Elastic Image Registration using Parametric Deformation Models

Jan Kybic

Thèse N° 2439 (2001)

Thèse présentée au département de microtechnique
pour l'obtention du grade de docteur ès sciences techniques
et acceptée sur proposition du jury:

Prof. Philippe Renaud, *président*
Prof. Michael Unser, *directeur de thèse*
Prof. Martin Vetterli, *co-rapporteur*
Prof. Gabor Szekely, *co-rapporteur*
Prof. Gary E. Christensen, *co-rapporteur*

Ecole Polytechnique Fédérale de Lausanne—2001

Cover design by Annette Unser
Cover printed by Copy-Tunnel
Printing and binding by Repro-EPFL
Typeset with LaTeX
Copyright © 2001 by Jan Kybic
Available at `http://bigwww.epfl.ch/kybic/`

Abstract

The main topic of this thesis is elastic image registration for biomedical applications. We start with an overview and classification of existing registration techniques. We revisit the landmark interpolation which appears in the landmark-based registration techniques and add some generalizations.

We develop a general elastic image registration algorithm. It uses a grid of uniform B-splines to describe the deformation. It also uses B-splines for image interpolation. Multiresolution in both image and deformation model spaces yields robustness and speed. First we describe a version of this algorithm targeted at finding unidirectional deformation in EPI magnetic resonance images. Then we present the enhanced and generalized version of this algorithm which is significantly faster and capable of treating multidimensional deformations. We apply this algorithm to the registration of SPECT data and to the motion estimation in ultrasound image sequences.

A semi-automatic version of the registration algorithm is capable of accepting expert hints in the form of soft landmark constraints. Much fewer landmarks are needed and the results are far superior compared to pure landmark registration.

In the second part of this thesis, we deal with the problem of generalized sampling and variational reconstruction. We explain how to reconstruct an object starting from several measurements using arbitrary linear operators. This comprises the case of traditional as well as generalized sampling. Among all possible reconstructions, we choose the one minimizing an a priori given quadratic variational criterion. We give an overview of the method and present several examples of applications. We also provide the mathematical details of the theory and discuss the choice of the variational criterion to be used.

Résumé

Le sujet principal de cette thèse est la mise en correspondance élastique pour des applications biomédicales. Nous commençons par résumer et classer les méthodes existantes de mise en correspondance. Nous revisitons les techniques d'interpolation utilisées pour la mise en correspondance basée sur des repères.

Nous avons développé un algorithme général de mise en correspondance élastique. La déformation est décrite à l'aide de B-splines posés sur une grille uniforme. Nous utilisons les B-splines également pour interpoler les images. L'apport de la multirésolution — pour l'image et pour le modèle de déformation — rend l'algorithme rapide et robuste.

Premièrement, nous présentons une version de l'algorithme destinée à la correction des déformations unidirectionnelles dans les images fonctionnelles de résonance magnétique. Deuxièmement, nous décrivons une version améliorée et généralisée de cet algorithme, capable de traiter des déformations multidimensionnelles. Nous appliquons cet algorithme aux problèmes de la création d'un atlas des images SPECT et de l'estimation de mouvement dans les séquences ultrasoniques.

Un extension de l'algorithme permet d'utiliser les repères choisises par un expert comme des indications pour améliorer la qualité de la mise en correspondence dans les cas difficiles.

Dans la deuxième partie de la thèse, nous traitons le problème de l'échantillonnage généralisé et de la reconstructions variationnelle. Nous expliquons comment reconstruire un objet à partir de plusieurs mesures linéaires. Parmi toutes les reconstructions, nous choisissons celle qui minimise un critère quadratique variationel. Nous présentons une vue d'ensemble de cette méthode ainsi que plusieurs exemples. Nous discutons le choix du critère à employer.

To Petr Čížek, the brightest person I have ever met.
For sharing his joy of thinking with me
and being a great friend.

Acknowledgements

This thesis describes the work I carried out as a part of my Ph.D. studies at *Ecole Polytechnique Fédérale de Lausanne*. Although I can swear I wrote every letter of this document, I know it would have never existed without the help of many other people.

First, I am very grateful to my thesis advisor, Prof. Michael Unser, for receiving me kindly back in winter 1998 and convincing me to stay with him. I have never regretted this decision. The mathematical erudition of Thierry Blu left me standing in awe. I am glad I could collaborate on the variational reconstruction problem with him.

I would like to thank Arto Nirkko from the *Inselspital Bern*, for providing us with the MRI data, and Daniel Slosman and Christian Chicherio from the *Hôpital cantonal universitaire de Genève*, for supplying the SPECT images. I am also much obliged to María Jesús Ledesma-Carbayo from *Universidad Politécnica de Madrid* who let me use the ultrasound sequences from *Laboratorio de Ecocardiografía del Hospital General Universitario Gregorio Marañón, Madrid*, which I hitherto acknowledge; I thoroughly enjoyed working with her.

I sincerely appreciated the sympathetic environment we could create in our *Biomedical Imaging Group*. My office mate Arrate Muñoz should be especially recognized for the support she gave me during all those years.

My foremost thanks go to all my friends and acquaintances I met while in Lausanne, too numerous to list here. You made me feel welcome, and I had a great time with you. In particular, our little Czech community brought me a touch of home that I valued highly. I am also more than grateful to all the friends I left behind but who have not forgotten me nevertheless.

Last but not least, I would have never got this far without the support and love of my parents and Family. Let this be a gratification to you.

