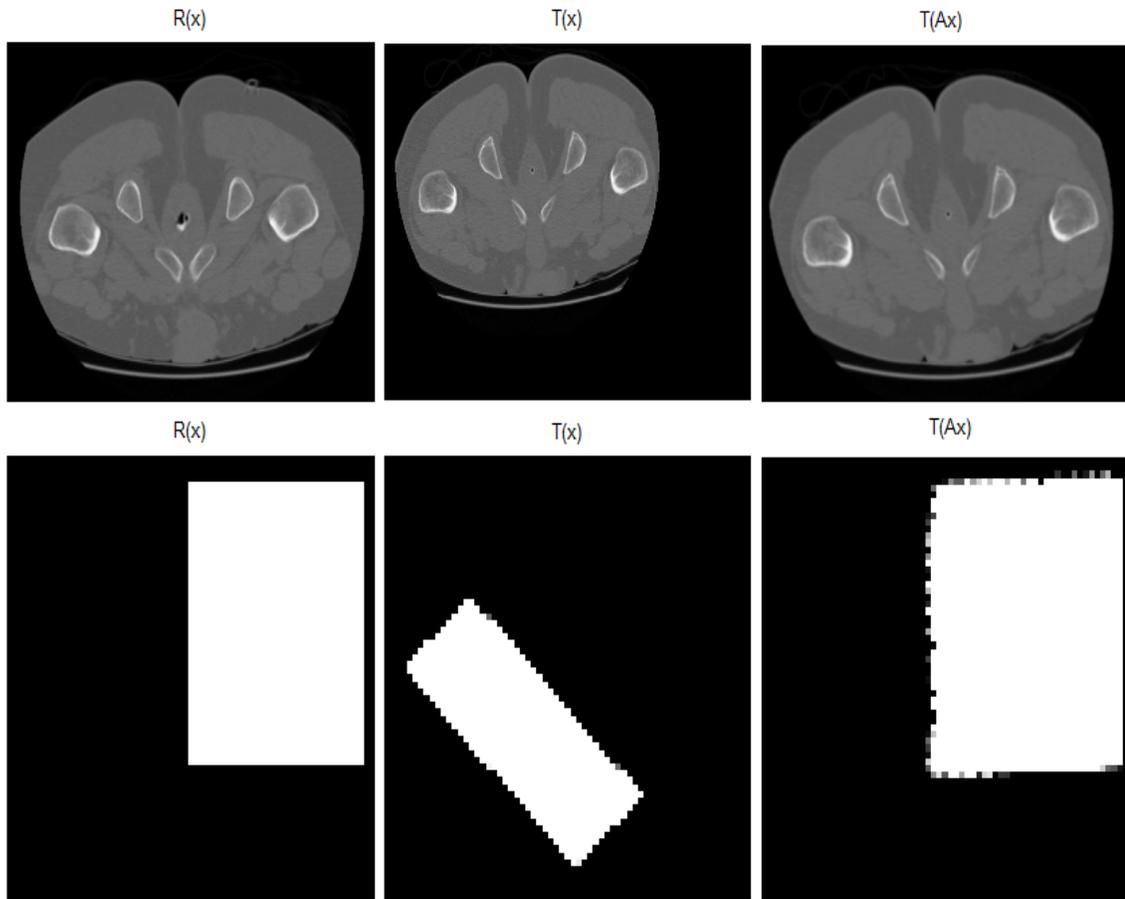


Fig. 2.4: The registration of two CT slices (top row) and two synthetic images (bottom row) using the NMS method. The first column shows the reference images  $R(\mathbf{x})$ , the second column shows the template images  $T(\mathbf{x})$  and the third column shows the transformed or registered template image  $T(\mathbf{Ax})$ .

Unlike the LM method the NMS method was found to be relatively less sensitive to bad initialization. The search space allowed in the design and implementation of this method covers only affine transformations, and as can be observed from Fig. 2.4 global rotation, translation, shear and anisotropic scale have been corrected. The normalized dissimilarity measure  $\frac{\|T(\mathbf{Ax}) - R(\mathbf{x})\|_{L^2}^2}{\|T(\mathbf{x}) - R(\mathbf{x})\|_{L^2}^2}$  was reduced from 1 to 0.16244 in 17 seconds (CPU time) for CT slices and from 1 to 0.19257 in 13 seconds (CPU time) for the synthetic images. The NMS method took is relatively slower than the LM method but is more robust to initialization than the LM method.

### 2.4.4 $\alpha$ - Branch and Bound ( $\alpha$ BB)

All the search methods that have been discussed above are local optimization methods, since they do not guarantee a global optimal solution but rather the locally optimal solution. For a nonconvex functional they often get trapped in local minima and fail to converge to the global minimum. The  $\alpha$ BB method is a very popular global optimization method that is deterministic and can guarantee  $\varepsilon$ -convergence to the global minimum within a finite number of iterations [27]. The basic principle of this method is to partition the solution domain into smaller elements  $[\mathbf{a}^L, \mathbf{a}^U]$  and process the elements sequentially. The main goal of this method is to find a tight convex under-estimator of the cost functional and use that under-estimator to locate the global minimum. Given a cost functional  $f(\mathbf{a})$ , the method uses the quadratic under-estimator to arrive at the following under-estimator of the cost functional [27]:

$$L(\mathbf{a}, \alpha) = f(\mathbf{a}) + \sum_{i=1}^6 \alpha_i (a_i^L - a_i)(a_i^U - a_i). \quad (2.33)$$

The quantity  $L(\mathbf{a}, \alpha)$  is a guaranteed under-estimator of  $f(\mathbf{a})$ , and since the quadratic term is convex the nonconvexities in  $f(\mathbf{a})$  can be overpowered with sufficiently large values of the  $\alpha_i$  parameters. Since  $L(\mathbf{a}, \alpha)$  is convex if and only if its Hessian matrix  $\mathbf{H}_L(\mathbf{a})$  is positive semi-definite, the convexity condition is derived from the relationship between the Hessian matrix of the cost functional  $\mathbf{H}_f(\mathbf{a})$  and  $\mathbf{H}_L(\mathbf{a})$  as given below:

$$\mathbf{H}_L(\mathbf{a}) = \mathbf{H}_f(\mathbf{a}) + 2\Delta, \quad (2.34)$$

where  $\Delta$  is the diagonal shift matrix, which is the diagonal matrix whose diagonal elements are the  $\alpha_i$  parameters [28]. It can be seen that increasing the values of the  $\alpha_i$  parameters increases the diagonal dominance of  $\mathbf{H}_L(\mathbf{a})$ , thus  $\mathbf{H}_L(\mathbf{a})$  can be made positive semi-definite even when  $\mathbf{H}_f(\mathbf{a})$  is not. The main task now rests on the determination of appropriate  $\alpha_i$  parameters that guarantee positive semi-definiteness of  $\mathbf{H}_L(\mathbf{a})$ . One approach for obtaining the  $\alpha_i$  parameters is by using the scaled Gerschgorin method shown below:

$$\alpha_i = \max\left\{0, -\frac{1}{2}\left(\underline{f}_{ii} - \sum_{j \neq i}^6 \max\{|f_{ij}|, |\overline{f}_{ij}|\} \frac{d_j}{d_i}\right)\right\}, \quad (2.35)$$

where  $\underline{f}_{ij}$  and  $\overline{f}^{ij}$  are the lower and upper bounds of  $\frac{\partial^2 f}{\partial a_i \partial a_j}$  as calculated by interval analysis, and  $d_i$  is the positive distance commonly given as  $d_i = a_i^U - a_i^L$  (as explained in [28]). Other tighter under-estimators have been proposed, such as the  $\gamma$ BB whose under-

estimator is given in [28] as

$$L(\mathbf{a}, \gamma) = f(\mathbf{a}) - \sum_{i=1}^6 (1 - e^{\gamma(a_i - a_i^l)})(1 - e^{\gamma(a_i^u - a_i)}). \quad (2.36)$$

Despite its strength in capturing the global optimal solution, the  $\alpha$ BB method can be slow to converge and may not always be suitable for real time and online optimization applications.

## 2.4.5 Evolutionary Programming

Evolutionary algorithms (EAs) and genetic algorithms (GAs) have been used to solve the problem of image registration in the literature, such as in [23] where a genetic algorithm is used to solve the initialization problem in active shape models and for medical image registration in [29]. These algorithms are global search methods that employ no explicit direction in their search strategy, but rather maintain a population of solutions which are evaluated using the cost functional to determine their fitness. They work on the principle of survival of the fittest, whereby the less fit the solution the less likely it is to pass its genes to the next generation. The EA discussed in this section is the one called evolutionary programming (EP) and is summarized by the following steps as described in [19]:

---

**Algorithm 2.2** Steps in Evolutionary Programming (EP).

---

- 1. Initialization:** this process involves assigning the parameters for EP, such as the population size  $\mu$ , the mutation probability  $\sigma$  and the maximum number of generation *maxgen*. Then  $\mu$  uniformly distributed individuals are generated randomly to form the initial population and their fitness values are obtained through evaluation by the cost functional. Set the number of generation to zero, *gen*=0.
  - 2. Main Loop:** repeat the following steps until *gen*>*maxgen*.
    - Step 2.1:* repeat the following operations until a new population with  $\mu$  individuals has been generated. Perform a mutation for every gene of the individual to generate a new one.
    - Step 2.2:* calculate the fitness value for every new individual.
    - Step 2.3:* combine  $\mu$  current and  $\mu$  new individuals and pick the  $\mu$  best ones to form a new population. Set *gen*=*gen*+1.
  - 3. Submission:** submit the final  $\mu$  individuals as the result of EP.
-

More details on EP and other EAs can be found in [19].

## 2.5 The Affine Pre-Registration System

The affine pre-registration system comprises three solution methods described above, namely the PAT, the NMS method and the LM method. The system first applies PAT on the input images to bring them closer to each other translationally and rotationally. Even though PAT does not correct for scaling, shear and projections, it is able to improve the initialization for the next method. Having brought them close to each other, the two images are further registered with the NMS method. The NMS method has the advantage of not being easily trapped by the local minima, although it is by no means a global optimization method. It has been argued that this method does not always guarantee convergence to the optimal solution, and that it can converge to solution which is not optimal [30]. For this reason the results are further processed by the LM method to ensure convergence to a nearby locally optimal solution.

One way of improving the results of these optimization methods is to manipulate the input images such that the nonconvexities or the local minima in the cost functional are reduced. This is usually done by incorporating the multi-resolution technique in the images during the registration process. The basic idea in multi-resolution is to register low resolution versions of images and to interpolate level-by-level to higher resolutions until the last level of the original images. Every level of resolution provides a reliable initial guess (transformation) for the next level [2, 17]. This strategy was adopted in our affine pre-registration system. The performance of the pre-registration system on the CT slices and the synthetic images is shown in Fig. 2.5.

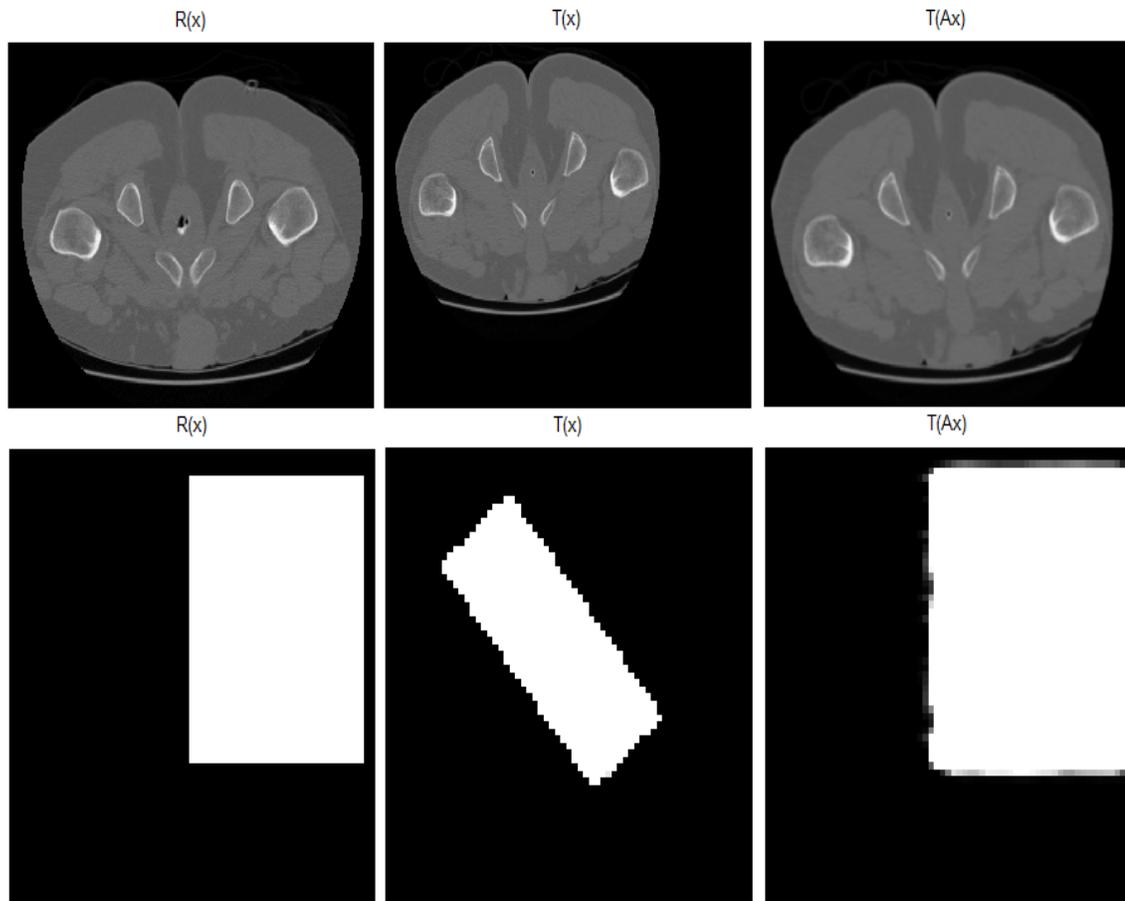


Fig. 2.5: The registration of two CT slices (top row) and two synthetic images (bottom row) using our pre-registration system. The first column shows the reference image  $R(\mathbf{x})$ , the second column shows the template image  $T(\mathbf{x})$  and the third column shows the transformed or registered template image  $T(\mathbf{Ax})$ .

The pre-registration system was tested on several images and found to be more robust relative to its constituents with regard to both initialization and convergence to a satisfactory solution. Much of the initialization strength is owed to PAT since it is immune to bad initialization. The combination of only PAT and LM was first attempted but proved to be not very robust since quite frequently the initialization provided by PAT to LM was not sufficient: the incorporation of NMS in the middle gave rise to a relatively more reliable pre-registration system. In Fig. 2.5 the same test images as the ones used for the previous methods were used and the normalized dissimilarity measure  $\frac{\|T(\mathbf{Ax}) - R(\mathbf{x})\|_{L^2}^2}{\|T(\mathbf{x}) - R(\mathbf{x})\|_{L^2}^2}$  was reduced from 1 to 0.11449 in 26 seconds (CPU time) for CT slices and from 1 to 0.077657 in 21 seconds (CPU time) for synthetic images. This is the pre-registration system that we use to initialize the deformable registration models for subsequent evaluation. The system is more robust than its individual components although it is much slower than its slowest

component. The run-time of this system is still less than the time taken by branch and bound method which takes more than a minute to run. Evolutionary algorithms also take a longer time to sufficiently minimize the cost function on average hence was not considered in building the pre-registration system.

# Chapter 3

## Variational-based Models

The deformable image registration aims to find a physically realizable deformation  $\phi(\mathbf{x})$  that transforms the template image  $T(\mathbf{x})$  into  $T(\phi(\mathbf{x}))$ , which matches the reference image  $R(\mathbf{x})$  with respect to some chosen similarity measure. It is vital that the deformation be physically realizable, which means that it is free of grid tangling and folding. Various models have been proposed to accomplish this task and most of them are based on physical models [31]. In [32] these deformable image registration models are classified based on the following transformations: *geometric transformations derived from physical models*, *geometric transformations derived from interpolation theory* and *knowledge-based geometric transformations*.

The models based on geometric transformations derived from physical models include the elastic model [3, 33], the fluid model [3, 5, 7, 34], the diffusion model [3, 35], the curvature model [3, 11, 36] and flows of diffeomorphisms [37, 38, 39, 40]. The models based on geometric transformations derived from interpolation theory include radial basis functions [41], elastic body splines [42], free form deformations [43], basis functions from signal processing [44] and locally affine models [45]. Lastly, the models based on knowledge-based geometric transformations include statistically-constrained geometric transformations [46] and geometric transformations inspired by biomechanical or biophysical models [47].

Some of the most popular models used in medical imaging applications are variational-based models, which are based on the geometrical transformations derived from physical models. In this chapter we cover four of these variational-based models, which will be used in the evaluation of the optimal control-based model. These four variational-based

models are *elastic, fluid, diffusion* and *curvature* models, details of which can be found in [3, 11]. In [3] these four models are unified such that the implementation of one can easily be changed into that of another. In all of these variational-based models the aim is to minimize the following weighted objective functional with respect to the displacement field  $\mathbf{u}(\mathbf{x})$  as outlined in [3]:

$$J[\mathbf{u}(\mathbf{x})] = \frac{1}{2} \|T(\phi(\mathbf{x})) - R(\mathbf{x})\|_{L^2}^2 + \alpha S[\mathbf{u}(\mathbf{x})], \quad \alpha > 0, \quad (3.1)$$

where  $\phi(\mathbf{x}) = \mathbf{x} - \mathbf{u}(\mathbf{x})$  in the case of variational-based models,  $\alpha$  is the regularity or smoothness constant and  $S[\mathbf{u}(\mathbf{x})]$  is the regularity or smoothness measure of the displacement field  $\mathbf{u}(\mathbf{x})$ .

As mentioned in [48], solving variational calculus and optimal control problems with direct methods requires: (1) a *partial differential equation (PDE) solver*, (2) a *numerical integration tool* and (3) an *optimization algorithm*. All our implementations were carried out in the Matlab Language and the Matlab PDE solver was obtained from source code available in the book by Strang [49]. However, some of the codes were modified to fit our problems (such as the finite element method code given in section 3.6 of the book). To further confirm if the PDE solvers are working correctly we used them to solve some simple PDEs whose exact solutions are known, and compared the approximate solutions from the solvers with the exact solutions. The Gauss-Legendre quadrature scheme was used for numerical integrations and the Matlab source code was obtained from the internet [50]. The optimization algorithm adopted for all our models is the gradient-based algorithm.

The necessary conditions for optimality are obtained by taking the *Gâteaux* derivative of Equation (3.1) to obtain the following general form of Euler-Lagrange equation, as given in [3]:

$$\mathbf{f}(\mathbf{x}, \mathbf{u}(\mathbf{x})) + \alpha \Lambda[\mathbf{u}(\mathbf{x})] = 0, \quad (3.2)$$

where  $\mathbf{f}$  and  $\Lambda$  denote the *Gâteaux* derivatives of the  $L^2$ -norm term and the regularity term  $S[\mathbf{u}(\mathbf{x})]$  in Equation (3.1) respectively. That is

$$\mathbf{f}(\mathbf{x}, \mathbf{u}(\mathbf{x})) = (T(\phi(\mathbf{x})) - R(\mathbf{x})) \cdot \nabla T(\phi(\mathbf{x})). \quad (3.3)$$

Having obtained the Euler-Lagrange equation for Equation (3.1) above the steps for registering the template image to the reference image are summarized as follows.

---

**Algorithm 3.1** Steps for registering the template image to the reference image.

---

- **Initialization:**
    - (a) Initialize the deformation  $\mathbf{u}(\mathbf{x})$  and set parameters like  $\alpha$ ,  $\lambda$  and  $\mu$ .
    - (b) Use  $\phi(\mathbf{x}) = \mathbf{x} - \varepsilon\mathbf{u}(\mathbf{x})$  to evaluate Equation (3.1), where  $\varepsilon$  is the scaling factor chosen as mentioned in Chapter 2 for gradient descent (GD) method.
  - **Main Loop:**
    - (c) Compute  $\mathbf{f}$  using as shown in Equation (3.3).
    - (d) Solve for the displacement field  $\mathbf{u}(\mathbf{x})$  from Equation (3.2).
    - (e) Solve for the deformation  $\phi(\mathbf{x}) = \mathbf{x} - \varepsilon\mathbf{u}(\mathbf{x})$  and evaluate Equation (3.1).
    - (f) If the stopping criteria not met, go to (c) else submit the current solution  $\phi(\mathbf{x})$ .
  - **End Loop**
- 

In the following sections we describe the four variational-based models and condition the associated equations to be ready for the solvers mentioned above. For every model we use two synthetic images and two CT slices, as part of data set provided by iThemba LABS, to show its behavior. The diagrams shown in this chapter and the next are not part of the results for this work but serve for demonstration purposes.