Jan
Lausanne, May 2001

Contents

Abstract **1**

Résumé **2**

Acknowledgements **5**

Contents **7**

1 Introduction **13**
1.1 Image registration 13
1.1.1 Examples of image registration 13
1.1.2 Image warping 14
1.2 Applications of image registration 14
1.2.1 Biomedical applications 17
1.3 Contributions of this thesis 17

2 A Review of Registration Techniques **19**
2.1 Feature space 22
2.1.1 Pixel-based registration 22
2.1.2 Transform-based registration 22
2.1.3 Feature-based registration 22
2.2 Search space 23
2.2.1 Local models 23
2.2.2 Global models 24
2.2.3 Semi-local model 24
2.2.4 Image dependent models 24
2.3 Similarity metrics 24
2.3.1 Data term 24
2.3.2 Regularization 25
2.4 Search strategy 25

2.4.1 Direct solution . . . 25
2.4.2 Exhaustive search . . . 25
2.4.3 Dynamic programming . . . 26
2.4.4 Partial differential equations . . . 26
2.4.5 Multidimensional optimization methods . . . 26
2.4.6 Multiresolution . . . 26
2.5 Other attributes . . . 27
2.6 Complementary surveys . . . 27

3 Landmark Registration **29**
3.1 Landmark registration . . . 29
3.1.1 Motivation of this study . . . 30
3.2 Interpolation versus approximation . . . 30
3.2.1 Hard constraints . . . 30
3.2.2 Soft constraints . . . 31
3.2.3 Desirable properties . . . 31
3.3 Thin-plate splines . . . 32
3.3.1 Variational formulation . . . 32
3.3.2 Interpolation formula . . . 33
3.3.3 Solution justified . . . 34
3.4 Fractional splines . . . 34
3.4.1 Fractional derivatives . . . 34
3.4.2 The fractional criterion . . . 35
3.4.3 Fractional Brownian motion . . . 36
3.4.4 Bayesian interpretation . . . 36
3.4.5 Fractional landmark interpolation . . . 37
3.5 Localization . . . 38
3.5.1 Localizing $|x|^3$ . . . 38
3.5.2 Localizing general variational problem solution . . . 40
3.6 Conclusions . . . 43

4 Unwarping of EPI **45**
4.1 Abstract . . . 45
4.2 Introduction . . . 46
4.2.1 EPI features . . . 46
4.2.2 The reasons to unwarp . . . 46
4.2.3 Existing distortion correction techniques . . . 48
4.2.4 Unwarping by registration . . . 48
4.3 Proposed algorithm . . . 49
4.3.1 Semi-local model . . . 49
4.3.2 Univariate case . . . 50

4.3.3 Splines—a perfect fit . . . 51
4.3.4 Data criterion . . . 52
4.3.5 Bivariate case . . . 52
4.3.6 Optimization algorithm . . . 53
4.3.7 Image interpolation model . . . 54
4.3.8 Multiresolution . . . 54
4.3.9 Invertibility . . . 55
4.3.10 Preprocessing . . . 56
4.4 Experiments . . . 57
4.4.1 General comments . . . 57
4.4.2 Error measurement . . . 57
4.4.3 Sources of error . . . 57
4.4.4 Deformation generator . . . 59
4.4.5 Ideal case . . . 60
4.4.6 Image interpolation order . . . 62
4.4.7 Noisy case . . . 63
4.4.8 Out-of-space deformation . . . 63
4.4.9 Multiresolution strategy . . . 64
4.4.10 Artificial EPI images . . . 65
4.4.11 Real EPI images . . . 65
4.5 Conclusions . . . 68
4.6 Acknowledgment . . . 68
4.6.1 Optimality of the Cubic Spline Model . . . 69
4.6.2 B-Splines . . . 70
4.6.3 Explicit Derivatives . . . 70

5 Fast Multidimensional Registration **73**
5.1 Abstract . . . 73
5.2 Proposed algorithm . . . 73
5.2.1 Organization of this Chapter . . . 74
5.2.2 Registration as minimization . . . 74
5.2.3 Image model . . . 76
5.2.4 Deformation model structure . . . 76
5.2.5 Difference measure . . . 76
5.3 Choosing the deformation basis . . . 77
5.3.1 Splines versus wavelets . . . 79
5.3.2 B-spline deformation model . . . 79
5.3.3 Subspace optimization . . . 80
5.4 Optimization strategy . . . 81
5.4.1 Optimization algorithms . . . 81
5.4.2 Choosing the best optimizer . . . 83

5.4.3 Multiresolution . . . 85
5.5 Semi-automatic registration . . . 86
5.5.1 Virtual springs . . . 86
5.6 Implementation issues . . . 88
5.6.1 Explicit derivatives . . . 88
5.6.2 Hessian approximation . . . 89
5.6.3 Gradient calculation as convolution . . . 89
5.6.4 Multiresolution spline representation . . . 90
5.6.5 Consequences of finite support . . . 91
5.6.6 Image size change . . . 91
5.6.7 Fast spline calculations . . . 92
5.6.8 Stopping criterion . . . 92
5.6.9 Masking . . . 92
5.6.10 Invertibility and regularization . . . 94
5.7 Experiments . . . 95
5.7.1 Registration of MRI brain slices . . . 95
5.7.2 Out-of-space deformation . . . 96
5.7.3 Choosing the spline degree . . . 100
5.7.4 Starting point . . . 103
5.7.5 Statistical distribution of errors . . . 103
5.8 Further applications . . . 103
5.8.1 SPECT atlas . . . 103
5.8.2 Ultrasound for heart motion recovery . . . 106
5.9 Conclusions . . . 109

6 Variational Reconstruction: Tutorial 111
6.1 Abstract . . . 111
6.2 Introduction . . . 112
6.2.1 Perfect reconstruction . . . 112
6.2.2 Consistent reconstruction . . . 112
6.2.3 Non-uniform sampling . . . 113
6.2.4 Motivation for variational reconstruction . . . 114
6.2.5 Proposed variational reconstruction . . . 114
6.2.6 Existing work . . . 115
6.2.7 Organization of the paper . . . 115
6.3 General description of the solution . . . 117
6.3.1 Problem description . . . 117
6.3.2 Choice of the criterion . . . 118
6.3.3 Fundamental solution . . . 121
6.3.4 Explicit solution . . . 122
6.3.5 Generalized approximation problem . . . 124

6.4 Examples . . . 125
6.4.1 Reconstruction from irregular samples . . . 125
6.4.2 Smoothing splines . . . 127
6.4.3 Derivative sampling . . . 127
6.4.4 Landmark based warping . . . 128
6.4.5 Reconstruction consistent with Laplace equation . . . 129
6.4.6 Derivative sampling in 2D . . . 130
6.4.7 Tomographic reconstruction . . . 131
6.5 Conclusions . . . 134

7 Variational Reconstruction: Theory **137**
7.1 Abstract . . . 137
7.2 Introduction . . . 137
7.3 Formulation . . . 139
7.3.1 Notation . . . 139
7.3.2 Distributions . . . 140
7.3.3 Problem definition . . . 140
7.3.4 Properties of the solution . . . 141
7.4 Bilinear forms . . . 143
7.4.1 Quadratic criterion and linear constraints . . . 143
7.4.2 Operator kernel of a bilinear form . . . 144
7.4.3 Convolutional kernel . . . 144
7.4.4 Fourier form . . . 144
7.4.5 Extending the bilinear form . . . 145
7.4.6 Factorizing the convolutional kernel . . . 145
7.5 Imposing invariance properties . . . 147
7.5.1 Geometric translation invariance . . . 147
7.5.2 Rotational invariance . . . 148
7.5.3 Scale invariance . . . 148
7.5.4 Matrix linearity . . . 149
7.5.5 Form of the criterion . . . 149
7.5.6 Laplacian semi-norm . . . 150
7.5.7 Duchon's semi-norms . . . 150
7.5.8 Semi-norms for fractional derivatives . . . 151
7.5.9 Corresponding bilinear forms . . . 151
7.6 Solution of the variational problem . . . 151
7.6.1 Lagrange multipliers . . . 152
7.6.2 Introducing fundamental solutions . . . 152
7.6.3 Explicit solution of the variational problem . . . 153
7.6.4 Linear equation set . . . 154
7.6.5 Finding the fundamental solutions . . . 154

7.6.6 Green functions 155
7.6.7 Unicity of the solution 156
7.7 Approximation problem 157
7.7.1 Least-squares approximation 158
7.8 Conclusions 158
7.9 Appendix 159
A Linearity with respect to measurements 159
B Difference between two solutions 159
C GIP solution 159
D Dirac Laplacian 159

Bibliography **161**

Conclusions **173**

Availability of the software **175**

Curriculum vitæ **177**

Chapter 1

Introduction

1.1 Image registration

Suppose we have two images which we shall call *reference* and *test*. These images depict the same or similar object but they are not identical. The task of the *image registration* is to find homologous (corresponding) points in both images.

For us, the output of the image registration is a *correspondence function* $\mathbf{g}$ such that $\mathbf{x}_t = \mathbf{g}(\mathbf{x}_r)$. The function $\mathbf{g}$ takes a coordinate $\mathbf{x}_r$ of any point in the reference image and returns a coordinate $\mathbf{x}_t$ of a corresponding point in the test image.

We will call a registration *elastic*, if the family of correspondence functions $\mathbf{g}$ is sufficiently general, capable of expressing (almost arbitrary) nonlinear relations.[1] A registration which is not elastic might consider for example only linear functions $\mathbf{g}$.

1.1.1 Examples of image registration

As an example consider the two portraits in Figure 1.1, where corresponding points such as eyes, the chin, or the tip of the nose can be easily identified. A more complicated example is presented in Figure 1.2. Here some zones from the left image do not have any corresponding region in the right image. This illustrates some vagueness of the registration problem which we will have to address. Finally, Figure 1.3 shows an example from the biomedical domain presenting another difficulty: Note that although the images represent exactly the same object, the aspect of some of its parts is radically different in both.

[1] Elasticity in a stricter sense is sometimes associated with a reference state that the object returns to once all loads are removed. There is no such notion for our correspondence function.

Figure 1.1: Michael Unser (left), my thesis director, and Philippe Thévenaz (right), the first assistant.

1.1.2 Image warping

Image warping or image deformation is in some sense the inverse of image registration, as illustrated in Figure 1.4. The image registration takes the test and reference images and yields a correspondence (deformation) function. Conversely, the image warping takes the test image and a correspondence function and outputs a warped image which is a deformed version of the test image. If all has worked correctly, then the warped test image should be aligned with the reference image. Conversely, applying an image registration on an original of an image and its warped version recovers the deformation used.

Given an image $f(\mathbf{x})$ and a deformation function $\mathbf{g}$, the image warping yields a warped image $f_w(\mathbf{x}) = f\big(\mathbf{g}(\mathbf{x})\big)$.

1.2 Applications of image registration

Applications of image registration emerged in the domain of *motion analysis*. The task is to find changes between two subsequent frames in a video sequence. We assume that these changes can be completely explained by movements of the objects in the scene or movements of the observer. The extracted motion field can be exploited in numerous ways. We can use it directly to measure the trajectories, distances, speeds,

Figure 1.2: James Bond (left), his friend Lupe (right).