### 3.1 Elastic Model

The elastic model was proposed by Briot [33] and is identified by the smoothness measure which has the form of a linear elastic potential of the displacement field  $S[\mathbf{u}(\mathbf{x})]$ . As in the physics of elastic bodies, the elastic potential penalizes motions that lead to physically unrealizable deformations such as ones which lead to objects splitting up. This property of the elastic potential becomes useful in deformable image registration since it penalizes displacement fields that lead to grid tangling, grid folding and other unwanted deformations. Some of the unwanted deformations include splitting of image objects into incoherent pieces or piling up of image objects into a point, thus not conserving information in the image.

Even though this model is favorable with regard to keeping the transformed image intact and conserving pixels, it is significantly stiff and only allows small deformations [3]. This presents difficulties in using this model for applications that involve large deformations, such as for soft tissue registration. This has led to the elastic model being used in appli-

cations that involve small deformations, such as in brain registration [3]. Although this model may exhibit poor convergence in soft tissue applications and in highly deformable object registration, it always has good deformation quality measure that makes it a good candidate for evaluation in this work with regard to the deformation quality measure. This model also penalizes the affine and perspective transformations, and pre-registration is therefore necessary to accelerate convergence to an optimal solution.

### 3.1.1 Formulation and Implementation

The elastic model is distinguished by a deformation regularizer motivated by a linearized elastic potential of the displacement field shown below and given in [3]:

$$S^{\text{elas}}[\mathbf{u}(\mathbf{x})] = \int_{\Omega} \left[ \frac{\mu}{4} \sum_{j,k=1}^2 (\partial_{x_j} u_k + \partial_{x_k} u_j)^2 + \frac{\lambda}{2} (\nabla \cdot \mathbf{u}(\mathbf{x}))^2 \right] d\mathbf{x}, \quad (3.4)$$

where  $\mu$  and  $\lambda$  are called *Lamé-constants* and reflect the material properties. The abbreviation  $\partial_z$  is used to denote  $\frac{\partial}{\partial z}$  from here onwards. Combined with the *SSD* similarity measure, the regulated cost functional becomes

$$J^{\text{elas}}[\mathbf{u}(\mathbf{x})] = \|T(\phi(\mathbf{x})) - R(\mathbf{x})\|_{L^2}^2 + \alpha \int_{\Omega} \left[ \frac{\mu}{4} \sum_{j,k=1}^2 (\partial_{x_j} u_k + \partial_{x_k} u_j)^2 + \frac{\lambda}{2} (\nabla \cdot \mathbf{u}(\mathbf{x}))^2 \right] d\mathbf{x}, \quad (3.5)$$

and the elastic image registration problem can be stated as seeking the displacement field  $\mathbf{u}(\mathbf{x})$  that minimizes  $J^{\text{elas}}[\mathbf{u}(\mathbf{x})]$ . The most common approach is that of gradient-based optimization methods, which start by deriving the optimality conditions. The first order necessary conditions for optimality are obtained by taking the *Gâteaux* derivative of the cost functional  $J^{\text{elas}}[\mathbf{u}(\mathbf{x})]$  to obtain the following Euler-Lagrange equation as given in [3]:

$$(R(\mathbf{x}) - T(\phi(\mathbf{x}))) \nabla T(\phi(\mathbf{x})) + \alpha (\mu \Delta \mathbf{u}(\mathbf{x}) + (\lambda + \mu) \nabla (\nabla \cdot \mathbf{u}(\mathbf{x}))) = 0. \quad (3.6)$$

The solutions of this equation either minimize or maximize the cost functional. There are various ways or methods of obtaining the solutions of partial differential equations (PDEs) such as the Euler-Lagrange equation above. The most general and robust methods are numerical methods such as the finite difference method (FDM) [1], the successive over-

relaxation (SOR) method [32] and the finite element method (FEM) [51]. In this work, we adopt FEM for finding solutions to all PDEs that we come across and the choice of FEM is motivated by the simplicity it offers in handling boundary conditions as compared to FDM. There are fast solvers for PDEs other than FEM, but since convergence time is not of essence in this evaluation we are justified to proceed with FEM.

The Euler-Lagrange equation above can be decoupled into systems of simple first order equations. First we represent the Laplacian operator above in terms of curl, divergence and gradient, and this leads to the following:

$$(\lambda + 2\mu)\nabla\nabla \cdot \mathbf{u}(\mathbf{x}) - \mu\nabla \times \nabla \times \mathbf{u}(\mathbf{x}) = \mathbf{f}(\mathbf{x}) = \frac{1}{\alpha}T(\phi(\mathbf{x}))\nabla T(\phi(\mathbf{x})). \quad (3.7)$$

We then introduce two scalar fields  $p$  and  $q$ . Let  $\mathbf{f} = \frac{1}{\alpha}T(\phi(\mathbf{x}))\nabla T(\phi(\mathbf{x}))$  and drop off the arguments for convenience to obtain this linear equation:

$$(\lambda + 2\mu)\nabla p - \mu\nabla \times q = \mathbf{f}, \quad (3.8)$$

together with the complementary *div-curl* system below:

$$\nabla \cdot \mathbf{u} = p, \quad \nabla \times \mathbf{u} = q. \quad (3.9)$$

Equations (3.8) and (3.9) are solved by the FEM to obtain the displacement field  $\mathbf{u}$  to be used in transforming the template image to match the reference image. The div-curl system can be transformed into a system of Poisson equations which are easily solvable by custom Poisson solvers, as follows:

$$\Delta u_x = \frac{\partial p}{\partial x} - \frac{\partial q}{\partial y}, \quad \Delta u_y = \frac{\partial p}{\partial y} + \frac{\partial q}{\partial x}, \quad (3.10)$$

where  $u_x$  and  $u_y$  are the  $x$  and  $y$  components of the displacement vector field  $\mathbf{u}$ . Once the displacement field is obtained it is used to obtain the deformation field  $\phi(\mathbf{x})$ , which in turn transforms the template image to match the reference image. This process is done recursively with the help of the gradient algorithm as the optimization method adopted in this work. Fig. 3.1 shows examples of this model applied to two CT slices and two synthetic images for registration.

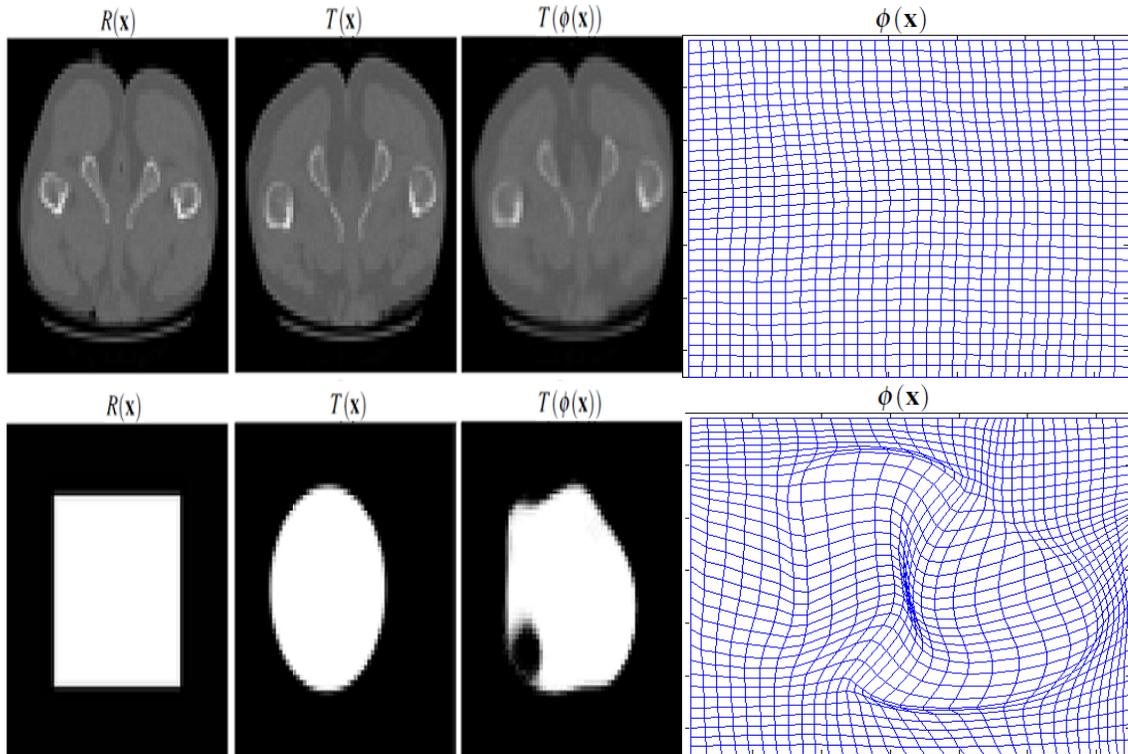


Fig. 3.1: The registration of two CT slices (top row) and two synthetic images (bottom row) using the elastic model. The first column shows the reference images  $R(\mathbf{x})$ , the second column shows the template images  $T(\mathbf{x})$ , the third column shows transformed or registered template image  $T(\phi(\mathbf{x}))$  and the last column shows the resulting deformation  $\phi(\mathbf{x})$ .

The normalized dissimilarity measure  $\frac{\|T(\phi(\mathbf{x})) - R(\mathbf{x})\|_{L^2}^2}{\|T(\mathbf{x}) - R(\mathbf{x})\|_{L^2}^2}$  decreased from 1 to 0.5506 for the CT slices and from 1 to 0.7848 for the synthetic images. The parameter settings for the CT slices were  $\lambda = 0.02$ ,  $\mu = 0.02$  and  $\alpha = 1$ , while for the two synthetic images they were  $\lambda = 0.01$ ,  $\mu = 0.01$  and  $\alpha = 0.01$ . Looking at these parameter settings in relation to the observations on Fig. 3.1 we notice that when the parameters are set to larger values the model becomes stiff and hardly optimizes the similarity measure. However, with small parameter values the regularization becomes insignificant and bad deformations start to occur. This is one of the problems mentioned in [1] on tuning the regularity weight. No pre-registration was used for initialization in these two examples.

## 3.2 Fluid Model

The Fluid model was proposed by Christensen [34] and is very similar in structure and formulation to the elastic model, with the only exception being that instead of regulariz-

ing the displacement field  $\mathbf{u}(\mathbf{x})$ , the fluid model regulates the velocity field,  $\mathbf{v}(\mathbf{x}, t)$ . The velocity field is obtained from the displacement field by computing the time derivative of the displacement field shown below [3, 11]:

$$\mathbf{v}(\mathbf{x}, t) = \partial_t \mathbf{u}(\mathbf{x}, t) + \nabla \mathbf{u}(\mathbf{x}, t) \cdot \mathbf{v}(\mathbf{x}, t). \quad (3.11)$$

The elastic potential of the velocity field has the benefit of permitting more flexibility in deformations, which allows the fluid model to account for larger deformations in objects than the elastic model [3]. However this comes at the cost of a decreased deformation quality measure. This follows from the fact that when a regular grid is subjected to more deformation then the regularity of the grid is increasingly compromised. Despite its flexibility the fluid model is reported to be very time demanding, so lighter alternatives like demon registration are preferred in the face of convergence time [3]. However, demon registration is not part of this work since convergence time has not been chosen as one of the performance criteria in the evaluation. Similar to the elastic model, the fluid model penalizes the affine and perspective transformations, and thus necessitates pre-registration for good initialization.

### 3.2.1 Formulation and Implementation

The fluid model is characterized by the following regularity measure [3, 11]:

$$S^{\text{fluid}}[\mathbf{u}(\mathbf{x})] = S^{\text{elas}}[\mathbf{v}(\mathbf{x})] = \int_{\Omega} \left[ \frac{\mu}{4} \sum_{j,k=1}^2 (\partial_{x_j} v_k + \partial_{x_k} v_j)^2 + \frac{\lambda}{2} (\nabla \cdot \mathbf{v}(\mathbf{x}))^2 \right] d\mathbf{x}, \quad (3.12)$$

where  $v_k$  is the  $k^{\text{th}}$  component of the velocity field  $\mathbf{v}(\mathbf{x})$ . The deformable image registration based on the fluid model then seeks the displacement field  $\mathbf{u}(\mathbf{x})$  which minimizes the following cost functional:

$$J^{\text{fluid}}[\mathbf{u}(\mathbf{x})] = \|T(\phi(\mathbf{x})) - R(\mathbf{x})\|_{L^2}^2 + \alpha \int_{\Omega} \left[ \frac{\mu}{4} \sum_{j,k=1}^2 (\partial_{x_j} v_k + \partial_{x_k} v_j)^2 + \frac{\lambda}{2} (\nabla \cdot \mathbf{v}(\mathbf{x}))^2 \right] d\mathbf{x}, \quad (3.13)$$

where  $\mathbf{v}(\mathbf{x})$  relates to  $\mathbf{u}(\mathbf{x})$  through Equation (3.11). The necessary conditions for optimality are given by the following Euler-Lagrange equation [3, 11]:

$$(R(\mathbf{x}) - T(\phi(\mathbf{x})))\nabla T(\phi(\mathbf{x})) + \alpha(\mu\Delta\mathbf{v}(\mathbf{x}) + (\lambda + \mu)\nabla\nabla \cdot \mathbf{v}(\mathbf{x})) = 0. \quad (3.14)$$

This equation has the similar structure to that in the elastic model above, and therefore the same procedure as in the elastic model can be used to obtain the solution. Fig. 3.2 shows examples of this model as applied on two CT slices and two synthetic images for registration.

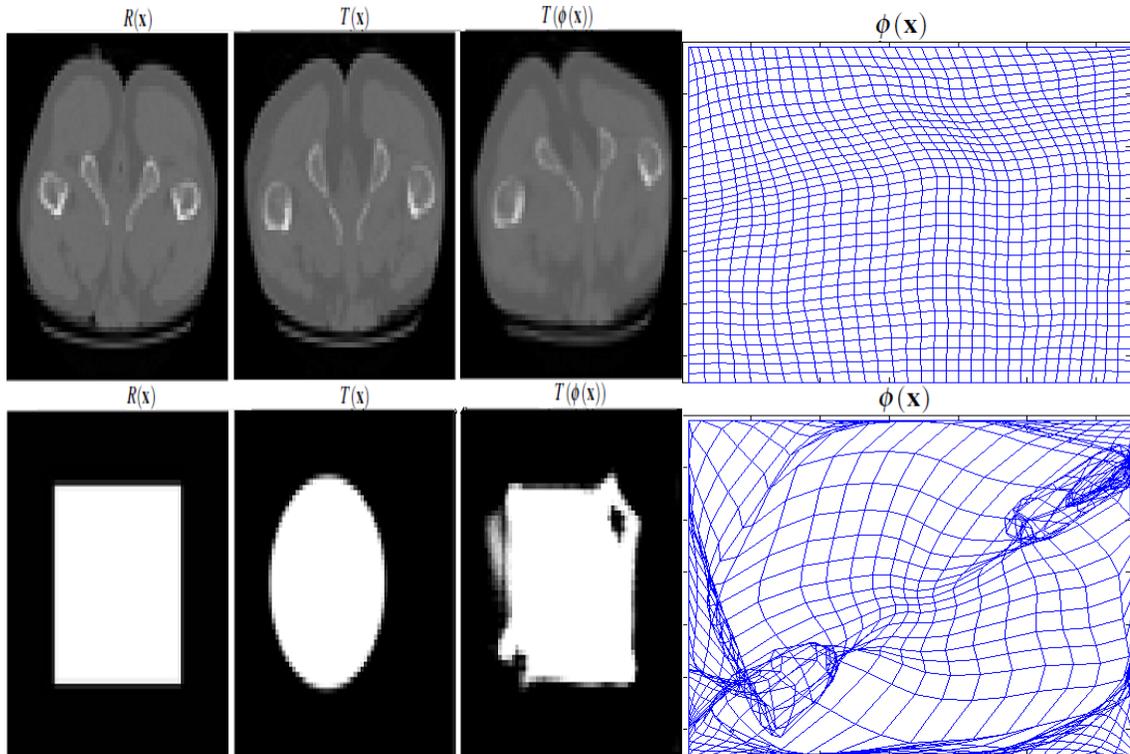


Fig. 3.2: The registration of two CT slices (top row) and two synthetic images (bottom row) using the fluid model. The first column shows the reference images  $R(\mathbf{x})$ , the second column shows the template images  $T(\mathbf{x})$ , the third column shows transformed or registered template image  $T(\phi(\mathbf{x}))$  and the last column shows the resulting deformation  $\phi(\mathbf{x})$ .

With the same parameter settings as those set in the elastic model we observe that the fluid model is able to reduce the normalized dissimilarity measure  $\frac{\|T(\phi(\mathbf{x})) - R(\mathbf{x})\|_{L^2}^2}{\|T(\mathbf{x}) - R(\mathbf{x})\|_{L^2}^2}$  from 1 to 0.4403 for the two CT slices and from 1 to 0.2810 for the two synthetic images. This more reduction in dissimilarity however comes at the cost of bad deformation in the synthetic images. It can be observed that the fluid is less stiff than the elastic model mentioned before. No pre-registration was used for initialization in these two examples.

### 3.3 Diffusion Model

The diffusion model proposed by Thirion in [35] is based on the magnitude of the gradient of the displacement field  $\mathbf{u}(\mathbf{x})$ . This is another way to penalize the physically unrealizable deformations which occur due to ill-exposure of the image registration problem. The Euler-Lagrange equations resulting from this construction are very similar in structure to the diffusion equation, hence the model being named the diffusion model. This model possesses interesting features of speed and flexibility thus making it very attractive for high-resolution applications such as 3D magnetic resonance imaging in conjunction with breast cancer surgery [3]. Similar to elastic and fluid models the diffusion model penalizes the affine and perspective transformations, and therefore requires the pre-registration step to aid in fast convergence.

#### 3.3.1 Formulation and Implementation

As mentioned in [3] the diffusion model is based on the following regularizer:

$$S^{\text{diff}}[\mathbf{u}] = \frac{1}{2} \sum_{l=1}^2 \int_{\Omega} \|\nabla u_l\|_{L^2}^2 d\mathbf{x}. \quad (3.15)$$

The diffusion based image registration problem proceeds by seeking the displacement field  $\mathbf{u}(\mathbf{x})$  minimizing the following weighted cost functional:

$$J^{\text{diff}}[\mathbf{u}(\mathbf{x})] = \|T(\phi(\mathbf{x})) - R(\mathbf{x})\|_{L^2}^2 + \frac{\alpha}{2} \sum_{l=1}^2 \int_{\Omega} \|\nabla u_l\|_{L^2}^2 d\mathbf{x}. \quad (3.16)$$

The resulting Euler-Lagrange equation has the following form, given in [3, 11], which is similar to the standard diffusion equation:

$$(R(\mathbf{x}) - T(\mathbf{x} - \mathbf{u}(\mathbf{x})))\nabla T(\mathbf{x} - \mathbf{u}(\mathbf{x})) + \alpha\Delta\mathbf{u}(\mathbf{x}) = 0. \quad (3.17)$$

Letting  $f = \frac{1}{\alpha}T(\mathbf{x} - \mathbf{u}(\mathbf{x}))\nabla T(\mathbf{x} - \mathbf{u}(\mathbf{x}))$  as before and dropping off the arguments we get the Poisson equation below:

$$-\Delta\mathbf{u} = \mathbf{f}, \quad (3.18)$$

which is easily solved by FEM to obtain the displacement field  $\mathbf{u}$  for deforming the template image. It is intuitive to expect this model to be fast given that only Equation (3.18) has to be solved, unlike in the elastic and fluid models where there are more equations to be solved. With the displacement field obtained the gradient algorithm is used to minimize the cost functional (3.16), and Fig. 3.3 shows examples where this model is applied to two CT slices and two synthetic images for registration.