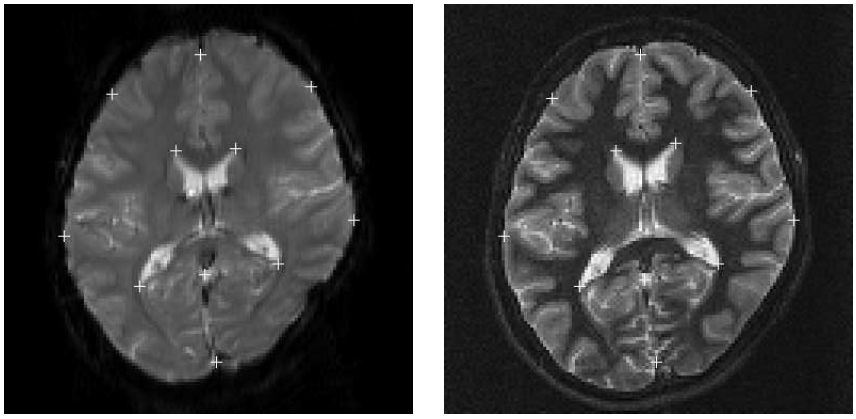


Figure 1.3: Corresponding magnetic resonance brain slices from EPI (left) and anatomical (right) modalities. Landmarks (white crosses) have been manually placed at corresponding locations.

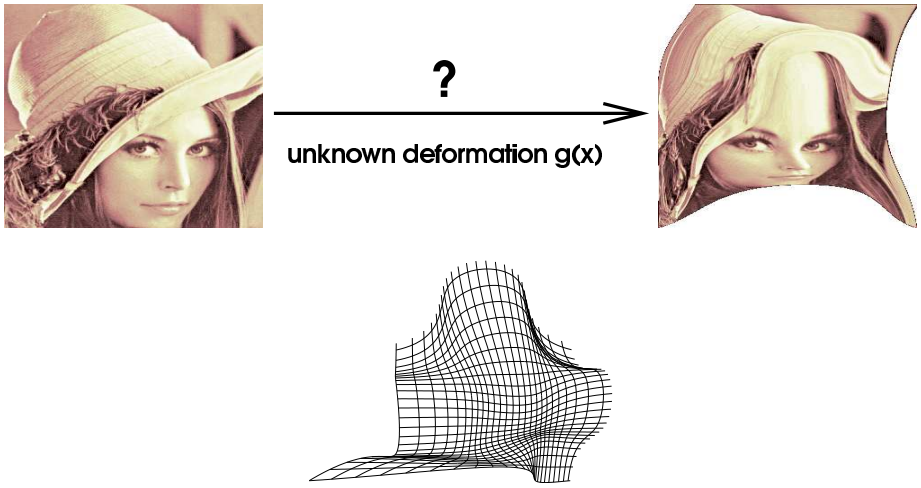


Figure 1.4: Given an image and a deformation function, a deformed image is created by warping. Inversely, given two images, the corresponding deformation function is found by image registration.

and accelerations.

In *video compression*, the movement information enables us to take advantage of the temporal redundancy in the video sequence. This information can be also used for *object tracking* or *image stabilization*.

Separating the image field into several regions according to their speeds leads to *segmentation* algorithms. If the scene is static and the inter-frame changes are caused mainly by the camera movement, the motion field can be used to identify the camera (observer) movement. Knowing the camera trajectory provides the depth information necessary for the *3D reconstruction* of the scene.

When several cameras with known relative positions are available, registering the corresponding frames from sequences by different cameras yields directly the *3D trajectories* of the objects in the scene. A special case is reconstructing depth from *stereo images*.

1.2.1 Biomedical applications

In the biomedical domain, there is a frequent need for comparing images for analysis and diagnostic purposes. However, for an efficient comparison, the images first need to be aligned. This is accomplished by registering the images and warping them using the correspondence function found.

Intra-subject analysis compares images of the same subject taken at different times in order to detect or quantify the changes that might have taken place in-between acquisitions. *Inter-subject* analysis considers corresponding images from different subjects. Aligning and combining images from many subjects permits to create an *atlas*, an annotated reference image of a given modality. Images from individual subjects can then be compared with the atlas for identification and to detect abnormalities. Registering the individual image with the atlas helps us to reuse the segmentation and annotation performed on the atlas. The correspondence function can be used to quantitatively characterize the shape and size of the features in individual images.

Inter-modality registration is used to align images of the same subject taken at approximately the same time but with different imaging modalities. These images provide complementary information which can be combined together to get a more complete picture of the subject anatomy and physiology.

Furthermore, registration helps to compensate for geometrical distortions inherent to some imaging methods, as well as for unwanted motion during the acquisition.

1.3 Contributions of this thesis

Chapter 2 presents an overview and classification of existing registration techniques.

The first contribution of this thesis is a development of a general elastic image registration algorithm. It uses a control grid of uniform B-splines to describe the deformation.

It also uses B-splines for image interpolation. Multiresolution in both image and deformation model spaces yields robustness and speed. One version of this algorithm targeted on finding unidirectional deformation in EPI magnetic resonance images is described in Chapter 4. An enhanced and generalized version of this algorithm which is significantly faster and capable of treating multidimensional deformations is presented in Chapter 5 together with applications in PET and ultrasound image registration.

The second contribution of this thesis is the application of variational techniques to the problem of generalized sampling and reconstruction. In Chapter 3, we describe the landmark interpolation problem which appears in the landmark-based registration techniques. We will see that landmark interpolation can be seen as a special case of a more general class of reconstruction problems. Chapter 6 contains tutorial introduction to the variational approach for solving these problems together with several examples of applications. In Chapter 7, we treat the mathematical details of the theory and discuss the choice of the variational criterion to be used.

Chapter 2

A Review of Registration Techniques

In this chapter we intend to categorize the multitude of existing registration algorithms. We classify them by the choice of *feature space*, *warp space* (search space), *similarity metrics* (cost function), and *search strategy* they use, similarly as in [1]. Most of the algorithms can be cast into this framework. Let us explain the four attributes by creating the following general picture of a registration algorithm:

- In the first step, some intermediate data is extracted from the two images being registered. This data lives in a *feature space*.
- The algorithms' representation of the correspondence between the two images is taken from a *search space*. This is the space in which the algorithm looks for a solution. A point from this space that is returned is the solution of the registration problem.
- To find the solution in the warp space, the algorithm needs a way to measure the quality of the correspondence for different points in the warp space. This measure is provided by a *similarity metric*.
- Finally, the *search strategy* governs the movements of the algorithm in the search space in his quest for the optimum.

Figures 2.1 and 2.2 shows the classification according to the first two criteria in a tree form.

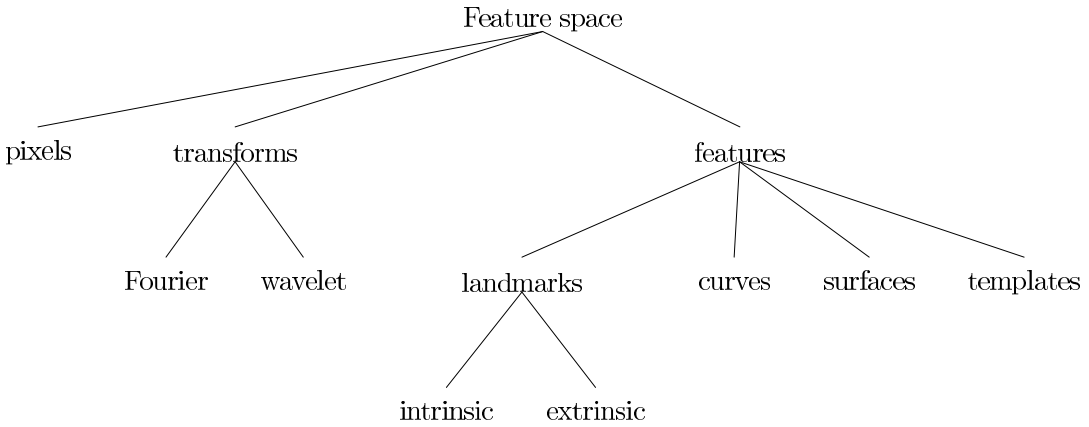


Figure 2.1: Simplified classification of registration algorithms according to the feature spaces used.

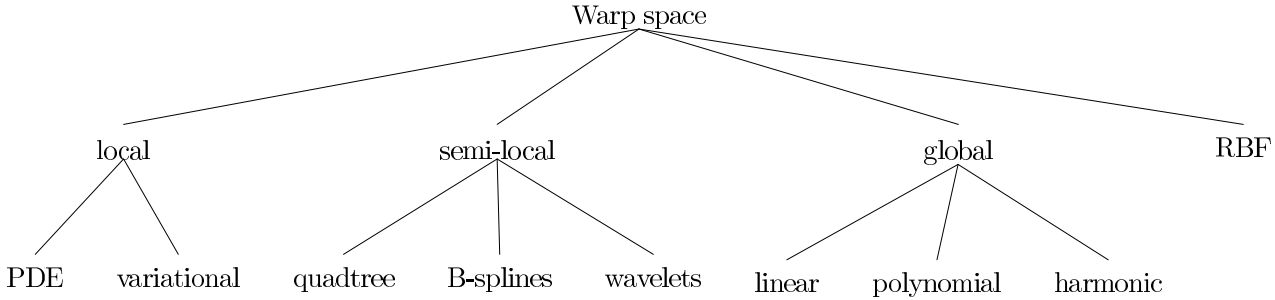


Figure 2.2: Simplified classification of registration algorithms according to the warp space used.

2.1 Feature space

According to the feature space employed, we can identify three classes of registration algorithms: *pixel-based*, *transform-based*, and *feature-based*.

2.1.1 Pixel-based registration

Pixel-based algorithms work directly with the (totality of) pixel values of the images being registered. Preprocessing is often used to suppress the adverse effects of noise and differences in acquisition [2], or to increase or uniformize pixel resolution [3].

It is possible to work directly with the pixel values on the discrete coordinate grid. However, to get a subpixel resolution, the problem is often cast into the continuous framework. The images are considered as functions of real arguments: the image coordinates. The correspondence between the discrete and continuous versions of the image is established using *interpolation*. The crudest method is the nearest-neighbor, and the most often used one is linear (resp. bi- or trilinear) interpolation. Among the high-end methods, *spline interpolation* [4–6] provides the best tradeoff between accuracy and the computational cost [7, 8].

Occasionally, the image model occupies more dimensions than the original data. Example include representing 2D image as a surface in a 3D space, either directly [9], or using level sets [10]. The main advantage of this approach is a more global vision of the algorithm, which increases its robustness.

2.1.2 Transform-based registration

Transform-based algorithms exploit properties of the Fourier, wavelet, Hadamard, and other transforms, making use of the fact that certain deformations manifest themselves more clearly in the transform domain. These methods are used mainly in connection with linear deformation fields. Nevertheless, there are examples of methods that estimate locally linear optical flow using Gabor filters [11, 12] and B-spline wavelets [13]. Typical characteristics of the transforms employed are linearity and independence on the actual image contents.