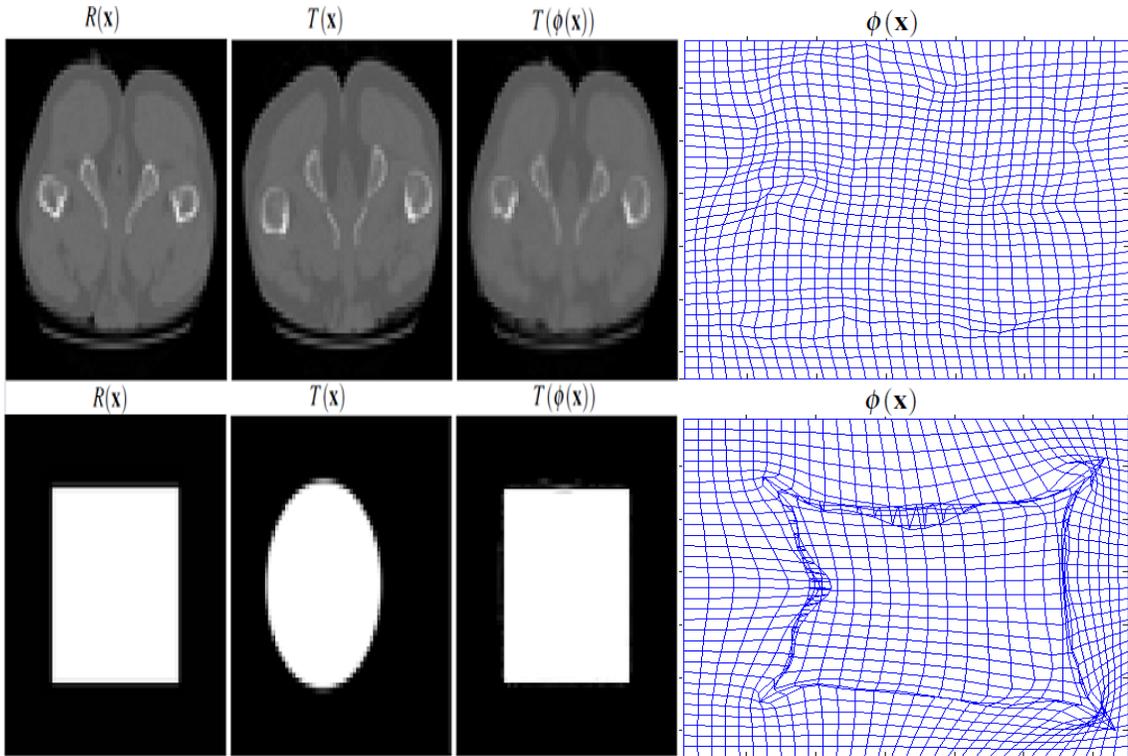


Fig. 3.3: The registration of two CT slices (top row) and two synthetic images (bottom row) using the diffusion model. The first column shows the reference images  $R(\mathbf{x})$ , the second column shows the template images  $T(\mathbf{x})$ , the third column shows transformed or registered template image  $T(\phi(\mathbf{x}))$  and the last column shows the resulting deformation  $\phi(\mathbf{x})$ .

Diffusion was found to be significantly faster than the first two models, and is due to there being fewer equations to solve. This model was able to reduce the normalized dissimilarity measure  $\frac{\|T(\phi(\mathbf{x})) - R(\mathbf{x})\|_{L^2}^2}{\|T(\mathbf{x}) - R(\mathbf{x})\|_{L^2}^2}$  from 1 to 0.092465 for the CT slices and from 1 to 0.008001 for synthetic images with the parameter set at  $\alpha = 0.0005$ . This is a significant reduction in the dissimilarity with a visually acceptable deformation in the CT slices but not in the synthetic images. Grid folding can be observed in the registration of synthetic images and as such it is not a good deformation. No pre-registration was used for initialization in these two examples.

### 3.4 Curvature Model

The curvature model proposed by Fischer and Modersitzki [36] is based on the magnitude of the Laplacian of the displacement field [5]. This model can be viewed as an approximation of the curvature of the components of the displacement field that penalizes oscillations [3]. Modersitzki in [3] shows that the curvature model, unlike other models above, has a non-trivial kernel containing affine and perspective transformations:

$$S^{\text{curv}}[\mathbf{Ax} + \mathbf{b}] = 0. \quad (3.19)$$

Thus in contrast to many other nonlinear registration models, including the elastic, fluid and diffusion models, the curvature model does not necessarily require affine pre-registration step to be successful.

#### 3.4.1 Formulation and Implementation

The curvature model is given by the following regularizer [3, 11]:

$$S^{\text{curv}}[\mathbf{u}] = \frac{1}{2} \sum_{l=1}^2 \int_{\Omega} (\Delta u_l)^2 d\mathbf{x}. \quad (3.20)$$

The problem of image registration based on this model can be stated as seeking the displacement field  $\mathbf{u}(\mathbf{x})$  which minimizes the cost functional

$$J^{\text{curv}}[\mathbf{u}(\mathbf{x})] = \|T(\phi(\mathbf{x})) - R(\mathbf{x})\|_{L^2}^2 + \frac{\alpha}{2} \sum_{l=1}^2 \int_{\Omega} (\Delta u_l)^2 d\mathbf{x}. \quad (3.21)$$

The resulting Euler-Lagrange equation is the following bi-harmonic equation, as given in [3, 11, 36]:

$$(R(\mathbf{x}) - T(\mathbf{x} - \mathbf{u}(\mathbf{x}))) \nabla T(\mathbf{x} - \mathbf{u}(\mathbf{x})) + \alpha \Delta^2 \mathbf{u}(\mathbf{x}) = 0. \quad (3.22)$$

This equation can be restructured into a system of two standard Poisson equations as follows, with the help of a vector field  $\mathbf{p}$ :

$$-\Delta \mathbf{p} = \mathbf{f}, \quad -\Delta \mathbf{u} = \mathbf{p}, \quad (3.23)$$

with  $\mathbf{f} = (R(\mathbf{x}) - T(\mathbf{x} - \mathbf{u}(\mathbf{x})))\nabla T(\mathbf{x} - \mathbf{u}(\mathbf{x}))$ . These two Poisson equations are solved by the FEM to obtain the displacement field  $\mathbf{u}$ . Fig. 3.4 shows examples of this model applied to two CT slices and two synthetic images for registration.

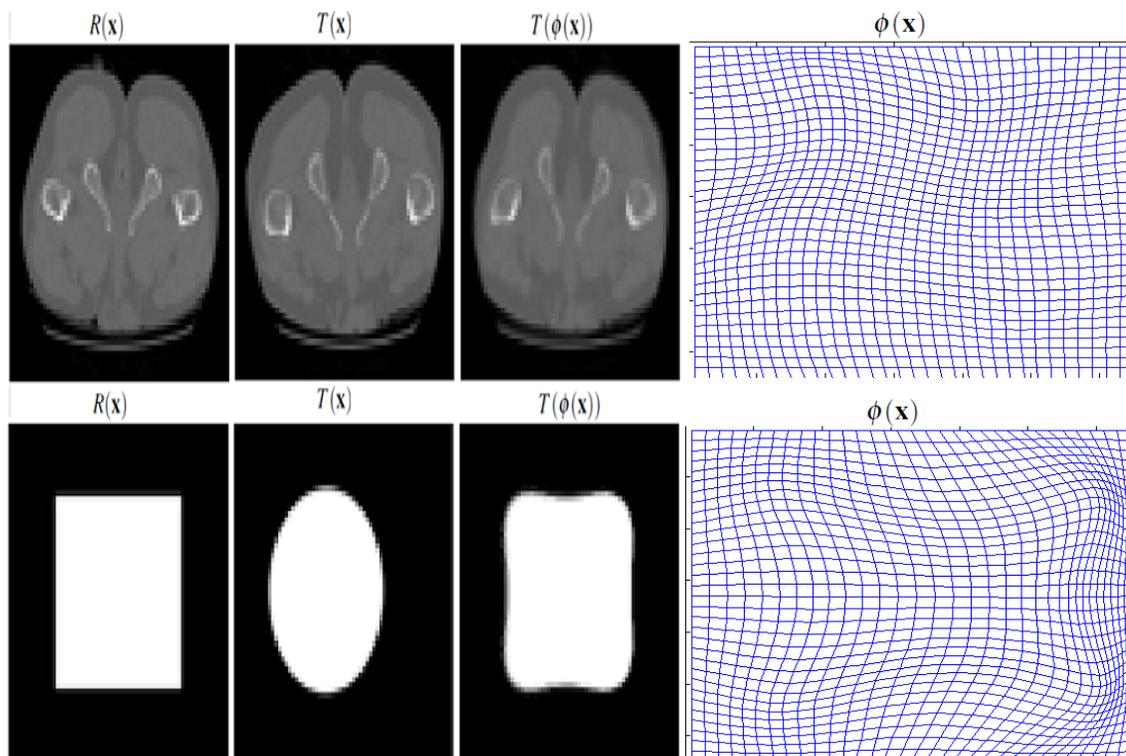


Fig. 3.4: The registration of two CT slices (top row) and two synthetic images (bottom row) using the curvature model. The first column shows the reference images  $R(\mathbf{x})$ , the second column shows the template images  $T(\mathbf{x})$ , the third column shows transformed or registered template image  $T(\phi(\mathbf{x}))$  and the last column shows the resulting deformation  $\phi(\mathbf{x})$ .

The curvature model reduced the normalized dissimilarity measure  $\frac{\|T(\phi(\mathbf{x})) - R(\mathbf{x})\|_{L^2}^2}{\|T(\mathbf{x}) - R(\mathbf{x})\|_{L^2}^2}$  from 1 to 0.23090 for CT slices and from 1 to 0.093247 for synthetic images with the same parameter setting as the diffusion model above. This model was able to produce visually acceptable deformations for both registrations. No pre-registration was used for initialization in these two examples.

# Chapter 4

## Optimal Control-based Model

The variational formulation of the deformable image registration problem is widely used to find the matching transformations in a number of approximate methods, such as with the models discussed in the previous chapter. Many of these approaches require high regularity in the control variables [8]. Very closely related to the variational-based models for solving the deformable image registration problem is the optimal control-based model. In [1, 8, 9] the problem of deformable image registration is formulated as an optimal control problem and the approach is based on the grid deformation method (GDM). This method has its origin in the field of differential geometry, as mentioned in [52], and it was formulated for grid generation by Liao and Anderson in [53]. This chapter covers the concepts of the optimal control-based model for solving the deformable image registration problem and shows examples of it being applied to sample images.

For implementation of models in this chapter, the same tools were used as for the previous chapter. Some of the details left out in the previous chapter, such as how the Poisson equations were reduced to the level of implementation by the finite element method, are briefly described in this chapter.

### 4.1 Grid Deformation Method

The GDM is used for construction of differentiable and invertible transformations to solve grid adaptation problems [9]. This method has some interesting properties: (a) it offers a way of generating a grid of the desired density distribution that is free from grid folding, and (b) it gives direct control over the cell size of the adaptive grid and determines the

node velocities directly. This is achieved through the use of a positive monitor function which disallows grid folding and provides a way to control the ratio of the areas between the original and the transformed grid [1]. Unlike the variational-based models in the previous chapter the GDM does not require the use of any regularization term [31]. These properties of the grid deformation method are favorable to the aim of the deformable image registration of seeking grid transformations which do not result in grid folding and tangling.

The GDM is based on a solid mathematical foundation. In particular, it accounts for local volume changes through the divergence of the transformation and it also accounts for local rotation through the curl vector of the transformation. The GDM is based on a linear differential system, thus its numerical implementation is fast, stable, simple and robust [1]. As mentioned in [1] this method is general in the sense that it may be used in any optimization problem that involves motion estimation. The basic idea in this method is to move the nodes with correct velocities so that the nodal mapping has a desirable Jacobian determinant [9]. It uses some ideas in Moser's deformation method which constructs a differentiable and invertible transformation between two domains equipped with Riemannian metrics, which deforms the volume element from one to the other [1].

There are three versions of GDM, the first two of which are static and the last is dynamic. Unlike the static versions, which cannot handle problems with moving boundaries very well, the dynamic version is appropriate for problems with moving boundaries since it has real time adaptation based on its formulation. This dynamic version of GDM was used for solving the deformable image registration problem in [54]. The least squares finite element method (LSFEM) implementation approach of this dynamic version can be found in [52]. This chapter covers briefly these three versions. We adopt the first version in this work: the one used in the formulation of the deformable image registration problem as an optimal control problem in [1, 8, 9].

### 4.1.1 The Deformation Method: Version One

The first version is one of the static versions of the grid deformation method where the transformation Jacobian determinant is specified on the old grid  $\mathbf{x}$  before adaptation. It was developed in [53, 55, 56, 57] and the 2D version was proposed in [58]. The description of this version proceeds by letting  $f(\mathbf{x})$  denote a positive monitor function, defined on the image domain  $\Omega$ , and satisfies  $\int_{\Omega} (f(\mathbf{x}) - 1) d\mathbf{x} = 0$ . Then there exists an injective

transformation  $\phi : \Omega \mapsto \Omega$  such that:

$$\det \nabla \phi(\mathbf{x}) = f(\mathbf{x}) > 0, \quad \forall \mathbf{x} \in \Omega. \quad (4.1)$$

The GDM then employs the monitor function  $f(\mathbf{x})$  to generate a time-dependent mapping  $\phi(t, \mathbf{x})$  that transforms points,  $\mathbf{x} \in \Omega$ , in a desired way. This mapping can be obtained by solving the following nonlinear ordinary differential equation:

$$\begin{aligned} \frac{\partial}{\partial t} \phi(t, \mathbf{x}) &= \mathbf{h}(t, \phi(t, \mathbf{x})), \quad 0 < t \leq 1, \\ \phi(0, \mathbf{x}) &= \mathbf{x}, \\ \mathbf{h}(t, \mathbf{x}) &= \frac{\mathbf{u}(t, \mathbf{x})}{t + (1-t)f(\mathbf{x})}, \quad 0 < t \leq 1, \end{aligned} \quad (4.2)$$

where  $\mathbf{u}(\mathbf{x})$  satisfies the following linear partial differential equations:

$$\begin{aligned} \nabla \cdot \mathbf{u}(\mathbf{x}) &= f(\mathbf{x}) - 1, \quad \forall \mathbf{x} \in \Omega, \\ \nabla \times \mathbf{u}(\mathbf{x}) &= 0, \quad \forall \mathbf{x} \in \Omega, \\ \mathbf{n} \cdot \mathbf{u}(\mathbf{x}) &= 0, \quad \forall \mathbf{x} \in \Gamma, \end{aligned} \quad (4.3)$$

with  $\mathbf{n}$  as an outward normal unit vector and  $\Gamma$  the boundary of the domain  $\Omega$ . The required mapping or transformation is then given by  $\phi(1, \mathbf{x})$ . Using these sets of equations as path constraints the deformable image registration problem is formulated as an optimal control problem in [1, 8, 9]. More details on how to choose the monitor function can be found in [1, 52, 59].

### 4.1.2 The Deformation Method: Version Two

The second version of GDM is also static but differs from the first in that the Jacobian is specified on the new grid coordinates before grid refinement [59]. It is required that the monitor function  $f$  be normalized and that the following condition holds over the entire domain  $\Omega$ :

$$\int_{\Omega} \left( \frac{1}{f} - 1 \right) d\mathbf{x} = 0. \quad (4.4)$$

As in the first version there is a need to find a transformation  $\phi$  such that Equation (4.1) holds, and this requires the solution of the following nonlinear ordinary differential equa-

tion:

$$\begin{aligned}\frac{\partial}{\partial t}\phi(t, \mathbf{x}) &= \mathbf{h}(t, \phi(t, \mathbf{x})), & 0 < t \leq 1, \\ \phi(0, \mathbf{x}) &= \mathbf{x}, \\ \mathbf{h}(t, \mathbf{x}) &= \frac{\mathbf{u}(t, \mathbf{x})}{\frac{t}{f(\phi(t, \mathbf{x}))} + (1-t)} & 0 < t \leq 1,\end{aligned}\tag{4.5}$$

where  $\mathbf{u}(\mathbf{x})$  satisfies the linear partial differential equation

$$\begin{aligned}\nabla \cdot \mathbf{u}(\mathbf{x}) &= 1 - \frac{1}{f(\mathbf{x})}, \quad \forall \mathbf{x} \in \Omega, \\ \nabla \times \mathbf{u}(\mathbf{x}) &= 0, \quad \forall \mathbf{x} \in \Omega, \\ \mathbf{n} \cdot \mathbf{u}(\mathbf{x}) &= 0, \quad \forall \mathbf{x} \in \Gamma.\end{aligned}\tag{4.6}$$

### 4.1.3 The Deformation Method: Version Three

The third version of GDM has real-time adaptation and is convenient in solving problems involving moving boundaries such as motion tracking and computational fluid dynamics (CFD). It is required that the monitor function be normalized and that the following condition holds:

$$\int_{\Omega} \frac{1}{f(t, \mathbf{x})} d\mathbf{x} = |\Omega(t=0)|,\tag{4.7}$$

where  $|\Omega(t=0)|$  denotes the area or volume of the domain  $\Omega$  at time  $t=0$  [31]. Now as with the previous versions the required mapping or transformation from  $\Omega(t=0)$  to  $\Omega(t)$  should satisfy the following relation concerning the Jacobian  $J(\phi(t, \mathbf{x}))$ :

$$J(\phi(t, \mathbf{x})) = \det \nabla \phi(t, \mathbf{x}) = f(t, \phi(t, \mathbf{x})) > 0, \quad \forall \mathbf{x} \in \Omega.\tag{4.8}$$

The required transformation  $\phi$  is then given by the following ordinary differential equation:

$$\begin{aligned}\frac{\partial}{\partial t}\phi(t, \mathbf{x}) &= \mathbf{h}(t, \phi(t, \mathbf{x})) & 0 < t \leq 1, \\ \mathbf{h}(t, \phi(t, \mathbf{x})) &= f(t, \phi(t, \mathbf{x}))\mathbf{u}(t, \phi(t, \mathbf{x})) & 0 < t \leq 1,\end{aligned}\tag{4.9}$$

where  $\mathbf{u}(t, \mathbf{x})$  satisfies the linear partial differential equation

$$\begin{aligned}\nabla \cdot \mathbf{u}(t, \mathbf{x}) &= -\frac{\partial}{\partial t} \frac{1}{f(t, \mathbf{x})} \quad \forall \mathbf{x} \in \Omega, \\ \nabla \times \mathbf{u}(t, \mathbf{x}) &= 0, \quad \forall \mathbf{x} \in \Omega, \\ \mathbf{n} \cdot \mathbf{u}(t, \mathbf{x}) &= 0 \quad \forall \mathbf{x} \in \Gamma.\end{aligned}\tag{4.10}$$

All details together with proofs arguing for the validity of these three versions in providing the required transformations can be found in [60].

## 4.2 Optimal Control Formulation

In this section we present the optimal control formulation of the deformable image registration problem based on the first version of GDM. The resulting optimality system is also presented as derived in [9].

### 4.2.1 Problem Statement

An optimal control formulation of the image registration problem based on GDM seeks controls  $f(\mathbf{x})$  and  $g(\mathbf{x})$  and states  $\phi(t, \mathbf{x})$  and  $\mathbf{u}(\mathbf{x})$  that minimize

$$\begin{aligned}J(\phi(1, \mathbf{x}), f(\mathbf{x}), g(\mathbf{x})) &= \frac{1}{2} \|T(\phi(1, \mathbf{x})) - R(\mathbf{x})\|_{L^2}^2 \\ &\quad + \frac{\alpha}{2} \|f(\mathbf{x})\|_{H^1}^2 + \frac{\beta}{2} \|g(\mathbf{x})\|_{H^1}^2,\end{aligned}\tag{4.11}$$

subject to

$$\begin{aligned}\nabla \cdot \mathbf{u}(\mathbf{x}) &= f(\mathbf{x}) - 1, \quad \forall \mathbf{x} \in \Omega, \\ \nabla \times \mathbf{u}(\mathbf{x}) &= g(\mathbf{x}), \quad \forall \mathbf{x} \in \Omega, \\ \mathbf{n} \cdot \mathbf{u}(\mathbf{x}) &= 0, \quad \forall \mathbf{x} \in \Gamma, \\ \frac{\partial \phi(t, \mathbf{x})}{\partial t} &= \mathbf{u}(t, \phi(t, \mathbf{x})), \quad \forall t, \mathbf{x} \in (0, 1] \times \Omega, \\ \phi(0, \mathbf{x}) &= \mathbf{x}, \quad \forall \mathbf{x} \in \Omega, \\ \int_{\Omega} (f(\mathbf{x}) - 1) d\mathbf{x} &= 0, \quad \forall \mathbf{x} \in \Omega,\end{aligned}\tag{4.12}$$

where  $\alpha$  and  $\beta$  are penalty parameters. It is also necessary that  $\phi(t, \mathbf{x}) \in (0, 1] \times \Gamma$ ,  $\forall \mathbf{x} \in \Gamma$ . The original aim of image registration is to minimize the similarity measure (objective functional) term given by the  $L^2$ -norm in Equation (4.11) above. However, minimizing only that term would result in unbounded optimal control functions because of the lack of any explicit dependence of the objective functional on the controls [8]. It is therefore necessary to penalize the objective functional with additional terms, such as the  $H^1$ -norms (Sobolev norms) of the controls  $f$  and  $g$ . With this choice of a stronger norm than the  $L^2$ -norm for the controls it has been possible to prove the existence of optimal solutions in [8], something which the  $L^2$ -norm penalization may not be sufficient to guarantee.

The monitor function  $f(\mathbf{x})$  controls the local volume changes through the divergence of the transformation in the grid, while  $g(\mathbf{x})$  controls the local rotations through the curl vector of the transformation [1]. The monitor function is an important parameter for achieving proper grid refinement in this method [59]. From the nonlinear ordinary differential Equation (4.2) the following assumption has been made:

$$\mathbf{h}(t, \mathbf{x}) \approx \mathbf{u}(t, \mathbf{x}), \quad (4.13)$$

and the soundness of this assumption has been argued in [61]. One way of approaching the minimization problem above is by applying the Lagrange multiplier method to get the optimality system as in [1, 8, 9] which consists of the state and co-state (adjoint) systems as well as the optimality conditions.

## 4.2.2 The Optimality System

The optimality system is given by the following set of equations derived in [9]:

- State Equations:

$$\nabla \cdot \mathbf{u}(\mathbf{x}) = f(\mathbf{x}) - 1, \quad \nabla \times \mathbf{u}(\mathbf{x}) = g(\mathbf{x}). \quad (4.14)$$

$$\frac{\partial \phi(t, \mathbf{x})}{\partial t} = \mathbf{u}(t, \phi(t, \mathbf{x})), \quad \phi(t_0, \mathbf{x}) = \mathbf{x}. \quad (4.15)$$

- Co-state Equations:

$$\begin{aligned}\nabla \cdot (\xi, -\eta) &= (T(\phi(1, \mathbf{x})) - R(\mathbf{x})) \frac{\partial T(\phi(1, \mathbf{x}))}{\partial \phi_1} = G_1, \\ \nabla \cdot (\eta, \xi) &= (T(\phi(1, \mathbf{x})) - R(\mathbf{x})) \frac{\partial T(\phi(1, \mathbf{x}))}{\partial \phi_2} = G_2,\end{aligned}\quad (4.16)$$

with  $(\xi, \eta)$  being the Lagrange multipliers and  $\phi_i$  the  $i^{\text{th}}$  component of the grid coordinates. The optimality conditions are

$$\alpha f = \xi, \beta g = \eta. \quad (4.17)$$

In [8] Lee and Gunzburger made a suggestion that since the resulting system is constituted of several coupled systems, it is more efficient to solve the system using iterative optimization rather than solve the system all at once. The optimization algorithm adopted in our work is provided in [62]. The ordinary differential Equation (4.15) is solved using Runge-Kutta (RK-4) and all the source codes used in our work can be obtained from the attached CD ROM. In the next section we discuss two approaches to obtaining the optimal solution to the system above.

Given the optimality system the steps for registering the template image to the reference image can be summarized as follows:

---

**Algorithm 4.1** Steps for registering the template image to the reference image.

---

- **Initialization:**

- (a) Initialize controls  $f(\mathbf{x})$ ,  $g(\mathbf{x})$  and penalty parameters  $\alpha$  and  $\beta$ .
- (b) Solve for the displacement field  $\mathbf{u}(\mathbf{x})$  from Equation (4.14).
- (c) Solve for the deformation  $\phi(1, \mathbf{x})$  from Equation (4.15) and evaluate Equation (4.11).

- **Main Loop:**

- (d) Solve for the Lagrange multipliers  $\xi$  and  $\eta$  using Equation (4.16).
- (e) Solve for the controls  $f(\mathbf{x})$  and  $g(\mathbf{x})$  using Equation (4.17).
- (f) Solve for the displacement field  $\mathbf{u}(\mathbf{x})$  from Equation (4.14).
- (g) Solve for the deformation  $\phi(1, \mathbf{x})$  from Equation (4.15) and evaluate Equation (4.11).
- (h) If the stopping criteria not met, go to (d) else submit the current solution  $\phi(1, \mathbf{x})$ ,  $f(\mathbf{x})$  and  $g(\mathbf{x})$ .

- **End Loop**

---

### 4.3 Solution to the Optimality System

One approach taken for solving the partial differential equations (PDEs) in the optimality system is the least squares finite element method (LSFEM) as suggested in [8, 62] and implemented in [52]. The LSFEM works with the system of first order PDEs. Another approach suggested in [9] and presented in [1] is to restructure the first order PDEs into a system of second-order Poisson equations which are easily solvable by the Galerkin finite element method (GFEM) and the finite difference method (FDM).

#### 4.3.1 The System of Poisson Equations

Representing the optimality system as a set of Poisson equations is motivated by the ease of solving Poisson equations and the availability of standard Poisson solvers. This approach was suggested in [9] and implemented in [1] for a 3D image registration problem. It proceeds by taking the gradient of the the div-curl system in the state equation and reorders the components, thus obtaining the following first system of Poisson equations:

$$\Delta \mathbf{u}_x = \frac{\partial f}{\partial x} - \frac{\partial g}{\partial y}, \quad \Delta \mathbf{u}_y = \frac{\partial f}{\partial y} + \frac{\partial g}{\partial x}. \quad (4.18)$$

Using the same logic on the co-state equations yields the following Poisson equations for the Lagrange multipliers  $(\xi, \eta)$ :

$$\Delta \xi = \frac{\partial G_1}{\partial x} + \frac{\partial G_2}{\partial y}, \quad \Delta \eta = \frac{\partial G_2}{\partial x} - \frac{\partial G_1}{\partial y}. \quad (4.19)$$

Substituting the optimality conditions (4.17) in the Equation (4.19) above we get

$$\Delta f = \frac{1}{\alpha} \left( \frac{\partial G_1}{\partial x} + \frac{\partial G_2}{\partial y} \right), \quad \Delta g = \frac{1}{\beta} \left( \frac{\partial G_2}{\partial x} - \frac{\partial G_1}{\partial y} \right). \quad (4.20)$$

The Poisson equations in Equations (4.18) and (4.20) can be reduced into a weak formulation with the help of a square integrable test function  $v$  to make them compatible with the Galerkin finite element method [51]. As shown in [51], given a Poisson equation

$-\Delta w = h$  the weak formulation is given as

$$-\int_{\Omega} v \Delta w = \int_{\Omega} \nabla w \cdot \nabla v - \int_{\Gamma} v \frac{\partial w}{\partial n} = \int_{\Omega} v h, \quad (4.21)$$

where the term  $g_n = v \frac{\partial w}{\partial n}$  describes the boundary conditions which can either be essential (Dirichlet) specified on  $v$  or natural (Neumann) specified on  $\frac{\partial w}{\partial n}$ . The FEM solution approximation  $w_h^e$  in the element grid  $e$  is given in terms of the coefficients  $\mathbf{w}_j^e$  and the support or shape functions  $\psi_j^e$  as:

$$w_h^e = \sum_{j=1}^4 \mathbf{w}_j^e \psi_j^e, \quad \text{such that } \sum_{j=1}^4 \psi_j^e = 1. \quad (4.22)$$

The shape functions  $\psi_j$  define the set of test functions  $v$  and given that, Equation (4.22) can be used in Equation (4.21) to yield the following representation:

$$\sum_{j=1}^4 \mathbf{w}_j^e \int_{\Omega} \nabla \psi_j^e \cdot \nabla \psi_i^e = \int_{\Omega} \psi_i^e h + \int_{\Gamma} \psi_i^e g_n. \quad (4.23)$$

This can be written compactly in matrix form as:

$$\mathbf{K}_e \mathbf{w}^e = \mathbf{H}_e, \quad (4.24)$$

such that the local stiffness matrix  $\mathbf{K}_e$  and the load vector  $\mathbf{H}_e$  are given by:

$$\begin{aligned} \mathbf{K}_e &= [k_{ij}] = \int_{\Omega_e} \nabla \psi_j^e \cdot \nabla \psi_i^e, \\ \mathbf{H}_e &= [h_{ij}] = \int_{\Omega_e} \psi_i^e h + \int_{\Gamma} \psi_i^e g_n. \end{aligned} \quad (4.25)$$

The system of linear equations (4.24) above is called the *Galerkin system*. To obtain the Galerkin solution  $\{w_h\}$  for all the grid elements the local equations are assembled and result in the global system of linear equations

$$[\mathbf{K}_e] \{\mathbf{w}^e\} = [\mathbf{H}_e], \quad (4.26)$$

which can be solved using linear algebra techniques to obtain the coefficients  $\{\mathbf{w}^e\}$ . The

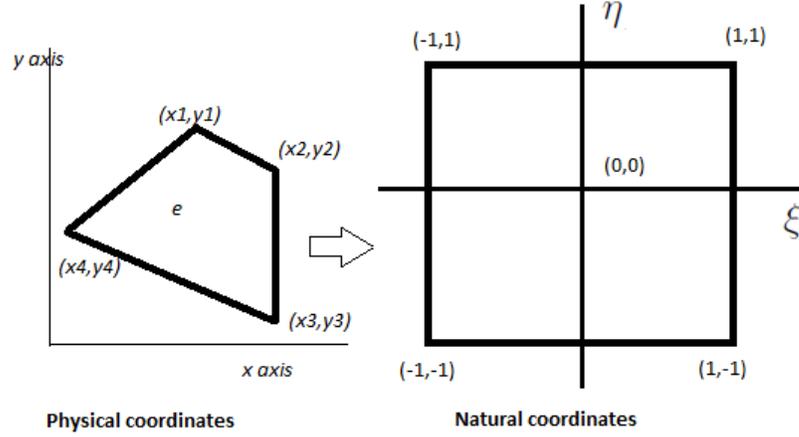


Fig. 4.1: Mapping an arbitrary quadrilateral element in the physical space onto the regular quadrilateral in the natural space to simplify integration.

Galerkin solution is then obtained using Equation (4.22). In our case the Gauss-Legendre Quadrature scheme is used for evaluating integrations in Equation (4.25). Many details are left out here in deriving the Galerkin system and more information can be found in [51] and any introductory book on finite element methods. The shape functions used here are those of quadrilateral grids shown below as given in [51]:

$$\begin{aligned}\psi_1 &= \frac{1}{4}(1 - \xi)(1 - \eta), & \psi_3 &= \frac{1}{4}(1 + \xi)(1 + \eta), \\ \psi_2 &= \frac{1}{4}(1 + \xi)(1 - \eta), & \psi_4 &= \frac{1}{4}(1 - \xi)(1 + \eta).\end{aligned}\quad (4.27)$$

These shape functions can also be thought of as interpolation functions since, given the solution at only four nodes of an element, they can be used to approximate the solution elsewhere within the element through interpolation. Only one shape function is nonzero at any particular node. Any quadrilateral element in the physical space ( $xy$ -coordinate space) with arbitrary shape can be mapped onto the regular quadrilateral in the natural space ( $\xi\eta$ -coordinate space) as shown in Fig. 4.1. This makes integration within the element of any quadrilateral element easy since we only have to perform integration in the natural space with the regular and symmetric quadrilateral.

For validation purposes several Poisson equations whose exact solutions are known were compared with our implementation results. One such Poisson equation and its exact solution is  $-\Delta u(x, y) = 2\pi^2 \sin(\pi x) \sin(\pi y)$  on the domain  $\Omega = [0, 1]^2$  subjected to the Dirichlet boundary conditions  $u(x, y) = 0, \forall (x, y) \in \Gamma$ . The exact solution of this Poisson equation is  $u(x, y) = \sin(\pi x) \sin(\pi y)$ .

Lastly, Runge-Kutta 4 is used for solving the ordinary differential Equation (4.15).

Fig. 4.2 shows examples of this optimal control-based model implemented via the Galerkin approach when applied on two CT slices and two synthetic images for registration.

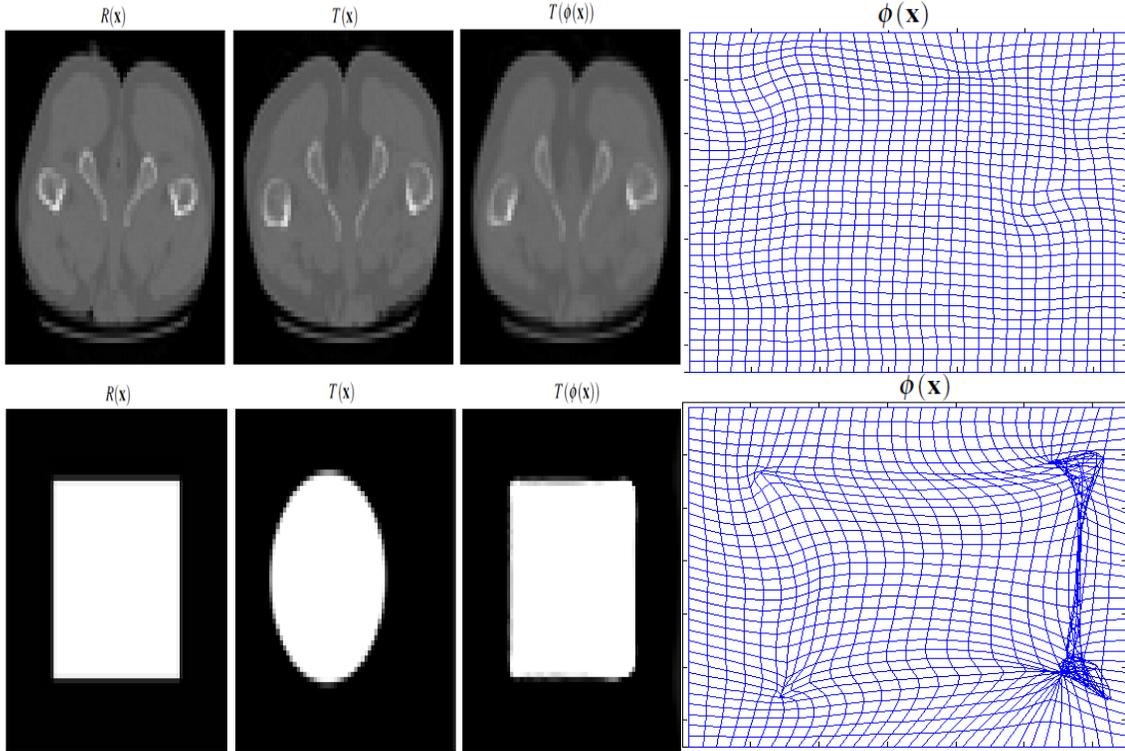


Fig. 4.2: Registration of two synthetic image using the optimal control-based model implemented with GFEM. The first column shows the reference images  $R(\mathbf{x})$ , the second column shows the template images  $T(\mathbf{x})$ , the third column shows transformed or registered template image  $T(\phi(\mathbf{x}))$  and the last column shows the resulting deformation  $\phi(\mathbf{x})$ .

This implementation reduced the normalized dissimilarity measure  $\frac{\|T(\phi(\mathbf{x})) - R(\mathbf{x})\|_{L^2}^2}{\|T(\mathbf{x}) - R(\mathbf{x})\|_{L^2}^2}$  from 1 to 0.12447 for CT slices and from 1 to 0.01103 for synthetic images with  $\alpha = \beta = 0.0005$  in both CT slices and synthetic images registration. The resulting deformation for the CT slices example is visually acceptable, but the one for synthetic images gave rise to grid folding which is not desirable. No pre-registration was used for initialization in these two examples.

### 4.3.2 The System of Linear Equations

Since the optimality system is already constituted by linear equations, one approach is to use LSFEM to obtain the Lagrange multipliers  $(\xi, \eta)$  and the displacement field  $\mathbf{u}(\mathbf{x})$ .

The LSFEM is based on the minimization of the residual in a least-squares sense [52]. For instance the first order equation  $\mathbf{A}\mathbf{w} = \mathbf{h}$  with  $\mathbf{A}$  as the differential operator is restated as the residual in the following particular form as shown in [52]:

$$I(\mathbf{w}) = \|\mathbf{A}\mathbf{w} - \mathbf{h}\|^2 = \int_{\Omega} (\mathbf{A}\mathbf{w} - \mathbf{h})^2 d\mathbf{x}. \quad (4.28)$$

The first order variation of Equation (4.28) results in the following equation which shares some similarities with the weak formulation in Equation (4.21):

$$\int_{\Omega} (\mathbf{A}\mathbf{w}) \cdot (\mathbf{A}\mathbf{v}) d\mathbf{x} = \int_{\Omega} \mathbf{h} \cdot (\mathbf{A}\mathbf{v}). \quad (4.29)$$

The major difference between the Galerkin method and the LSFEM rests upon the family of functions chosen as the test functions in each of them. The Galerkin method chooses the test functions  $v$  from the same class of functions as the approximating or trial functions  $w$ , while the LSFEM chooses the test functions  $\mathbf{A}\mathbf{v}$  from the class of derivatives of the approximating functions  $\mathbf{A}\mathbf{w}$ . In [63] the two approaches are compared on solving the first order differential equation and the Galerkin method was shown to fail in getting a solution due to the strict choice of test functions it allows. Using the FEM solution approximation as given in Equation (4.22) and as shown in [31, 52] the resulting local stiffness matrix  $\mathbf{K}_e$  and load vector  $\mathbf{H}_e$  are given by

$$\begin{aligned} \mathbf{K}_e &= \int_{\Omega_e} \begin{bmatrix} \mathbf{A}\psi_1 & \dots & \mathbf{A}\psi_4 \end{bmatrix}^T \begin{bmatrix} \mathbf{A}\psi_1 & \dots & \mathbf{A}\psi_4 \end{bmatrix} d\mathbf{x}, \\ \mathbf{H}_e &= \int_{\Omega_e} \begin{bmatrix} \mathbf{A}\psi_1 & \dots & \mathbf{A}\psi_4 \end{bmatrix}^T \mathbf{h} d\mathbf{x}. \end{aligned} \quad (4.30)$$

Similar to the Galerkin solution, the LSFEM solution is obtained by assembling the elemental linear equations  $\mathbf{K}_e \mathbf{w}^e = \mathbf{H}_e$  to obtain the global equation  $[\mathbf{K}_e] \{\mathbf{w}^e\} = [\mathbf{H}_e]$ . One advantage of the LSFEM is that the stiffness matrix  $\mathbf{K}_e$  is always symmetric and positive-definite [52]. It is tricky to perform integration over the original quadrilateral elements, which keep on deforming from time to time, so a reference quadrilateral which is regular and symmetric (as the one shown in Fig. 4.1) is used for integration. This approach requires the Jacobian matrix  $J(\xi, \eta)$  which maps every arbitrary element in the  $xy$ -coordinate space to the regular element in the  $\xi\eta$ -coordinate space. This required

Jacobian matrix is shown below as given in [51]:

$$J(\xi, \eta) = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix} \quad (4.31)$$

such that

$$\begin{aligned} \frac{\partial x}{\partial \xi} &= \frac{1}{4}(1 - \eta)(x_2 - x_1) + (1 + \eta)(x_3 - x_4) \\ \frac{\partial y}{\partial \xi} &= \frac{1}{4}(1 - \eta)(y_2 - y_1) + (1 + \eta)(y_3 - y_4) \\ \frac{\partial x}{\partial \eta} &= \frac{1}{4}(1 - \xi)(x_4 - x_1) + (1 + \xi)(x_3 - x_2) \\ \frac{\partial y}{\partial \eta} &= \frac{1}{4}(1 - \xi)(y_4 - y_1) + (1 + \xi)(y_3 - y_2) \end{aligned} \quad (4.32)$$

where  $(x_i, y_i)$  is the coordinate at a particular node in the  $xy$ -coordinate space. The integrations in Equation (4.30) can be represented in the  $\xi\eta$ -coordinate space by substituting the following relations obtained from [52] into Equation (4.30):

$$\begin{aligned} d\mathbf{x} &= |\det J(\xi, \eta)| d\xi d\eta, \\ x &= \sum_{j=1}^4 x_j \psi_j(\xi, \eta), \\ y &= \sum_{j=1}^4 y_j \psi_j(\xi, \eta), \end{aligned}$$

$$\begin{bmatrix} \frac{\partial \psi_j}{\partial x} \\ \frac{\partial \psi_j}{\partial y} \end{bmatrix} = J(\xi, \eta)^{-1} \begin{bmatrix} \frac{\partial \psi_j}{\partial \xi} \\ \frac{\partial \psi_j}{\partial \eta} \end{bmatrix}. \quad (4.33)$$

More details on the LSFEM can be found in [31, 52, 64]. The first order systems of PDEs presented in [62] as the least square formulation of the optimal control problem in [8] were solved using the LSFEM approach described briefly above, and the convergence rate was relatively slow compared with the Galerkin approach in the previous subsection. This is most probably because of the relatively large matrix equations that need to be solved. For instance, the  $\mathbf{A}\psi_j$  obtained from Equation (3.47) in [62] for the control  $f$  had the following

form:

$$\mathbf{A}\psi_j = \begin{bmatrix} \psi_j & 0 & -\frac{\partial\psi_j}{\partial x} \\ 0 & \psi_j & -\frac{\partial\psi_j}{\partial y} \\ -\frac{1}{\beta}\frac{\partial\psi_j}{\partial x} & -\frac{1}{\beta}\frac{\partial\psi_j}{\partial y} & \beta\psi_j \end{bmatrix}, \quad (4.34)$$

and upon substituting Equation (4.34) into Equation (4.30) the resulting matrices are relatively large compared with the ones obtained in the Galerkin formulation. Thus the following approach was considered and implemented: the div-curl system given in the optimality system can be written as follows:

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{\partial\mathbf{u}_x}{\partial x} \\ \frac{\partial\mathbf{u}_y}{\partial x} \end{bmatrix} + \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} \frac{\partial\mathbf{u}_x}{\partial y} \\ \frac{\partial\mathbf{u}_y}{\partial x} \end{bmatrix} = \begin{bmatrix} f-1 \\ g \end{bmatrix} \quad (4.35)$$

and with the help of Equation (4.22) the resulting  $\mathbf{A}\psi_j$  is given as

$$\mathbf{A}\psi_j = \begin{bmatrix} \frac{\partial\psi_j}{\partial x} & \frac{\partial\psi_j}{\partial y} \\ -\frac{\partial\psi_j}{\partial y} & \frac{\partial\psi_j}{\partial x} \end{bmatrix}. \quad (4.36)$$

Also using a similar approach the co-state equations result in the local matrix equation

$$\begin{bmatrix} \frac{\partial\psi_j}{\partial x} & \frac{\partial\psi_j}{\partial y} \\ -\frac{\partial\psi_j}{\partial y} & \frac{\partial\psi_j}{\partial x} \end{bmatrix} \begin{bmatrix} \eta_j \\ \xi_j \end{bmatrix} = - \begin{bmatrix} G_{2j} \\ G_{1j} \end{bmatrix}, \quad (4.37)$$

and as it appears, the  $\mathbf{A}\psi_j$  matrix in Equation (4.37) is exactly the same as that found for the div-curl system. This means that one implementation is enough to solve both the state and the co-state equations, and the optimality conditions (4.17) are then used to obtain the controls from the Lagrange multipliers. This approach was implemented and the optimization algorithm outlined in [62] was adopted. Fig. 4.3 shows examples of this approach when applied on two CT slices and two synthetic images for registration. Even though this approach is faster than the one in [62] we use the Galerkin based approach in the evaluation instead.