2.1.3 Feature-based registration

Feature-based algorithms work on a set of characteristic features extracted from the images. The dimensionality of the features is usually drastically smaller than the dimensionality of the original image data. The extraction process is highly non-linear, mostly using thresholding.

Landmark-based methods [14–17] use a relatively small and sparse set of landmarks; these are important points which can be (manually or automatically) identified in both images. *Extrinsic markers* refer to specifically designed artificial features attached to

the object (or subject, in medical imaging) before acquisition to serve as landmarks. Unfortunately, extrinsic markers are difficult to deploy. In medical imaging they are not patient-friendly either. If extrinsic markers are not available, we have to content ourselves with features *intrinsic* to the images. In that case, however, the automatic landmark identification suffers from lack of robustness. The manual landmark identification is often tedious, time-consuming, imprecise, and unreproducible.

If the images cannot be characterized using points, it might be more appropriate to use curves such as edges [18], or volume boundaries [19]. Likewise, in the case of 3D data, surfaces can be used instead of working with the complete volumes.

Popular features are also templates, small sub-images of important regions [20], which can be used directly, or which can form a higher level feature map [21].

2.2 Search space

One of the important factors to categorize registration algorithms is the search space used. We also call it a *warp space*, because it contains warping functions. Warping functions are candidate solutions of the registration problem. From the analogy between warping and deformation, the deformation (warping) functions play also the role of correspondence functions.

Because we work with finite memory computers, every warping function from the search space is described by a finite set of real parameters (from a set of permissible values) by means of a *warping model*. We classify the warping models according to the number of parameters and the spatial extent of the area influenced by a single parameter.

2.2.1 Local models

At one end of the scale, we have *non-parametric, local methods*. The deformation function sought after is basically unconstrained, or belongs to a very large and unrestrictive functional space, e.g., the Sobolev space W_2^2 (of functions that are twice differentiable). We seek the values of this deformation at a very fine grid, usually coinciding with pixel locations. These methods are formulated either as *variational*, defining a scalar criterion to minimize, or (more generally) using partial differential equations (PDE). The continuously defined deformation function minimizes a given criterion, or solves a given PDE. The essence of these methods is thus *entirely in the criterion* (resp., PDE). The PDE come from the optical flow approach (gradient methods) [22], viscous fluid model [23–25], elastic deformations with physical analogs [3, 26] or without it [27].

Sometimes the deformation function is also modeled *indirectly*. For example, it can be modeled using a potential field [28]. This reduces the dimensionality of the problem, at the expense of reduced generality of the deformation.

Discretization allows for integer only displacements at pixel points [29].

2.2.2 Global models

At the other end, we have *parametric, global methods* that describe the correspondence function using a global model with a relatively small number of parameters [30]. The model mostly consists of expressing the warping function in a linear [31], global polynomial [32] or harmonic basis [33, 34]. For these methods, the deformation model corresponding to a specific warp space is as important as the criterion being minimized.

2.2.3 Semi-local model

In between the two extremes are semi-local models, using a moderate number of parameters with local influence. A grid of control points is placed over the image. Their spacing corresponds loosely to knot or landmark density. By changing the spacing, we can approach either of the limit cases or choose a compromise offering the best tradeoff.

Such models were used in the context of motion estimation [35] and are instrumental for the approach described in Chapters 4 and 5 of this thesis. B-spline models were independently used in [36].

2.2.4 Image dependent models

It is sometimes useful to adapt the warping model to the images considered. Quadtree-based deformation model [37] is refined only where it is needed.

In feature based methods, the basis functions of the warping model can be placed where the features are. The deformation field is interpolated in regions where no information is available. Typical example are *radial basis functions* such as thin plate splines [14, 15, 38]. See also Chapter 3 for more details.

2.3 Similarity metrics

2.3.1 Data term

The quality of registration is described by a cost function. This function has a predominant term measuring the quality of the matching which we shall call the *data term*.

For *feature-based* methods it is a mean distance between the corresponding features in the source and target images after warping. Note that the case of *interpolation* with landmarks can be considered as a limit case, when a maximum weight is given to this distance, constraining it to be zero. If the pairing between source and target features is not known, the iterative closest point algorithm [39] can be used to determine it.

For *pixel-based* methods, the data term is a *similarity measure* between the two images after warping. *Correlation*, especially local normalized correlation, is an important cost function [3, 40] because of its probabilistic interpretation. It is rather costly to

evaluate and sensitive to noise. Instead, the similarity between images is most often expressed using their difference in l_1 or l_2 norms. For images from different modalities, the *mutual information* criterion [41–43] seems to be superior [44], at the expense of more computation. For local criteria, such as local normalized correlation, local variance, or local mutual information, the neighborhood size must be properly chosen.

The above mentioned criteria will use *image interpolation* to calculate the warped version of the test image. See also Section 2.1.1.

In *template-based* methods, the template can be compared with a specific region in the target image using any of the similarity measures suitable for pixel-based methods.

For *transform-based* methods, the least-squares measure in the transformed domain is usually used.

2.3.2 Regularization

In most applications, it appears to be necessary to add an additional term to the criterion. We shall call it the *regularization term*. We mainly do it in order to make the problem well-posed. Regularization terms are also used to express our *a priori* knowledge, or eventually to stabilize the algorithm. In the variational setting, the regularization term can define the warping space. For instance, in the case of landmark interpolation, minimization of the norm of the Laplacian $\|\Delta \mathbf{g}\|$ is often used in practice [14, 15]. This leads to a *thin-plate spline* solution [45]. Minimizing other similar measures leads to generalized splines [46] determined either directly or using PDEs.