This implementation approach was able to reduce the normalized dissimilarity measure  $\frac{\|T(\phi(\mathbf{x}))-R(\mathbf{x})\|_{L^2}^2}{\|T(\mathbf{x})-R(\mathbf{x})\|_{L^2}^2}$  from 1 to 0.14062 for CT slices and from 1 to 0.017795 for synthetic images with  $\alpha = \beta = 0.0005$  in both CT slices and synthetic images registration. The resulting deformation for the CT slices example is visually acceptable however the one for synthetic images is not as good. No pre-registration was used for initialization in these two examples.

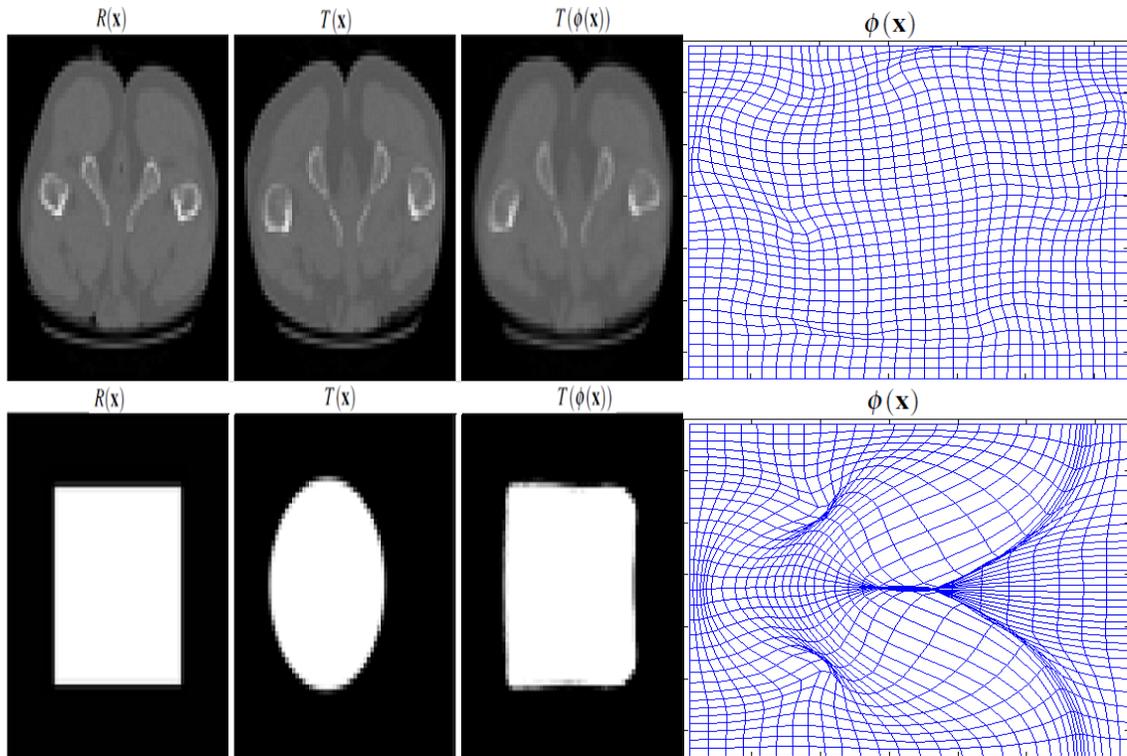


Fig. 4.3: Registration of two synthetic image using the Optimal Control-based model implemented with LSFEM. The first column shows the reference images  $R(\mathbf{x})$ , the second column shows the template images  $T(\mathbf{x})$ , the third column shows transformed or registered template image  $T(\phi(\mathbf{x}))$  and the last column shows the resulting deformation  $\phi(\mathbf{x})$ .

# Chapter 5

## Evaluation in Fitness Space

There are different methods used for evaluating the performance of algorithms or models, but broadly speaking the evaluation methods are divided into two categories: analytical methods and empirical methods [65]. Analytical methods directly examine the algorithm through analysis of their properties and principles. For instance a long and computationally intensive algorithm is necessarily slower than a short and simple one. These methods avoid direct implementation of algorithms to be evaluated, and thus they do not suffer from influences caused by the arrangement of evaluation experiments. However, it is difficult to compare algorithms solely by analytical studies. Hence analytical methods have not received much attention [65].

On the other hand, empirical methods work by testing algorithms on a particular problem and measuring the quality of the results. These methods require implementation of the algorithms to be evaluated and availability of test problems. The empirical methods use some goodness measure or performance index which gives a measure of how good the algorithm is or how well the model performs. The performance index can be a variable in the model or a function chosen such that it varies monotonically with the performance of the algorithm [66]. The empirical methods evaluate and rank algorithms through computation of some chosen performance index without the need for a priori knowledge of the reference. These methods have been widely used in the evaluation of image segmentation algorithms [65, 66, 67].

In [68] four deformable image registration models namely, B-spline, free-form deformation (FFD), Horn-Schunk optical flow (OF) and Demons were evaluated on thoracic images. The registration error for the evaluation was defined as the difference between the

manually measured displacement vector field (mDVF) obtained from the coordinate list of anatomical landmarks and the automatically calculated displacement vector field (aDVF) obtained using each of the four models. The mean 3D registration errors were  $2.7 \pm 0.8mm$  for B-spline,  $3.6 \pm 1.0mm$  for FFD,  $2.4 \pm 0.9mm$  for OF and  $2.4 \pm 1.2mm$  for Demons. It was concluded that B-spline, OF and Demons showed reasonable accuracy, and thus have potential to be used for 4D dose calculation and 4D CT ventilation imaging in patients with thoracic cancer. It was also suggested that the accuracy for all the models could be improved by using optimized parameter tuning.

In [69] an entirely computer-driven systematic method, which eliminates biased interpretations associated with human evaluation, was proposed and used to quantify the ability of a modified basis spline (B-Spline) registration algorithm to recover artificially introduced deformations. The method is fully automated and it can be applied to any chosen registration algorithm without the requirement of user interaction. With this method the evaluated deformable image registration model was shown to be accurate and stable in the presence of noise in the images used for evaluation. In [70] the Demons and the thin-plate spline interpolation were evaluated using a concept based on unbalanced energy. The evaluation was performed on a prostate patient's CT images. The detection of unbalanced energy together with a conditioned finite element model gives a quantitative assessment of every registration instance.

In [71] evaluation of the performance of surface-based deformable image registration for adaptive radiotherapy of non-small cell lung cancer was carried out both visually and quantitatively. Their method requires manual segmentation of CT images prior to comparisons. Several other methods of evaluation are mentioned in [72] together with the registration errors obtained from the deformable image registration models used in evaluation. Wei in [73] uses average relative overlap, normalized region of interest overlap, intensity variance, inverse consistency error as well as transitivity error to evaluate deformable image registration models.

Most of these evaluation methods mentioned focus on the registration error or accuracy and have little to say of the resulting nature of the deformation. Sometimes the overall performance of an algorithm can be defined by at least two performance indices. The evaluation in this case can be thought of as a multiobjective optimization problem whereby multiple objectives are given by these multiple performance indices. One approach consists of forming a weighted linear aggregation of the performance indices which results in one composite performance index, minimization of which reflects just one solution point in space. The weights can be changed to vary the relative importance and significance

of one performance index over the other. A disadvantage of this approach is that it lacks objectivity since the weights are set subjectively by a designer.

One way to avoid subjectivity is by treating each performance index as a dimension in a fitness space rather than aggregating them. In this way all performance indices are orthogonal to each other and the performance of an algorithm is marked by a point in this fitness space. It is clear that the coordinate points which are closest to the ideal performance point in this fitness space, according to a certain distance measure, mark the best performing algorithms. This is the basis for constructing what is called a Pareto front. This is the approach we adopt when carrying out the evaluation process in this work.

The essence of this chapter is to explain the proposed evaluation procedure, which is based on a Pareto ranking scheme and proposes the fitness or objective functions to be used as performance indices. Since some of the fitness functions are conflicting we adopt the level diagrams analysis as a way of visualizing the Pareto fronts.

## 5.1 Motivation

Motivated by Everingham *et al.* [66] on evaluation of image segmentation using the Pareto front, we adopt a general form of aggregate fitness function as

$$H(a_{\mathbf{p}}, I) = \Phi(h_1(a_{\mathbf{p}}, I), \dots, h_n(a_{\mathbf{p}}, I)), \quad (5.1)$$

where  $a_{\mathbf{p}}$  represents algorithm  $a$  (registration algorithm in our case) with parameters  $\mathbf{p}$ ,  $I$  is a set of images and  $h_i(a_{\mathbf{p}}, I)$  are individual fitness functions defined to increase monotonically with the fitness of some particular aspect of the algorithm's behavior. The behavior of an algorithm  $a$  applied to a set of images  $I$  with a particular choice of parameters  $\mathbf{p}$  can be characterized by a point in the  $n$ -dimensional space defined by evaluation of the fitness functions. Variation of parameters  $\mathbf{p}$  of algorithm  $a$  produces a new point in the fitness space and this makes it possible to compare algorithms having different parameter sets. This representation decouples the fitness trade-off from the particular parameter set used by an individual algorithm.

## 5.2 Pareto Front Approximation

In this section we introduce the concepts of the Pareto front and discuss the approximation of Pareto fronts using genetic algorithms.

### 5.2.1 Pareto Front

Points generated by variations of algorithm parameters are plotted on the fitness space. The set of non-dominated points on the fitness space is given by

$$\{\langle a_{\mathbf{p}} \in P_a, H(a_{\mathbf{p}}, I) \rangle \mid \neg \exists a_{\mathbf{q}} \in P_a : H(a_{\mathbf{q}}, I) > H(a_{\mathbf{p}}, I)\}, \quad (5.2)$$

where  $P_a$  is the parameter space of algorithm  $a$ ,  $H$  is a vector of fitness functions denoted as  $\langle h_1(a_{\mathbf{p}}, I), \dots, h_n(a_{\mathbf{p}}, I) \rangle$  and a partial ordering relation on  $F$  is defined as

$$\begin{aligned} H(a_{\mathbf{q}}, I) > H(a_{\mathbf{p}}, I) &\iff \forall i : h_i(a_{\mathbf{q}}, I) \geq h_i(a_{\mathbf{p}}, I) \\ &\text{and} \\ \exists i : h_i(a_{\mathbf{q}}, I) > h_i(a_{\mathbf{p}}, I). \end{aligned} \quad (5.3)$$

This construction, referred to as the ‘‘Pareto front’’, is extendable to the case of multiple algorithms, and this has the natural consequence that any algorithm which does not contribute to the front can be considered a bad choice for any monotonic fitness function  $\Phi$  [66]. Representation in Equation (5.3) above is also abbreviated as  $\mathbf{p} \prec \mathbf{q}$  which means solution  $\mathbf{p}$  dominates  $\mathbf{q}$ .

### 5.2.2 Approximation by NSGA-II

Genetic algorithms are commonly used to approximate the Pareto front with a finite number of points, and have an advantage in that they require less prior knowledge about the behavior of an algorithm with respect to its parameters [66]. NSGA II is an elitist multi-objective evolutionary algorithm introduced by Deb *et al.* [74] as an improved version of the NSGA. A flow diagram illustrating this algorithm is shown in Fig. 5.1.

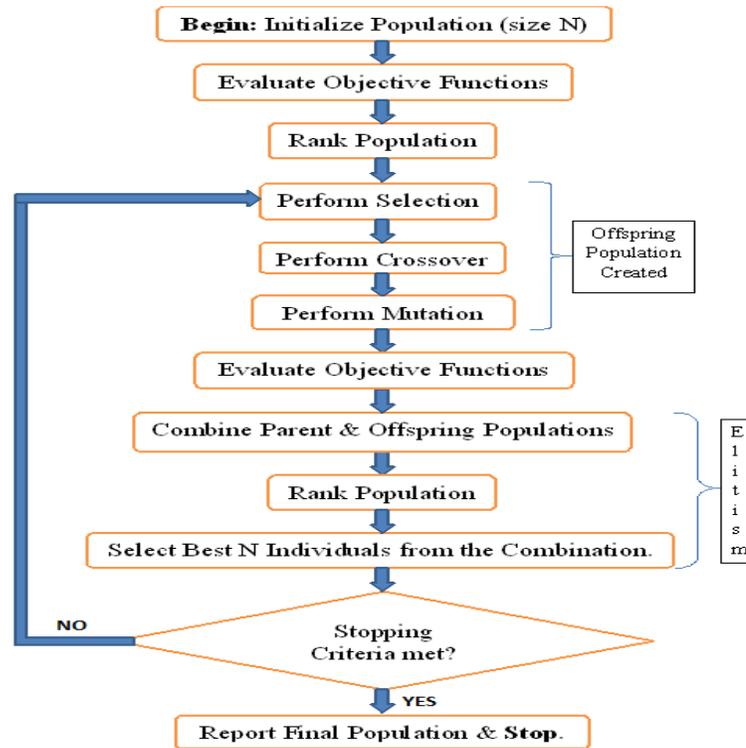


Fig. 5.1: The flow diagram of NSGA-II.

More details about NSGA-II can be found in [74]. A Matlab code by Seshadri [75] was used in this work.

In NSGA-II, for each solution  $\mathbf{p} \in P$  one has to determine the following:

- The set  $S_{\mathbf{p}}$  of solutions  $\mathbf{q}$  dominated by solution  $\mathbf{p}$ . i.e.  $S_{\mathbf{p}} = \{\mathbf{q} : \mathbf{p} \prec \mathbf{q}\}$
- The integer number  $n_{\mathbf{p}}$  of solutions dominating solution  $\mathbf{p}$ .
- The integer number  $p_{rank}$  indicating the *rank* of solution  $\mathbf{p}$ . i.e. the front index.
- *Crowding-distance* of solution  $\mathbf{p}$ , which gives the density of solutions surrounding solution  $\mathbf{p}$  with respect to each objective function.

*Binary tournament selection* is performed using a *crowding-comparison operator* which takes into account the following:

- Non-domination *rank*,  $p_{rank}$  of individual  $\mathbf{p}$ : in the tournament, during selection, non-dominated solutions with lesser *rank* are preferred.

- *Crowding-distance* of solution  $\mathbf{p}$ : if competing solutions in the tournament share the same *rank* then the solution with higher *crowding-distance* is preferred.

The following *genetic operators* are used to produce offspring:

- *Simulated Binary Crossover (SBX)*:

$$\begin{aligned} c_{1,k} &= \frac{1}{2}[(1 - \beta_k)p_{1,k} + (1 + \beta_k)p_{2,k}], \\ c_{2,k} &= \frac{1}{2}[(1 + \beta_k)p_{1,k} + (1 - \beta_k)p_{2,k}], \end{aligned} \quad (5.4)$$

where  $c_{i,k}$  is the  $i^{\text{th}}$  child with  $k^{\text{th}}$  component,  $p_{i,k}$  is the selected parent and  $\beta_k \geq 0$  is a sample from a random number generated having the density:

$$\begin{aligned} p(\beta) &= \frac{1}{2}(\eta_c + 1)\beta^{\eta_c}, \quad 0 \leq \beta \leq 1, \\ p(\beta) &= \frac{1}{2}(\eta_c + 1)\frac{1}{\beta^{\eta_c+2}}, \quad \beta > 1, \end{aligned} \quad (5.5)$$

where  $\eta_c$  is the distribution index for crossover. This distribution can be obtained from a uniformly sampled random number  $u$  between (0,1). That is

$$\begin{aligned} \beta(u) &= (2u)^{\frac{1}{\eta_c+1}}, \\ \beta(u) &= \frac{1}{[2(1-u)]^{\frac{1}{\eta_c+1}}}. \end{aligned} \quad (5.6)$$

- *Polynomial Mutation*:

$$c_k = p_k + (p_k^u - p_k^l)\delta_k, \quad (5.7)$$

where  $c_k$  is the child and  $p_k$  is the parent with  $p_k^u$  and  $p_k^l$  being the upper and lower bounds on the parent component respectively. The parameter  $\delta_k$  represents a small variation which is calculated from a polynomial distribution using

$$\begin{aligned} \delta_k &= (2r_k)^{\frac{1}{\eta_m+1}} - 1, \quad \text{if } r_k < 0.5, \\ \delta_k &= 1 - [2(1 - r_k)]^{\frac{1}{\eta_m+1}}, \quad \text{if } r_k \geq 0.5, \end{aligned} \quad (5.8)$$

where  $r_k$  is a uniformly sampled random number in the open interval (0,1) and  $\eta_m$  is the mutation index.

The offspring population is combined with current generation population and selection is

performed to set the individuals of the next generation. Since all the previous and current best solutions are preserved in the population, elitism is ensured. One example of the Pareto front generated by NSGA-II is shown in Fig. 5.2 with two objective functions  $f1 = 1 + \cos\theta$  and  $f2 = 1 + \sin\theta$ . In the next section we formulate the objective functions necessary for the purpose of the evaluation process in this work.

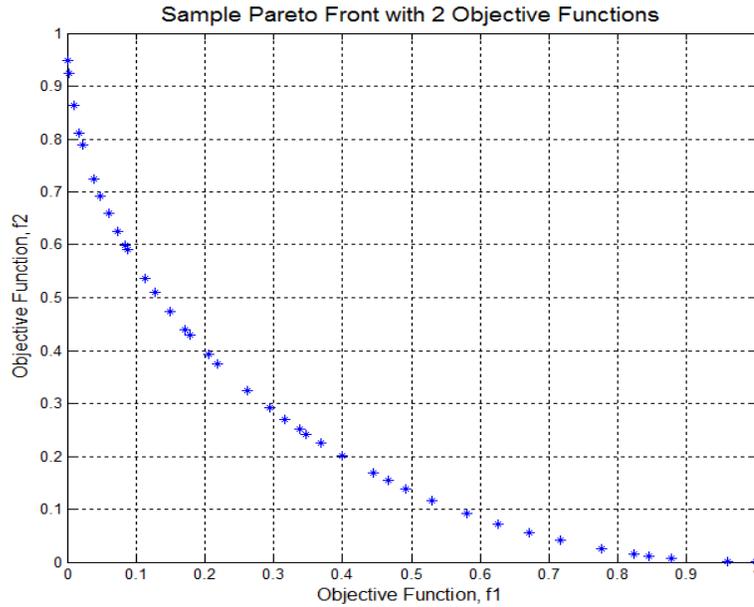


Fig. 5.2: Approximation of Pareto Front generated by NSGA-II.

### 5.3 Fitness (Objective) Functions

Fitness or objective functions are the necessary components in any empirical or quantitative evaluation scheme. They serve as performance indices which indicate the performance of an algorithm or model in question. In the case where there are two or more objective functions in the evaluation process, the evaluation is considered multidimensional where the dimensions of a fitness space are constituted by the objective functions themselves such as with  $f1$  and  $f2$  in Fig. 5.2. For the Pareto front evaluation scheme it is necessary that there be at least two objective functions. Sometimes each objective function is fully decoupled from others and in such cases it is almost trivial to use Pareto fronts as an evaluation tool. It is in cases where at least two objective functions are not only coupled but also conflicting whereby the Pareto front becomes the vital tool for evaluation.

In the sections below we construct two major conflicting objective functions to be used in evaluating the deformable image registration models. The deformation quality measure is

itself composed of three independent quality measures which complement each other to give a better quantitative measure of the deformation quality.

### 5.3.1 Similarity Measure

SSD and CC are the most popular geometry-based similarity measures. Although restricted to monomodal applications, SSD is adopted here due to it being simple, intuitive and computationally efficient. The normalized SSD is much more convenient as a well-bounded objective function  $h_1$ :

$$h_1(a_{\mathbf{p}}, R, T, \phi) = \frac{\|T(\phi(\mathbf{x})) - R(\mathbf{x})\|_{L_2}^2}{\|T(\mathbf{x}) - R(\mathbf{x})\|_{L_2}^2} \quad (5.9)$$

where  $a_{\mathbf{p}}$  stands for a registration algorithm for a particular model and subscript  $\mathbf{p}$  stands for parameters of the model. These can be varied as part of the decision variables to investigate optimal tuning of the model in algorithm  $a$ . For variational-based models we have  $\phi(\mathbf{x}) = \mathbf{x} - \mathbf{u}(\mathbf{x})$  and for the optimal control-based model we have  $\frac{\partial}{\partial t}\phi(t, \mathbf{x}) = \mathbf{u}(t, \phi(t, \mathbf{x}))$ .

### 5.3.2 Deformation Quality Measure

The quality of a grid is quantified by *skewness*, *shape*, *size*, *aspect ratio* and *orientation* [76, 77], and a proposed quality measure should account for most if not all of these quantities. However, skewness varies almost linearly with shape. Also, aspect ratio is meaningful only if skew is significant. Shape is preferred because it contains both skew and aspect ratio [78]. In this work we adopt two of these measures as formulated for quadrilateral grid elements in [78]. Fig. 5.3 shows example of a structured or regular grid and an unstructured or irregular grid, each with four quadrilateral elements labeled  $e_1, e_2, \dots, e_4$  where each element has four nodes labeled  $k, k+1, \dots, k+3$ .

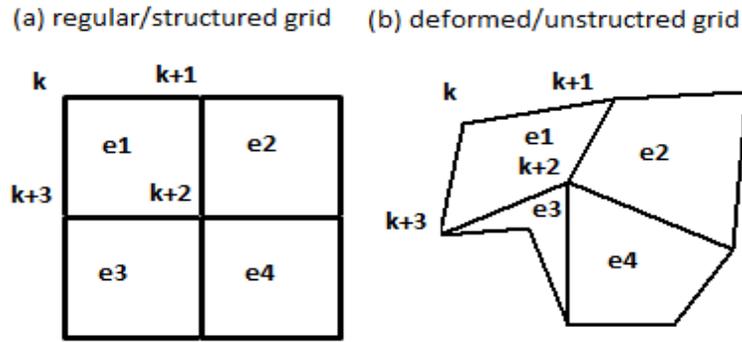


Fig. 5.3: A regular grid with quadrilateral elements (left) and its deformed version (right).

Unlike in the case of simplicial elements such as the triangle and tetrahedron, which possess very convenient properties simplifying the definition of quality measures, the quadrilateral element does not. The main problem is that the Jacobian matrix of a non-simplicial element like the quadrilateral cannot be defined in terms of a single Jacobian matrix [78]. There are actually four Jacobian matrices for the quadrilateral element, one for each node  $k$ . However, it has been shown in [78] that given any node  $k$ , the sum of the determinants of the Jacobian matrices at nodes  $k$  and  $k+2$  is always twice the total area of the quadrilateral. This invariance of the element area under node transformation helps us to avoid using four of the Jacobian matrices to construct the quality measure. We proceed to define the Jacobian matrix at node  $k$  of element  $e$  in the grid  $\phi(\mathbf{x})$  as follows:

$$A_k^e = \begin{bmatrix} \phi_{1_{k+1}}^e - \phi_{1_k}^e & \phi_{1_{k+3}}^e - \phi_{1_k}^e \\ \phi_{2_{k+1}}^e - \phi_{2_k}^e & \phi_{2_{k+3}}^e - \phi_{2_k}^e \end{bmatrix}, \quad k \in \{0, 1, 2, 3\}, \quad (5.10)$$

where  $(\phi_{1_k}^e, \phi_{2_k}^e)$  is the coordinate pair at node  $k$  of element  $e$  in the deformed grid  $\phi(\mathbf{x})$ . Thus  $\alpha_k^e = \det(A_k^e) + \det(A_{k+2}^e)$  is twice the total area of the element  $e$ , and using these settings the relative size measure as given in [78] is

$$f_{size}^e = \min\left(\frac{\alpha_k^e}{2w}, \frac{2w}{\alpha_k^e}\right), \quad (5.11)$$

where  $w$  is the area of a reference element. This measure has other essential properties which are:

- $f_{size}^e = 1$  if and only if the physical quadrilateral has the same total area as the reference quadrilateral  $w$ .

- $f_{size}^e = 0$  if and only if the physical quadrilateral has a total area of zero.
- The value of  $f_{size}^e$  depends directly on the area of the physical quadrilateral.

However,  $f_{size}^e$  is just local in the grid and thus cannot give the size quality for the entire grid. We propose solving this problem by introducing the infinity norm taken over the entire grid. The infinity norm is convenient in the sense that it is able to capture and expose the worst element in the whole grid and set it as the quality measure of the entire grid. In short we are adopting the worst case element size as the quality measure for the whole grid with regard to element size quality. The other inconvenience with  $f_{size}^e$  is that it conforms well with the maximization methods for optimization instead of the minimization methods which we require for consistency purposes here. We alleviate this problem by considering  $1 - f_{size}^e$  rather than  $f_{size}^e$  itself. The final proposed relative size quality measure is

$$h_2(a_{\mathbf{p}}, \phi) = \left\| 1 - f_{size}^e \right\|_{\infty}. \quad (5.12)$$

The relative size measure is not a very useful quality measure on its own because a grid element can be poorly shaped yet still maintain a good size measure. Complementary to the relative size quality measure we need the shape quality measure. Unfortunately, describing the shape of the quadrilateral requires not just two as with size but rather all four Jacobian matrices  $A_k^e$  to give the following shape quality measure as given in [78]:

$$f_{shape}^e = \frac{8}{\sum_{k=0}^3 \frac{\text{trace}(A_k^{eT} A_k^e)}{\det(A_k^e)}}. \quad (5.13)$$

This shape measure is nodally invariant and has the following other properties:

- $f_{shape}^e = 1$  if and only if the physical quadrilateral is a square.
- $f_{shape}^e = 0$  if and only if the physical quadrilateral is degenerate or has zero area.
- The value of  $f_{shape}^e$  is scale invariant.

Following a similar procedure as in with the size quality measure above, the shape quality measure above is modified to give the following:

$$h_3(a_{\mathbf{p}}, \phi) = \left\| 1 - f_{shape}^e \right\|_{\infty}. \quad (5.14)$$

Even though  $h_2$  and  $h_3$  are necessary for quantifying the deformation or grid quality, they are not sufficient since they do not give a measure of how many grid elements are degenerate or inverted (deformation failure). For this reason we impose another global quality measure to account for that, and it is a modification of the one given in [76]:

$$h_4(a_{\mathbf{p}}, \phi) = \frac{1}{2N} \sum_{e=1}^N \frac{\{|\alpha_k^e - \rho| - \alpha_k^e + \rho\}}{|\alpha_k^e - \rho| + \varepsilon}, \quad (5.15)$$

where  $N$  is the total number of elements in the grid,  $0 < \varepsilon \leq 1$  is set to avoid division by zero and  $0 < \rho < 1$  is a threshold used to isolate inverted elements with zero area. All objective functions  $h_i$  above are dimensionless and all range from zero to one.

## 5.4 Level Diagrams Analysis of Pareto Fronts

Whenever two or more objectives are conflicting in the Pareto front, visualization and trade-offs become nontrivial and this necessitates tools to tackle the visualization and trade-off issues. Level diagrams are used for visualizing Pareto fronts by providing a geometrical visualization of the Pareto front based on a distance measure from an ideal solution point, which optimizes all objectives simultaneously [79]. Given normalized objectives  $h_i$ , as the ones above, a suitable norm is chosen to evaluate the distance from an ideal solution point to a point on the Pareto front. In this work we adopt the infinity norm for evaluating distance since it offers a compact visualization of the Pareto front which is useful for trade-off analysis [79]. With the distances computed, the plot of level diagrams proceeds as follows.

The distance of all points from the Pareto front are plotted against each objective function  $h_i$  to produce two-dimensional plots with distance along the  $y$ -axis and objective function  $h_i$  along the  $x$ -axis. In this way, a single point from the Pareto front is plotted at one level ( $y$ -axis), carrying with it all the information about its performance on the Pareto front. The smaller the distance measure on the level diagrams the better the performance. Thus a decision maker can use this distance measure to decide which point, corresponding to a particular algorithm, suits their design specifications.

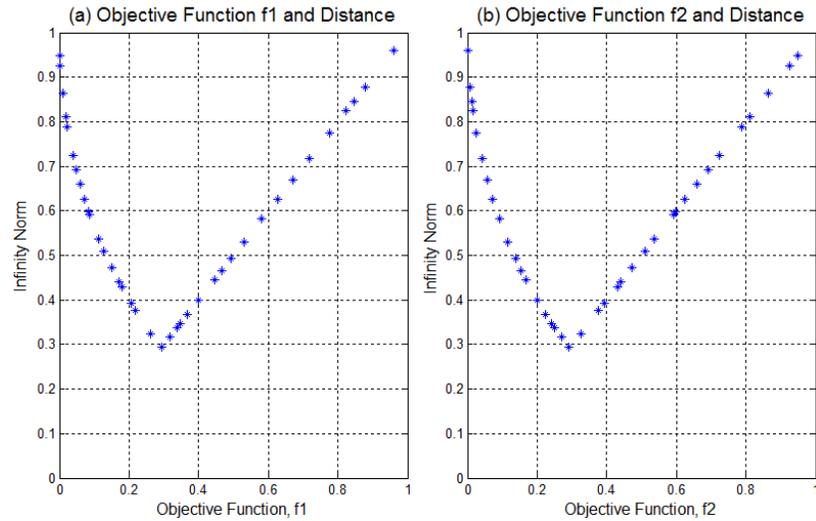


Fig. 5.4: The Level Diagram of the Pareto Front example in Fig. 5.2.

More details about level diagrams can be found in [79, 80]. In Fig. 5.4 we show the level diagram visualization of the example Pareto front in Fig. 5.2. The point with the lowest infinity norm distance is the best point for minimizing all objective functions simultaneously.

# Chapter 6

## Evaluation Results

In radiation therapy the accuracy and precision on targeted unhealthy tissues or organs is of utmost importance because if not, healthy tissues nearby can get damaged by high radiation doses meant only for the tumors. Medical imaging techniques have been very successful in providing this precision in various medical applications. One such application is in the diagnosis, monitoring and treatment of prostate tumor and growth. Prior to treatment phase the patient is first taken through the planning phase and data collected is stored as 3D CT data. During actual treatment, images of the targeted tissues are produced by a real-time digital X-Ray device as *Portal Radiographs* (PRs). For proper correspondence between planning and treatment phases, the 3D CT data is projected from a known viewpoint and angle to produce 2D images called *digitally reconstructed radiographs* (DRRs), which are then compared with PR to check for misalignment. Fig. 6.1 shows the example of a CT slice and a DRR.

As part of the misalignment correction process, a preliminary patient positioning can be carried out based on external markers, either on the patient's skin or using the mask system [16]. However, even after this alignment the organs or tissues inside may still not be aligned properly, especially if the regions of interest are in the abdominal region where organs and tissues are soft and highly deformable. For instance the prostate whose pose and shape depend on the conditions in the bladder and rectum. This problem necessitates the use of deformable image registration techniques to alleviate the remaining misalignment between the PR and the DRR.

In this work we prepare a setup which mimics much of the application explained above and use it as a platform for evaluating the deformable image registration models presented in

Chapters 3 and 4. That is to mimic the registration of DRRs to the PR of a human prostate to obtain the tissue transformation from the planning phase to the treatment phase. The major limitation in our setup is the lack of PR data. Thus we divide DRRs and CT slices into two sets: one acting as DRRs while the other mimicking PRs. This prostate registration problem is used in this work as a test application to facilitate proper evaluation of the deformable registration models. This chapter presents two experiments on which the models are evaluated using the procedure explained in Chapter 5.

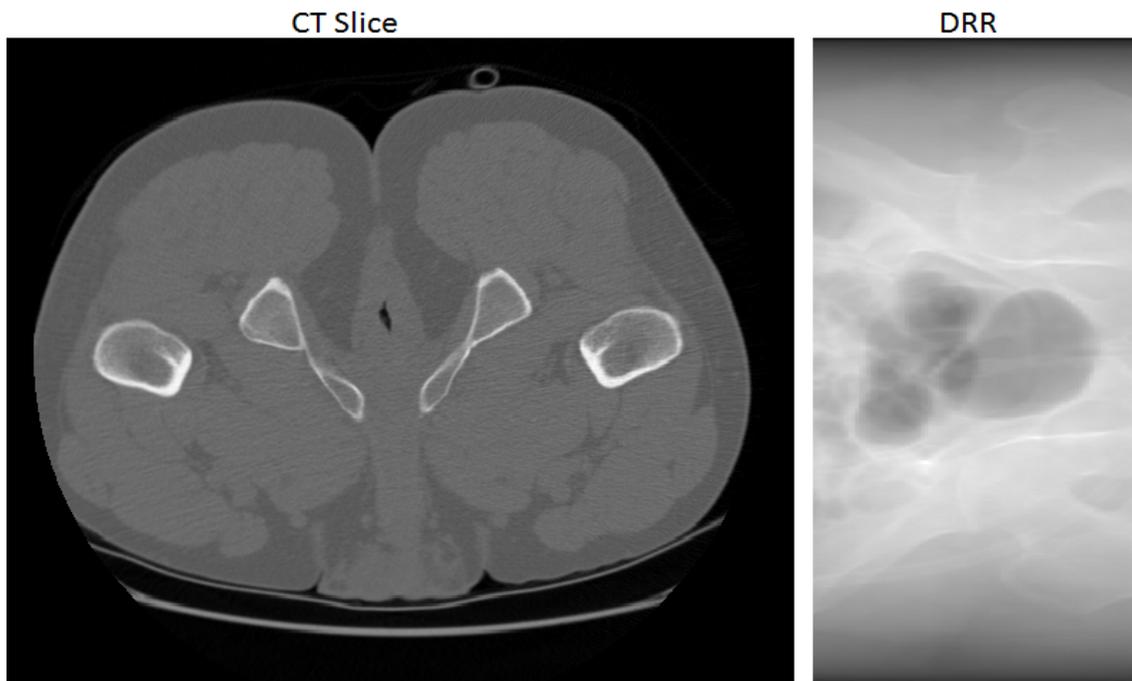


Fig. 6.1: Example of a CT slice (left) and a DRR (right).

## 6.1 Experiment One

The purpose of this first experiment is to compare the performance of the models on prostate registration application based on DRRs formed by projection of CT data along the cross sectional view of the human body. As part of the comparison the model performance with and without the pre-registration is carried out as a way of investigating the relative reliance of the models on pre-registration. A demonstration of how evaluation can aid in decision making is shown through level diagram interpretation.

### 6.1.1 Experimental Setup

In this experiment we used the DRRs from different patients to perform prostate registration. The DRRs were grouped into a template set  $T_s = \{T_i(\mathbf{x}) : i = 1, 2, \dots, 25\}$  and a reference set  $R_s = \{R_i(\mathbf{x}) : i = 1, 2, \dots, 25\}$  (representing PRs). The criteria used when pairing  $T_i(\mathbf{x})$  with  $R_i(\mathbf{x})$  is based on feature similarity and slice level correspondence between the two, but no  $T_i(\mathbf{x})$  was paired with itself.

A copy of  $T_s$  was passed through the pre-registration process and kept as set  $T_p$  of pre-registered templates. The experiment proceeded by testing all models with and without pre-registration separately on the prostate registration problem. For each model the NSGA-II optimization algorithm is used to optimize the initial population to produce the approximate Pareto front. The results are shown in Fig. 6.2 and their interpretation is presented in the sections below together with the Pareto fronts visualization shown in Fig. 6.3. Every parameter (decision variable), for all models, was varied in the open interval  $(0, 1)$  to produce the Pareto fronts. Decision variables  $\lambda$  and  $\mu$  in the elastic and fluid models are bulk and shear modulus measured in Pascal units. Every other decision variable is dimensionless.

### 6.1.2 Results and Interpretations

In this section we compare each variational-based model with the optimal control-based model using, as reference, the Pareto fronts shown in Fig. 6.2. The term *net Pareto front* refers to the non-dominated set of points contributed by the chosen Pareto fronts in Fig. 6.2. As a summary of the Pareto fronts shown in Fig. 6.2 (d), (e) and (f), and for decision making purposes, level diagrams representation of the Pareto fronts is shown in Fig. 6.3. Since all objective functions are dimensionless, no units are put in Fig. 6.2 and Fig. 6.3.

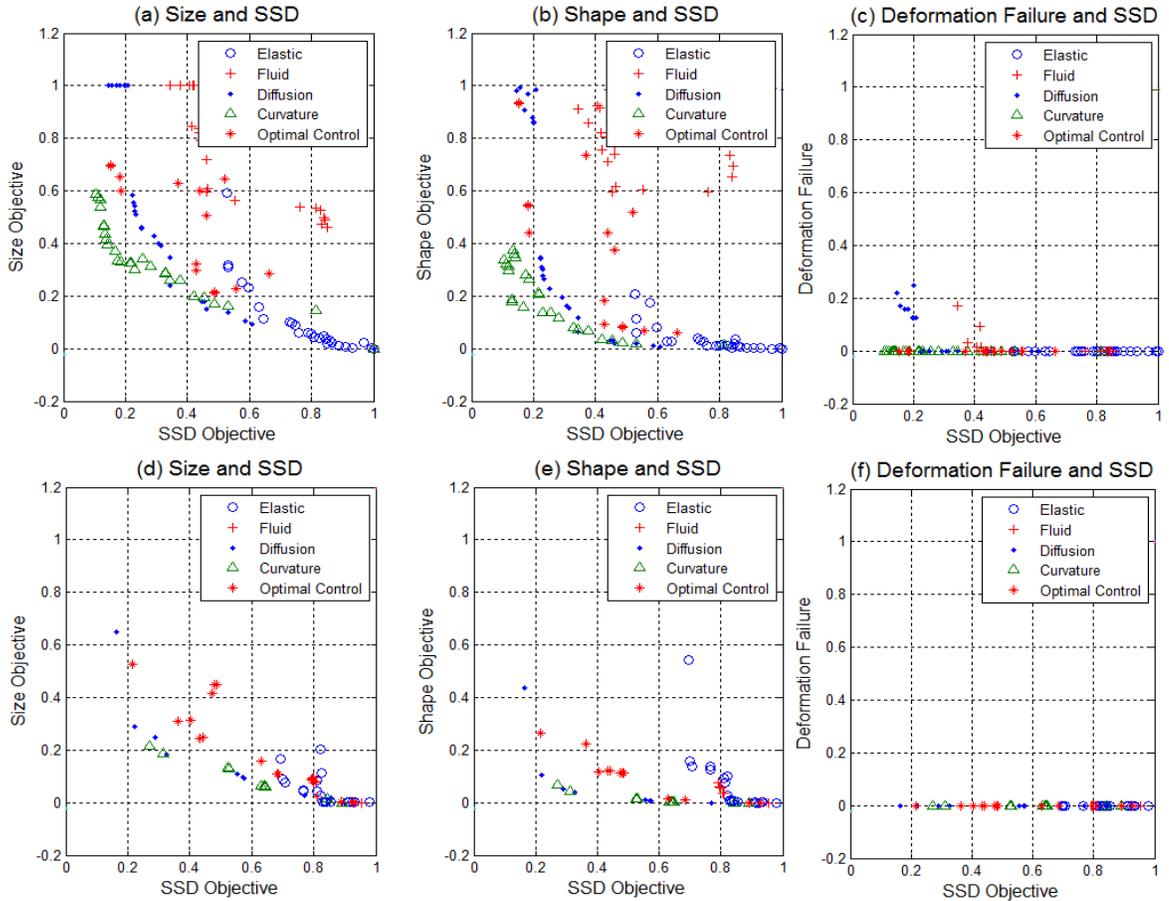


Fig. 6.2: Two dimensional side views of Pareto fronts approximation. The top row with sub-figures labeled (a), (b) and (c) shows the Pareto fronts generated without including the pre-registration step. The bottom row with sub-figures (d), (e) and (f) shows the Pareto fronts generated with the pre-registration step included.