Other regularizers are constructed by applying a non-linear term on the norm of the derivative operator, mainly to preserve discontinuities. Regularizers based on thresholding in wavelet domain are also used.

Implicit regularization for iterative methods works by alternatively driving the intermediate solution towards the data, and applying a smoothing operator.

2.4 Search strategy

2.4.1 Direct solution

In some cases, notably if the cost function is quadratic, the solution can be found in one step [33]. This is equivalent to neglecting the higher order terms in the Taylor expansion of the criterion [47]. For example, the transform methods using directional filters are often engineered in this way.

2.4.2 Exhaustive search

If the search space is finite, exhaustive search can be used. For example when small templates are extracted from the reference image, their positions in the target image are

found by trying all possible shifts [29, 48].

2.4.3 Dynamic programming

For integer 1D problems, *dynamic programming* is applicable [49] with complexity proportional to the number of decisions to take (size of the image) and the number of possible outcomes (shifts).

2.4.4 Partial differential equations

In this approach, a system of partial differential equation describing an evolution in time of the deformation field is formed, such that its final value corresponds to the desired solution. The PDE parameters can vary with time, to enforce robustness at first and relaxing the constraints towards the end, to allow for precise registration. Various numerical methods can be used to solve the PDEs, the principal ones being finite difference relaxation method [50], finite elements method (FEM) [51, 52], hierarchical finite element bases [53], multigrid methods [54], and wavelets [55, 56].

2.4.5 Multidimensional optimization methods

Many non-linear registration methods lead to a non-linear optimization problem. Various optimization methods are used, depending on the size and structure of the problem. The most popular choices include the Powell method, gradient descent, conjugated gradients, and variations of the Newton method, such as the Marquardt-Levenberg algorithm [50, 57].

Some minimization algorithms can be described using different paradigms, such as 'demons' [58].

2.4.6 Multiresolution

Multiresolution [3, 31] on the feature space (usually image size) helps to speed up the process and to increase its robustness by approaching the solution by gradual refinement. One first solves a reduced problem using a small amount of data, then uses the solution as an initial guess for a problem at a finer level. This is repeated until the finest (original) level is reached.

Multiresolution on the search space works similarly, adding degrees of freedom to the warping model at each step. One starts with a simple model which leads to a simple and easy to solve problem, and then gradually adds a manageable amount of complexity at each step, until the desired model is reached. The model can be augmented qualitatively, such as going from translation-only to general affine transform, or quantitatively, for example by decreasing the control node spacing in semi-local models.

Related to multiresolution are multigrid optimization methods, where occasional backward transitions from finer to coarser levels are used besides the coarse-to-fine refinement used in the multiresolution.

2.5 Other attributes

The *dimensionality* refers to the number of dimensions of the images being registered. The warping function normally works in the space of the same dimensionality, transforming one coordinate vector into another. However, degenerate warping functions can be expressed in a space of lower dimensionality.

Interactive algorithms need human supervision and interaction. They often perform well, taking advantage of the human expert, but are unsuitable for treating high volumes of data.

2.6 Complementary surveys

We conclude the chapter by providing additional general sources of information on registration algorithms. The survey by Brown [1] is rather general, while Warfield *et al.* [59] concentrate on nonlinear registration for brain warping applications. Bayesian interpretation of elastic matching are reviewed by Gee [60], also in the context of human neuroanatomy. The last two survey articles we mention deal specifically with medical imaging applications of image registration. An article by Van den Elsen *et al.* [61] contains very comprehensive and detailed classification of available methods. Finally, Lester and Arridge [62] emphasise the hierarchical concepts of the algorithms.

Chapter 3

Landmark Registration: From Thin-Plate Splines to Fractals

In this chapter we briefly present the landmark registration problem and discuss its motivation. We suggest the variational formulation and hint at its solution. The topic of variational reconstruction in general is treated in more breadth and mathematical rigour in Chapters 6 and 7. Here, in contrast, we put the emphasis on simplicity and clarity, favoring intuitive understanding of the technique.

The first part of this chapter has been conceived as a tutorial on thin-plate spline interpolation and the related method of radial basis functions (RBF) because these are often described superficially in the engineering literature, and perhaps too abstractly in the mathematical literature. In the second part, we discuss extensions of the classical landmark registration method of Bookstein [14] to non-integer orders of the criteria and present a Bayesian interpretation in terms of fractal priors. We also take the opportunity to make the link between radial basis functions and splines very explicit (see also [63]).

3.1 Landmark registration

Landmark registration [14–16] is a two step feature-based registration technique. (See Chapter 2 for details on registration algorithm classification.) In the first step, a set of landmark pairs is identified. We get two sequences of points, $\mathbf{x}_1, \ldots, \mathbf{x}_N$, and $\mathbf{z}_1, \ldots, \mathbf{z}_N$, such that an object at coordinates $\mathbf{x}_i$ in the reference image corresponds to the object at coordinates $\mathbf{z}_i$ in the test image.

Here we assume that the landmarks have been already determined. For an example,

see Figure 1.3 in Chapter 1. In most cases, this is done manually. Some authors have proposed to detect the landmarks automatically [16,64], but this approach has not been pursued here. Instead, if a completely automatic method is needed, we switch to a more robust intensity-based aalgorithm, described in Chapters 4 and 5.

Once the landmarks have been determined, we need to interpolate the correspondence function values between the landmark points. This is because, as we have mentioned in Chapter 1, the aim of image registration is to get the correspondence function everywhere, not just at a few points. Interpolating the correspondence function is the second crucial step of landmark registration and the main focus of this chapter [65,66].