### 6.1.2.1 Comparison with Curvature Model

From Fig. 6.2 (a) and (b), the curvature model dominates the left side of the net Pareto front with better SSD as well as deformation quality based on size and shape. This implies that under the assumption of no pre-registration the curvature model is able to simultaneously optimize SSD, size and shape objectives better than all other models. It can also be observed from Fig. 6.2 (c) that the curvature model has no deformation failure throughout which means it is able to prevent physically unrealizable deformations without the aid of pre-registration. Also its performance did not change much when pre-registration is included except for a slight improvement in size as observed in Fig. 6.2 (d), thus indicating the insensitivity of this model to pre-registration step.

In contrast the optimal control-based model shows a relatively significant improvement in SSD, size and shape objectives when pre-registration step is introduced hence the optimal control-based model is more sensitive to pre-registration than the curvature model. These observations confirm the assertion that the curvature model can do without a pre-registration step. It is also worth noting that the curvature model outperforms the optimal control-based model throughout with regard to size, shape and SSD measures. However, both perform equally well in deformation failure measure. With regard to the sensitivity to pre-registration the curvature model is the best choice in the absence of the pre-registration tools.

### 6.1.2.2 Comparison with Elastic Model

As anticipated, the elastic model manifests signs of stiffness as it is dominated by all methods on the left region of Fig. 6.2 (a) and (b) with respect to the SSD objective. This means the elastic model has a relatively poor image matching ability when the images are not close to one another. Given the high deformations involved in the prostate dynamics it was expected that the elastic model will suffer in convergence with regard to the similarity measure. On their own, elastic and optimal control-based models share their net Pareto front with elastic performing better in size and shape while optimal control is better in SSD (as shown in Fig. 6.2 (a), (b), (d) and (e)).

A stiff model like the elastic model does not distort the shape and size of the grid as much as the less stiff model hence why the elastic model performs better in size and shape objectives. However this good performance in deformation quality due to limited distortions comes at the cost of poor image matching ability. In a severe case it is almost as if no registration was performed due to small deformations occurring. With regard to deformation failure measure the two models are non-dominated. Including pre-registration results in the elastic model performing slightly better with respect to size objective but not in shape objective. Since the elastic model is already stiff and limited to small deformations in its nature, pre-registration cannot be expected to bring much difference with regard to the size, shape and deformation failure objectives. This makes the elastic model less sensitive to pre-registration step than the optimal control-based model with regard to the deformation quality measure in general.

### 6.1.2.3 Comparison with Fluid Model

In the absence of pre-registration, the fluid model is dominated by almost all the other models throughout in terms of size, shape and SSD, except for the elastic model, since the only way for it to participate in the net Pareto front is by allowing it to compete with the elastic model alone. The fluid model dominates the elastic model with regard to the SSD measure thus showing that it is less stiff than the elastic model. Its flexibility, however has come at the burden of compromising its performance with regard to size, shape and deformation failure objectives. This is likely due to not penalizing the displacement field  $\mathbf{u}(\mathbf{x})$  directly but rather the velocity field  $\mathbf{v}(\mathbf{x}, t)$  in its formulation yet the deformation  $\phi(\mathbf{x})$  is itself directly dependent on  $\mathbf{u}(\mathbf{x})$ .

Even though both optimal control and fluid models are not members of the overall net Pareto front, the optimal control-based model would participate in the absence of the curvature model while the fluid model still would not. The fluid model also produced a significant number of physically unrealizable elements, as shown in Fig. 6.2 (c) and this is likely a result of this indirect penalization of the displacement field through the velocity field with the intention of improving flexibility of the model. When pre-registration is included the fluid model significantly improves in all deformation quality measures and starts to participate in the overall net Pareto front thus showing that the fluid model is more reliant on pre-registration step than the optimal control-based model which does not improve as much.

### 6.1.2.4 Comparison with Diffusion Model

Excluding the curvature model for the moment in Fig. 6.2 (a), and (b), we realize that optimal control and diffusion models dominate the left-center region of the net Pareto front side by side. This observation is not surprising given that the first variations of these models are expressible by almost similar Poisson equations, with the exception of optimality equations in the case of the optimal control-based model. With regard to deformation failure, the optimal control-based model is non-dominated with a deformation failure of zero throughout, thus justifying the claim that it gives direct control of element structure. The diffusion model has shown a significant improvement in deformation quality measures relative to that of the optimal control-based model when pre-registration is included. This demonstrates higher sensitivity of the diffusion model to pre-registration with regard to deformation quality measure. With pre-registration included, the net Pareto front of dif-

fusion and optimal control-based models is now occupied mostly by the diffusion model. This shows that the diffusion model is relatively more flexible as mentioned in previous chapters.

### 6.1.2.5 Summary Table

Table 6.1 summarizes the comparisons made in sections 6.1.2.1 to 6.1.2.4 above. Each of the four variational-based models is compared to the optimal control-based model based on the sensitivity to pre-registration, SSD, size, shape and deformation failure objectives. For example, since the diffusion model is more sensitive to pre-registration than the optimal control based model the key word “more” is used to indicate this. With regard to the objectives like SSD the key word “dominated” is used if the a variational-based model is dominated by the optimal control-based model with respect to the SSD objective only otherwise the key word “non-dominated” is used.

Table 6.1: Comparison of variational-based models with the optimal control-based model based on sensitivity to pre-registration, similarity and deformation quality measures.

<b>Model/Objective</b>	<b>Elastic</b>	<b>Fluid</b>	<b>Diffusion</b>	<b>Curvature</b>
<b>Sensitivity</b>	less	more	more	less
<b>SSD</b>	dominated	dominated	non-dominated	dominated
<b>Size</b>	non-dominated	non-dominated	dominated	non-dominated
<b>Shape</b>	non-dominated	non-dominated	non-dominated	non-dominated
<b>Deformation Failure</b>	non-dominated	non-dominated	non-dominated	non-dominated

Considering optimization of all objectives at once as indicated by the distances between the Pareto fronts points and the ideal point (the origin), curvature and diffusion models outperformed the rest of the models. The optimal control-based model on the other hand outperformed both elastic and fluid models.

### 6.1.3 Visualization using Level Diagrams

The visualization presented in Fig. 6.3 is based only on the objectives and distance measures but no decision variables are included. Given Fig. 6.3, a decision maker who prioritizes the SSD objective over other objectives, for example, needs to use Fig. 6.3 (a) and to choose the model with the lowest infinity norm at the desired SSD objective value.

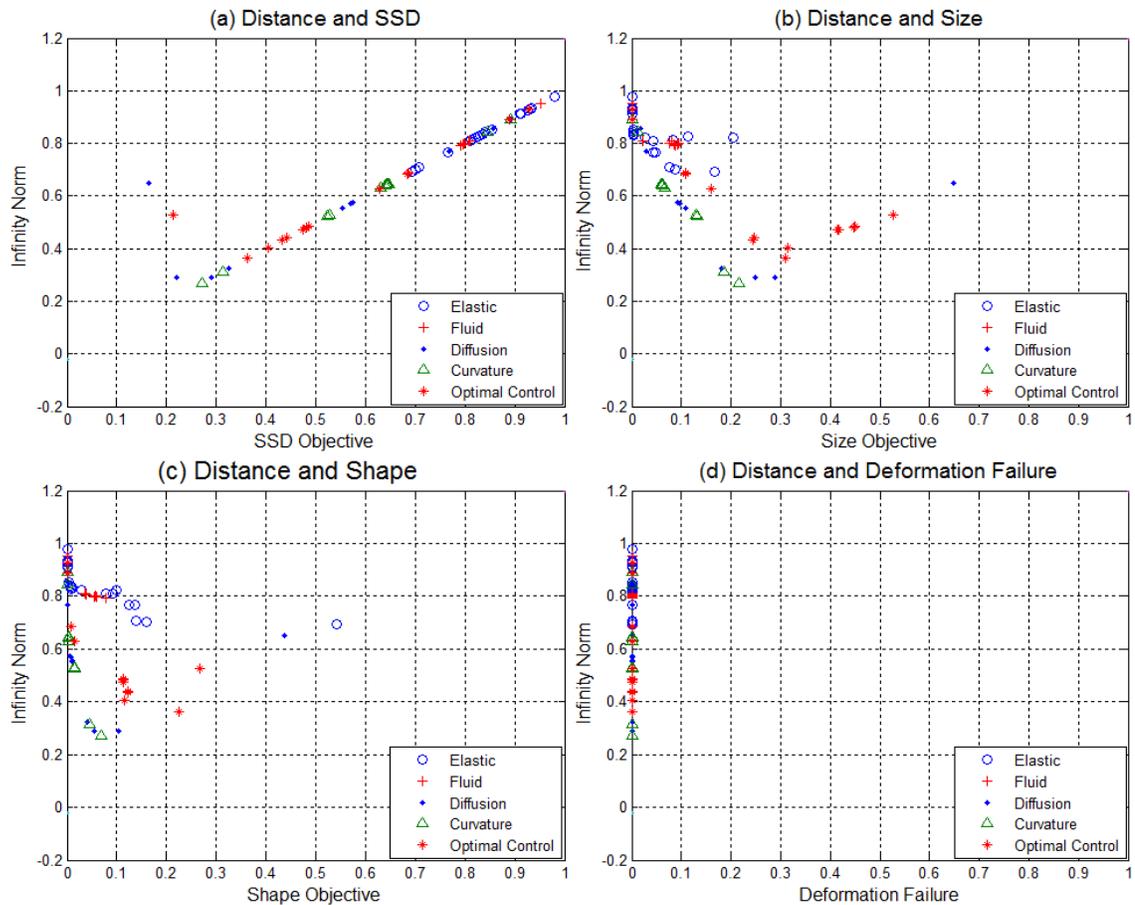


Fig. 6.3: The Level Diagrams for the Pareto Fronts in Fig. 6.2 above.

## 6.2 Experiment Two

In this experiment the CT data was projected from different angles to produce DRRs. The purpose of this experiment is to demonstrate how this evaluation procedure can help in obtaining better or optimal parameter tuning for a particular model.

### 6.2.1 Experimental Setup

The setup for this experiment is the same as the one in the previous experiment except that the experimental data has been changed by varying the projection angles or planes when generating DRRs from the CT data. Also the pre-registration step was not included in this experiment. The results are shown in Fig. 6.4 and their interpretation is presented in the sections below.

## 6.2.2 Results and Interpretations

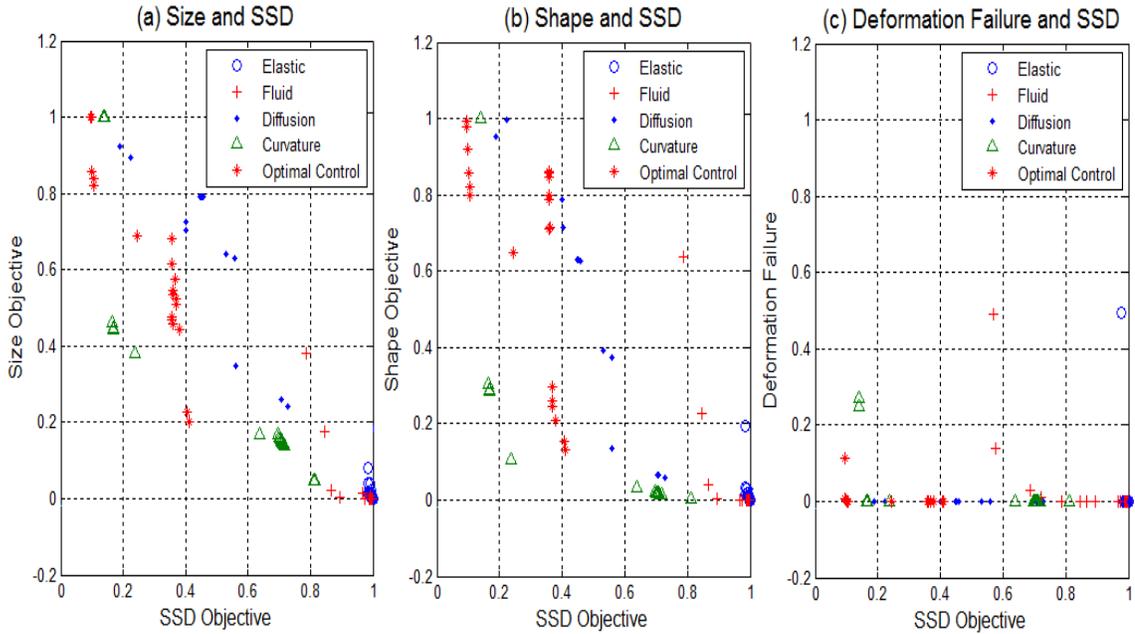


Fig. 6.4: Two dimensional side views of Pareto fronts approximation.

In Fig. 6.4 (a) curvature and optimal control-based models dominate the left region of the net Pareto front. The Diffusion model appears to be inferior to the optimal control-based model. This results from the relatively higher sensitivity of diffusion to pre-registration, as observed in the first experiment. The fluid and elastic models participate in the net front on the rightmost side of the Pareto front with bad SSD measure. Looking at Fig. 6.4 (b) the curvature model is dominating the middle region of the net Pareto front thus minimizing both SSD and shape objectives better than the other models. The optimal control-based model participates on the left side of the net Pareto Front with a good SSD measure but a bad shape measure. The elastic and fluid models dominate the right side of the net Pareto front with good shape measures but poor SSD measure. With regard to deformation failure Fig. 6.4 (c) most models are showing deformation failure and this is likely due to larger deformations involved with comparing differently projected data.

Since the chosen domain of model parameter is between zero and one, sometimes the NSGA-II can pick a parameter close to zero and result in less regulation strength. This leads to instabilities that cause more deformation failures, as mentioned in [1]. This has less to do with the limitations of the model but more to do with the choice of parameters for a particular model. For optimal parameter tuning we use the level diagrams in Fig. 6.5

and Table 6.2 to obtain the model parameters (decision variables) which give rise to the desired point on the level diagrams.

### 6.2.3 Visualization Using Level Diagrams

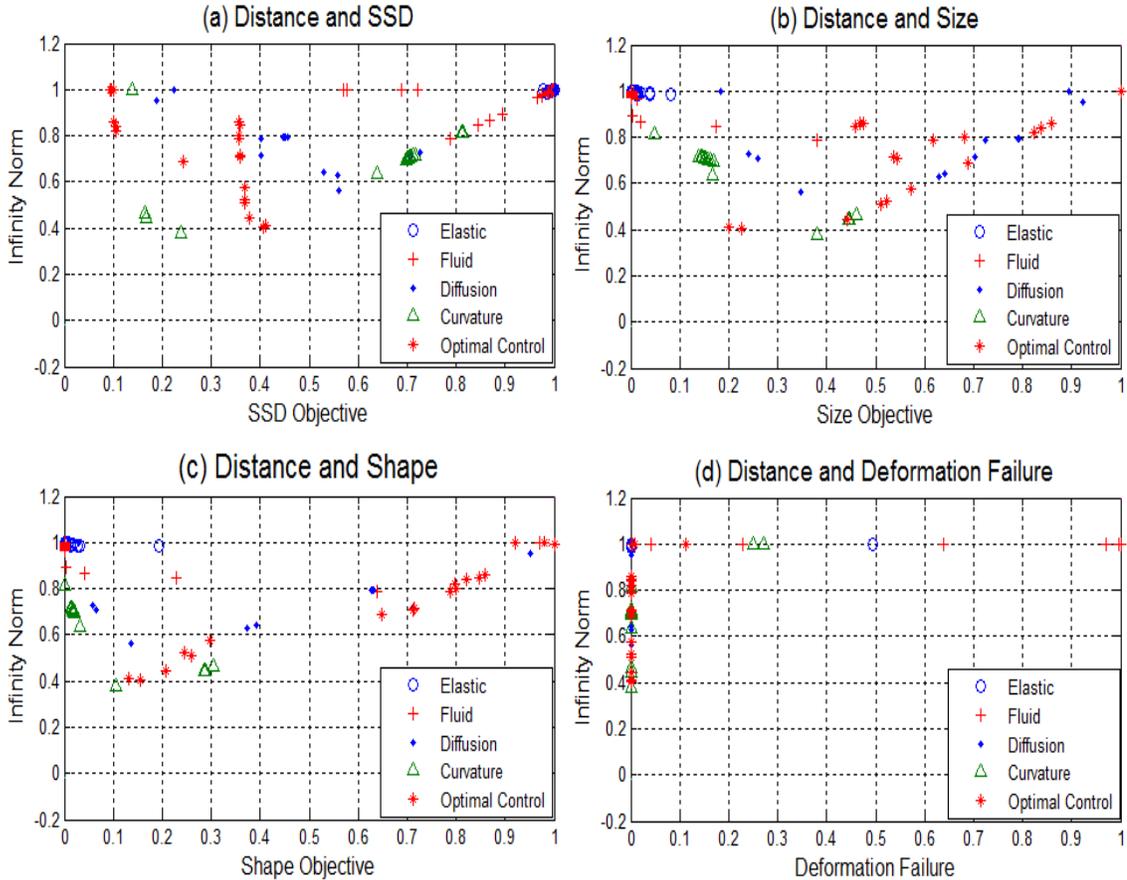


Fig. 6.5: The level diagram analysis for the Pareto fronts in Fig. 6.4 above.

The optimal parameter tuning for a particular model is achieved by locating the point of that particular model which is closest to the origin, thus minimizing all objectives simultaneously. The corresponding model parameters leading to that performance are then located from the table of points produced by NSGA-II. This is demonstrated with the curvature model whose NSGA-II is shown in Table 6.2. It can be observed from the level diagrams in Fig. 6.5 and the corresponding Table 6.2 that the closest curvature model point to the origin is the one with an infinity norm of 0.3793. This point corresponds to the model parameter  $\alpha = 0.0001$  from Table 6.2. Fig. 6.6 shows the result of applying this parameter value in registering two DRRs.

Table 6.2: The NSGA-II minimization output relating to the curvature model.

Parameter $\alpha$	SSD $h_1$	Size $h_2$	Shape $h_3$	Failure $h_4$	$\infty$ -norm
2.254e-05	7.116e-01	1.449e-01	1.558e-02	0	7.116e-01
1.498e-07	1.387e-01	1.000+00	1.000+00	2.715e-01	1.000+00
1.000e-04	2.372e-01	3.793e-01	1.060e-01	0	3.793e-01
1.386e-05	1.675e-01	4.438e-01	2.854e-01	0	4.438e-01
1.589e-05	6.972e-01	1.684e-01	2.131e-02	0	6.972e-01
1.386e-05	1.675e-01	4.438e-01	2.854e-01	0	4.438e-01
1.366e-05	1.672e-01	4.460e-01	2.879e-01	0	4.460e-01
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
2.024e-05	7.058e-01	1.529e-01	1.739e-02	0	7.058e-01

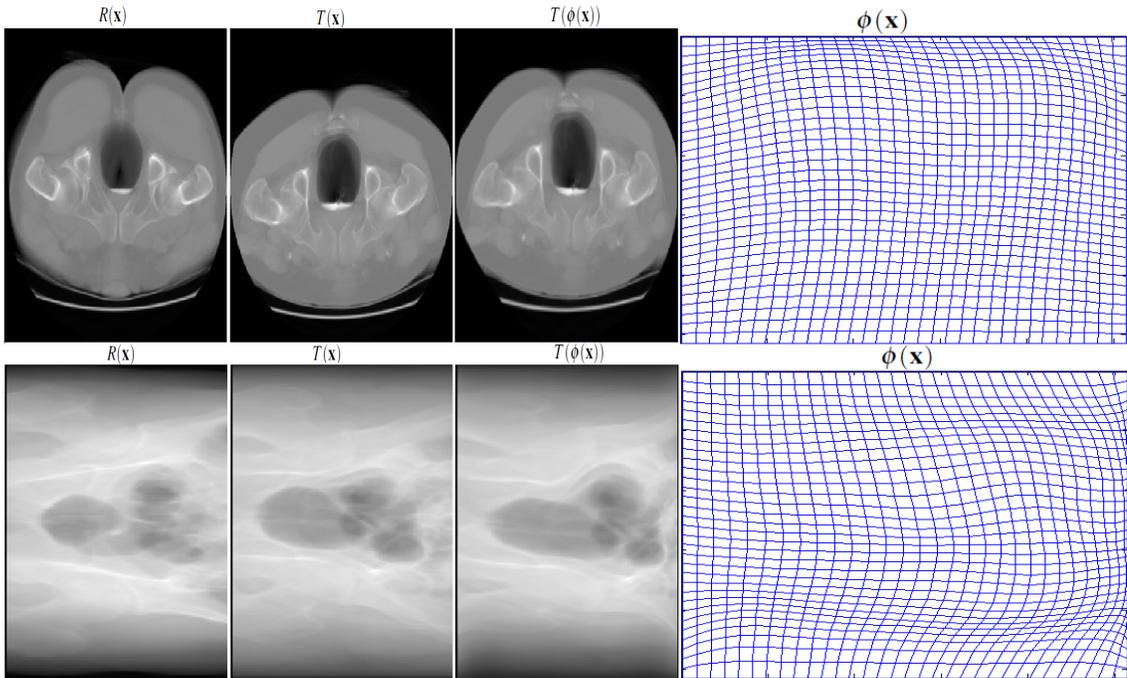


Fig. 6.6: The registration of DRRs by curvature model. Top row show two DRRs produced by cross sectional projection. Bottom row shows two DRRs produced by side view projection.

Having obtained the optimal deformation  $\phi(\mathbf{x})$ , it is used to guide the radiation beam to the targeted tissues precisely. In the case whereby pre-registration is included the overall optimal deformation is comprised of the the optimal transformation obtained from the pre-registration step as well as the optimal deformation  $\phi(\mathbf{x})$  obtained from the deformable model.

# Chapter 7

## Conclusion

The major goal of this work was to carry out an evaluation of the optimal control-based image registration model. As a way of facilitating this evaluation, we used four variational-based models: elastic, fluid, diffusion and curvature models for comparisons and reference purposes. The method of evaluation was chosen as the Pareto front ranking scheme and for facilitation of proper evaluation under this scheme, the objective functions were defined and served as performance markers for evaluating the models performance in multidimensional fitness space. From the Pareto fronts diffusion and curvature models dominated the optimal control-based model with respect to optimizing all objectives taken at once. Under the same scenario the optimal control-based model outperformed fluid and elastic models with regard to all objectives considered at once. So with regard to optimizing all objective functions simultaneously, the curvature model was best followed immediately by the diffusion model and then optimal control-based model. The elastic model performed relatively poorly with regard to optimizing the similarity measure thus confirming the assertion that it is stiff and more appropriate in applications involving small deformations.

The level diagrams were used for visualizing the Pareto fronts, and a demonstration was made on how level diagrams aid in decision making as well as for optimal parameter tuning for a particular model. Given that the decision maker requires the model which best optimizes all objectives, the level diagrams showed that the curvature model is the best choice since it had the least infinity norm of 0.3793. Under the experimental settings in the second experiment the optimal tuning of the curvature model is acquired when  $\alpha = 0.0001$ .

The pre-registration system was built by combining the principal axes transformation

(PAT) method which fast and suitable for initialization, the Levenberg-Marquardt (LM) method and the Nelder-Mead simplex (NMS) method. Evolutionary algorithms and branch and bound method were not used in building the pre-registration system since they are relatively slower than the methods mentioned above. The three methods used in building the pre-registration system together resulted in a sufficiently reliable pre-registration system although at the cost of comprising speed slightly in relation to the speed of individual method. The pre-registration system aided in the initialization of deformable image registration models, and upon comparison it was observed that the fluid and diffusion models are relatively more sensitive to initialization than the optimal control-based model which in turn is more relatively more sensitive than elastic and curvature models.

From this evaluation work we can conclude that the optimal control-based model is well suited for tackling medical image registration applications involving large deformations, such as with the prostate cancer treatment. This follows because the optimal control-based model has proven to be a competitive model for tackling the deformable image registration problem, the prostate image registration in specific. It can also be concluded that pre-registration is necessary for good performance of the optimal control-based model. The absence of deformation failures shown by the optimal control-based model can be counted as evidence that the model has a direct control of the grid structure thus leading to more ease in preventing physically unrealizable deformations as argued in the literature.

## 7.1 Future Work

All models were implemented on a framework of finite elements. However, this results in some models, like the fluid model, being significantly slow due to high computational intensity. In future work, when convergence time is considered as one of the objectives, every model should be implemented in the framework that allows its optimum run-time. Also, evaluation based on different image modalities needs to be considered in future work. More investigations are still required on the development of more computationally efficient and faster solutions to the optimal control problem set in the framework of deformable image registration. Also, the other two versions of GDM need to be investigated as to whether they can provide optimal solutions if used in reformulating the deformable image registration problem as an optimal control problem.

# Bibliography

- [1] Akinlar M. A. *A New Method for Nonrigid Registration of 3D Images*. PhD thesis, The University of Texas at Arlington, 2009.
- [2] Chumchob N. and Chen K. A Robust Affine Image Registration Method. *International Journal of Numerical Analysis and Modeling*, 6(2):311–334, 2009.
- [3] Fischer B. and Modersitzki J. A Unified Approach to Fast Image Registration and A New Curvature-based Registration Technique. *Linear Algebra and its Applications*, 380(0):107–124, 2004.
- [4] Wollny G. and Kruggel F. Computational Cost of Nonrigid Registration Algorithms Based on Fluid Dynamics [MRI Time Series Application]. *Medical Imaging, IEEE Transactions on*, 21(8):946–952, Aug 2002.
- [5] Cahill N. D., Noble J. A., and Hawkes D. J. Demons Algorithms for Fluid and Curvature Registration. In *Biomedical Imaging: From Nano to Macro, 2009. ISBI '09. IEEE International Symposium on*, pages 730–733, June 2009.
- [6] Christensen G. E. and He J. Consistent Nonlinear Elastic Image Registration. In *Mathematical Methods in Biomedical Image Analysis, 2001. MMBIA 2001. IEEE Workshop on*, pages 37–43, 2001.
- [7] Xiaoyan X. and Dony R. D. Fast Fluid Registration using Inverse Filtering for Non-rigid Image Registration. In *Biomedical Imaging: Nano to Macro, 2006. 3rd IEEE International Symposium on*, pages 470–473, April 2006.
- [8] Lee E. and Gunzburger M. An Optimal Control Formulation of an Image Registration Problem. *J. Math. Imaging Vis.*, 36(1):69–80, January 2010.
- [9] Salako S. T. *Optimal Control Approach to Image Registration*. PhD thesis, The University of Texas at Arlington, 2009.

- [10] Peter J. K. and Senthil P. Image Registration for MRI. *Modern Signal Processing*, 48:161–184, 2003.
- [11] MacGillivray S. A. Curvature-based Image Registration: Review and Extensions. Master's thesis, University of Waterloo, 2009.
- [12] Delmon V., Rit S., Pinho R., and Sarrut D. Registration of Sliding Objects using Direction Dependent B-Splines Decomposition. *Physics in Medicine and Biology*, 58(5):1303–1314, 2013.
- [13] Fischer B. and Modersitzki J. Ill-posed Medicine: An Introduction to Image Registration. *Inverse Problems*, 24(3):034008, 2008.
- [14] Sasikala D. and Neelaveni R. Correlation Coefficient Measure of Multimodal Brain Image Registration using fast Walsh Hadamard Transform. *Journal of Theoretical & Applied Information Technology*, 22(2):93 – 101, 2010.
- [15] Roshni V. S. and Revathy K. Using Mutual Information and Cross Correlation as Metrics for Registration of Images. *Journal of Theoretical & Applied Information Technology*, 4(6):474–481, 2008.
- [16] Bachman K. A. Mutual Information-based Registration of Digitally Reconstructed Radiographs and Electronic Portal Images. Master's thesis, University of Colorado at Denver, 2002.
- [17] Klein S. *Optimisation Method for Medical Image Registration*. PhD thesis, Utrecht University, 2008.
- [18] Lin M., Tsai J., and Yu C. A Review of Deterministic Optimization Methods in Engineering and Management. *Mathematical Problems in Engineering*, 2012:01–15, 2012.
- [19] Yu X. and Gen M. *Introduction to Evolutionary Algorithms*. 26–28. Springer, 2010.
- [20] Kelley C. T. *Iterative Methods for Optimization*. 22–24. Society for Industrial and Applied Mathematics, 1999.
- [21] Park Y. C., Chang M. H., and Lee T. Y. A New Deterministic Global Optimization Method for General Twice-Differentiable Constrained Nonlinear Programming Problems. *Engineering Optimization*, 39(4):397–411, 2007.

- [22] Cagnoni S., Lutton E., and Olague G. *Genetic and Evolutionary Computation for Image Processing and Analysis*. 1–22. Hindawi Publishing Corporation, 2007.
- [23] Cosío F. A. Automatic Initialization of an Active Shape Model of the Prostate. *Medical Image Analysis*, 12(4):469 – 483, 2008.
- [24] Hager W. W. and Park S. The Gradient Projection Method with Exact Line Search. *Journal of Global Optimization*, 30(1):103–118, 2004.
- [25] Klein S., Staring M., and Pluim J. P. W. Evaluation of Optimization Methods for Nonrigid Medical Image Registration Using Mutual Information and B-Splines. *Image Processing, IEEE Transactions on*, 16(12):2879–2890, Dec 2007.
- [26] Jeffrey C. L., James A. R., Margaret H. W., and Paul E. W. Convergence Properties of the Nelder-Mead Simplex Method in Low Dimensions. *SIAM J. OPTIM.*, 9(1):112–147, 1998.
- [27] Adjiman C. S., Dallwig S., Floudas C. A., and Neumaier A. A Global Optimization Method,  $\alpha$ BB, for General Twice-Differentiable Constrained NLPs – I. Theoretical Advances. *Computers & Chemical Engineering*, 22:1159–1179, 1998.
- [28] Akrotirianakis I. G. and Floudas C. A. A New Class of Improved Convex Underestimators for Twice Continuously Differentiable Constrained NLPs. *Journal of Global Optimization*, 30(4):367–390, 2004.
- [29] Damas S., Cordon O., and Santamaria J. Medical Image Registration Using Evolutionary Computation: An Experimental Survey. *Computational Intelligence Magazine, IEEE*, 6(4):26–42, Nov 2011.
- [30] Gao F. and Han L. Implementing the Nelder-Mead Simplex Algorithm with Adaptive Parameters. *Computational Optimization and Applications*, 51(1):259–277, 2012.
- [31] Hsieh C. *Nonrigid Image Registration by The Deformation Based Grid Generation*. PhD thesis, The University of Texas at Arlington, 2008.
- [32] Sotiras A., Davatzikos C., and Paragios N. Deformable Medical Image Registration: A Survey. *IEEE Transactions on Medical Imaging*, 32:1153–1189, 2013.
- [33] Bajcsy R. and Kovačič S. Multiresolution Elastic Matching. *Comput. Vision Graph. Image Process.*, 46(1):1–21, April 1989.

- [34] Christensen G. E., Rabbitt R. D., and Miller M. I. Deformable Templates using Large Deformation Kinematics. *Image Processing, IEEE Transactions on*, 5(10):1435–1447, Oct 1996.
- [35] Thirion J. P. Image Matching as a Diffusion Process: An Analogy with Maxwell’s Demons. *Medical Image Analysis*, 2(3):243–260, 1998.
- [36] Fischer B. and Modersitzki J. Curvature Based Image Registration. *Journal of Mathematical Imaging and Vision*, 18(1):81–85, 2003.
- [37] Forsberg Daniel, Andersson Mats, and Knutsson Hans. Non-rigid Diffeomorphic Image Registration of Medical Images Using Polynomial Expansion. In Auralio Campilho and Mohamed Kamel, editors, *Image Analysis and Recognition*, volume 7325 of *Lecture Notes in Computer Science*, pages 304–312. Springer Berlin Heidelberg, 2012.
- [38] Vercauteren T., Pennec X., Perchant A., and Ayache N. Diffeomorphic Demons: Efficient Non-parametric Image Registration. *NeuroImage*, 45(1, Supplement 1):S61–S72, 2009. Mathematics in Brain Imaging.
- [39] Ashburner J. A Fast Diffeomorphic Image Registration Algorithm. *NeuroImage*, 38(1):95–113, 2007.
- [40] Vercauteren Tom, Pennec Xavier, Perchant Aymeric, and Ayache Nicholas. Non-parametric diffeomorphic image registration with the demons algorithm. In *Proceedings of the 10th International Conference on Medical Image Computing and Computer-assisted Intervention*, MICCAI’07, pages 319–326, Berlin, Heidelberg, 2007. Springer-Verlag.
- [41] Shusharina N. and Sharp G. Image Registration using Radial Basis Functions with Adaptive Radius. *Med Phys*, 39(11):6542–6549, 11 2012.
- [42] Kohlrausch J., Rohr K., and Stiehl H. S. A new class of elastic body splines for non-rigid registration of medical images. *J. Math. Imaging Vis.*, 23(3):253–280, November 2005.
- [43] Rueckert D., Sonoda L. I., Hayes C., Hill D. L. G., Leach M. O., and Hawkes D. J. Nonrigid Registration using Free-form Deformations: Application to Breast MR Images. *IEEE Transactions on Medical Imaging*, 18:712–721, 1999.

- [44] Dinsenhacher T., Rohde G. K., Hardin D. P., Aldroubi A., and Dawant B. M. Multiscale Nonrigid Data Registration using Adaptive Basis Functions. *Proc. SPIE*, 4119:1076–1083, 2000.
- [45] Freiman M., Voss S. D., and Warfield S. K. Abdominal Images Non-rigid Registration Using Local-affine Diffeomorphic Demons. In *Proceedings of the Third International Conference on Abdominal Imaging: Computational and Clinical Applications*, MICCAI'11, pages 116–124, Berlin, Heidelberg, 2012. Springer-Verlag.
- [46] Wang Y. and Staib H. L. Elastic Model Based Non-rigid Registration Incorporating Statistical Sshape Information. In WilliamM. Wells, Alan Colchester, and Scott Delp, editors, *Medical Image Computing and Computer-Assisted Intervention - MICCAI'98*, volume 1496 of *Lecture Notes in Computer Science*, pages 1162–1173. Springer Berlin Heidelberg, 1998.
- [47] Schnabel J. A., Tanner C., Castellano-Smith A. D., Leach M. O., Hayes C., Degenhard A., Hose R. D., Hill D. L. G., and Hawkes D. J. Validation of Non-rigid Registration Using Finite Element Methods. In *Proceedings of the 17th International Conference on Information Processing in Medical Imaging*, IPMI '01, pages 344–357, London, UK, UK, 2001. Springer-Verlag.
- [48] Rao A. V. A Survey of Numerical Methods for Optimal Control. *Advances in the Astronautical Sciences*, 135:01–32, 2010.
- [49] Strang G. *Computational Science and Engineering*. 303–304. Wellesley Cambridge Press, 01 Nov 2007.
- [50] Winckel G. Legendre-Gauss Quadrature Weights and Nodes. <http://www.mathworks.com/matlabcentral/fileexchange/4540-legendre-gauss-quadrature-weights-and-nodes>, Feb 2004. Online; Accessed 11-Feb-2014.
- [51] Liu G. R. and Quek S. S. *The Finite Element Method: A Practical Course*. 143–144. Elsevier Science Ltd., 2003.
- [52] Fleitas D. L. *The Least-Squares Finite Element Method for Grid Deformation and Mesh Applications*. PhD thesis, The University of Texas Arlington, 2005.
- [53] Liao G. and Anderson D. A New Approach to Grid Generation. *Applicable Analysis*, 44(3-4):285–298, 1992.

- [54] Chen H., Hsieh C., and Liao G. Nonrigid Image Registration Using Adaptive Grid Generation: Preliminary Results. In *Biomedical Imaging: From Nano to Macro, 2007. ISBI 2007. 4th IEEE International Symposium on*, pages 580–583, April 2007.
- [55] Liao G. and Su J. A Direct Method in Dacorogna-Moser’s Approach of Grid Generation Problems. *Applicable Analysis*, 49(1-2):73–84, 1993.
- [56] Liao G. and Su J. Grid Generation via Deformation. *Applied Mathematics Letters*, 5(3):27–29, 1992.
- [57] Liao G., Pan T., and Su J. Numerical Grid Generator Based on Moser’s Deformation Method. *Numerical Methods for Partial Differential Equations*, 10(1):21–31, 1994.
- [58] Liao G. J. and Su J. Z. A Moving Grid Method for (1 + 1) Dimension. *Applied Mathematics Letters*, 8(4):47–49, 1995.
- [59] Patel M. Algorithm for Adaptive Grid Generation Using Galerkin Finite Element Method. Master’s thesis, The University of Texas at Arlington, December 2009.
- [60] Liu J. *New Development of the Deformation Method*. PhD thesis, The University of Texas at Arlington, 2006.
- [61] Liao G., Cai X., Fleitas D., Luo X., Wang J., and Xue J. Optimal Control Approach to Data Set Alignment. *Applied Mathematics Letters*, 21(9):898–905, 2008.
- [62] Lee E. and Gunzburger M. Analysis of Finite Element Discretizations of an Optimal Control Formulation of the Image Registration Problem. *SIAM Journal on Numerical Analysis*, 49(4):1321–1349, 2011.
- [63] Jiang B. *The Least-Squares Finite Element Method*. Theory and Applications in Computational Fluid Dynamics and Electromagnetics. 15–17. Springer, 1998.
- [64] Hsieh C., Chen H., Lin T., Hsiao H., Chu M., Liao G., and Zhong H. On the Development of a New Non-Rigid image Registration using Deformation-based Grid Generation. *Proc. SPIE 6914, Medical Imaging 2008: Image Processing 69140W*, 6914:69140W–69140W–12, March 11, 2008.
- [65] Cardoso J. S. and Corte-Real L. Toward a Generic Evaluation of Image Segmentation. *Image Processing, IEEE Transactions on*, 14(11):1773–1782, Nov 2005.
- [66] Everingham M., Muller H., and Thomas B. Evaluating Image Segmentation Algorithms Using the Pareto Front. In *Proc. Seventh European Conf. Computer Vision*, pages 34–48, 2002.

- [67] Zhang H., Fritts J. E., and Goldman S. A. Image Segmentation Evaluation: A Survey of Unsupervised Methods. *Comput. Vis. Image Underst.*, 110(2):260–280, May 2008.
- [68] Kadoya N., Fujita Y., Katsuta Y., Dobashi S., Takeda K., Kishi K., Kubozono M., Umezawa R., Sugawara T., Matsushita H., and Jingu K. Evaluation of Various Deformable Image Registration Algorithms for Thoracic Images. *Journal of Radiation Research*, 55(1):175–182, January 2014.
- [69] Lawson D. J., Schreibmann E., Jani B. A., and Fox T. Quantitative Evaluation of a Cone Beam Computed Tomography (CBCT)-CT Deformable Image Registration Method for Adaptive Radiation Therapy. *Journal of Applied Clinical Medical Physics*, 8(4):96–113, 2007.
- [70] Zhong H. and Siebers J. V. Quantitative Evaluation of Deformable Image Registration. In *Biomedical Imaging: From Nano to Macro, 2007. ISBI 2007. 4th IEEE International Symposium on*, pages 724–727, April 2007.
- [71] Guckenberger M., Baier K., Richter A., Wilbert J., and Flentje M. Evaluation of Surface-based Deformable Image Registration for Adaptive Radiotherapy of Non-small Cell Lung Cancer (NSCLC). *Radiation Oncology*, 4(1):1–13, 2009.
- [72] Liu F., Hu Y., Zhang Q., Kincaid R., Goodman K. A., and Mageras G. S. Evaluation of Deformable Image Registration and a Motion Model in CT Images with Limited Features. *Physics in Medicine and Biology*, 57(9):2539–2554, 2012.
- [73] Ying Wei. Nonrigid Image Registration Evaluation Using Common Evaluation Databases. Master’s thesis, The University of Iowa, 2009.
- [74] Deb K., Pratap A., Agarwal S., and Meyerivan T. A Fast and Elitist Multi-Objective Genetic Algorithm: NSGA-II. *Evolutionary Computation, IEEE Transactions on*, 6(2):182–197, Apr 2002.
- [75] Seshadri A. NSGA - II: A Multi-objective Optimization Algorithm. [http://www.mathworks.com/matlabcentral/fileexchange/10429-nsga-ii-a-multi-objective-optimization-algorithm/content/NSGA-II/html/non\\_domination\\_sort\\_mod.html](http://www.mathworks.com/matlabcentral/fileexchange/10429-nsga-ii-a-multi-objective-optimization-algorithm/content/NSGA-II/html/non_domination_sort_mod.html), 19 March 2009. A Documentation File for Matlab Code Implementation. Online; Accessed 13-Feb-2014.
- [76] Knupp P. M. Hexahedral Mesh Untangling Algebraic Mesh Quality Metrics. In *9th International Meshing Roundtable*, pages 173–183, October 2000.

- [77] Knupp P. M. Algebraic Mesh Quality Metrics. *SIAM J. Sci. Comput.*, 23(1):193–218, 2001.
- [78] Knupp P. M. Algebraic Mesh Quality Metrics for Unstructured Initial Meshes. *Finite Elem. Anal. Des.*, 39(3):217–241, January 2003.
- [79] Zio E. and Bazzo R. Level Diagrams Analysis of Pareto Front for Multiobjective System Redundancy Allocation. *Reliability Engineering & System Safety*, 96(5):569–580, 2011.
- [80] Blasco X., Herrero J. M., Sanchis J., and Martinez M. A New Graphical Visualization of n-Dimensional Pareto Front for Decision-making in Multiobjective Optimization. *Information Sciences*, 178(20):3908–3924, 2008. Special Issue on Industrial Applications of Neural Networks 10th Engineering Applications of Neural Networks 2007.