3.1.1 Motivation of this study

There are several reasons to study landmark registration. The first of them is the comparative testing of other registration methods. It is usually possible to set up a completely automatic test procedure where the results of the algorithm under test are compared with a known ground truth. We will see examples of such tests in Chapter 4. Nevertheless, these tests fail to capture the real conditions in which the algorithm will work. On the other hand, using landmark registration with manually selected landmarks permits us to compare the performance of the automatic algorithm with a reference (landmark registration) on real images and under realistic working conditions.

Second, landmark interpolation merits a study in its own right. Choosing an interpolation method or an interpolation function is difficult because the implications of this choice are not immediately apparent. On the other hand, in the variational formulation, the user is asked instead to choose a criterion of optimality, which is usually more tangible and often related to the physics (or other specificities) of the problem.

Third, the variational formulation of landmark interpolation will allow us to make an interesting link with splines [67]. This gives additional justification for the choice of B-splines as basis functions for our automatic registration method described in Chapters 4 and 5.

3.2 Interpolation versus approximation

We can distinguish two classes of constraints in landmark interpolation: *hard* constraints which need to be fulfilled exactly and *soft* constraints which are meant to be fulfilled only approximatively, allowing for the tradeoff with the smoothness of the solution.

3.2.1 Hard constraints

The obvious choice is to enforce the interpolating constraints. The task is to find a continuously-defined correspondence function $\mathbf{g}$. At the landmark position $\mathbf{x}_i$, the

correspondence function must coincide with the landmark in the test image; i.e.,

$$\mathbf{g}(\mathbf{x}_i) = \begin{bmatrix} g_x(\mathbf{x}_i) \\ g_y(\mathbf{x}_i) \end{bmatrix} = \mathbf{z}_i \quad \text{for all } i \in \{1, \ldots, N\} \tag{3.1}$$

3.2.2 Soft constraints

Alternatively, if the landmark positions are noisy or not accurately known, we can replace the interpolation condition (3.1) by an *approximation* inequality. We will only require that the correspondence function $\mathbf{g}$ passes close enough to the landmarks. For example, we might impose the inequality:

$$\sum_{i=1}^{N} ||\mathbf{g}(\mathbf{x}_i) - \mathbf{z}_i||^2 \leq \varepsilon \tag{3.2}$$

where ε is an *a priori* given error bound. Clearly, when ε tends to zero, the approximation problem (3.2) becomes equivalent to the interpolation defined by (3.1).

Although here we will mostly deal with the interpolation case, we will learn later (see Chapters 6 and 7) that most of the properties are directly applicable to the approximation case too, and the formula of the solution is also very similar.

3.2.3 Desirable properties

We agree that landmarks are points in space, as opposed to just coordinate values. Similarly, the correspondence function $\mathbf{g}$ is more than a mathematical function: it describes correspondence of real points. It is an object in space, anchored to the landmarks. Consequently, it seems reasonable to require that the interpolated function $\mathbf{g}$ be invariant with respect to the choice of the coordinate system. In other words, the correspondence between points in the two images should remain the same, regardless of how we measure the position of these points.

A key requirement is that the interpolation problem should always have a solution, if possible a unique one.

Another property worth having is the reproduction of identity [68]. If the underlying true correspondence function is identity, the interpolation should yield identity correspondence function with a minimum of landmarks. In addition, we might want the reproduction property for other simple transformations, such as shifts or scalings; more generally, affine transformations.

We want the reconstructed correspondence function to be close to the (unknown) true underlying correspondence function. We want the reconstruction error to decrease rapidly with the number of landmarks. This way we can adapt the landmark density to insure that the error is below any *a priori* given tolerance threshold. These are called good approximation properties [69].

Finally, we want the interpolation procedure to accommodate easily non-exact fits, useful when the landmark positions are only known approximately. In this approximation setting, the reconstructed correspondence function will pass close to the landmarks, making a compromise between the closeness of the fit and the overall smoothness.

3.3 Thin-plate splines

The use of thin-plate spline technique for landmark interpolation is attributed to Bookstein [14]. Here we present the method from the variational point of view, as a preparation for the extensions presented in Section 3.4.

3.3.1 Variational formulation

Instead of imposing an empirical interpolation formula, the essence of the variational formulation consists of choosing a variational criterion $J(\mathbf{g})$ and then finding among all possible functions satisfying (3.1) the one that minimizes J [70, 71].

It is customary to interpret J as strain energy and to take a physical model of a thin steel plate to calculate this energy [17]. When the scalar field g is interpreted as a small vertical displacement, the deformation (strain) energy is given by

$$J(g) = \int \left(\frac{\partial^2 g}{\partial x^2}\right)^2 + 2\left(\frac{\partial^2 g}{\partial x \partial y}\right)^2 + \left(\frac{\partial^2 g}{\partial y^2}\right)^2 \mathrm{d}x\mathrm{d}y \tag{3.3}$$

Under specific conditions on the solution space (cf. Chapter 7) we can use integration by parts to come up with an equivalent criterion

$$J(g) = \int \left(\nabla^2 g\right)^2 \mathrm{d}x\mathrm{d}y \tag{3.4}$$

where ∇^2 denotes the Laplacian. For simplicity, we will neglect the difference between (3.3) and (3.4) here.

The Laplacian energy (3.4) is a member of a more general family of scale and rotation invariant cost functions which satisfy the requirements of Section 3.2.3 (see also Chapter 7). Another more pragmatic justification comes from the fact that it is the simplest criterion that does not penalize affine transforms.

The criterion for the vector form $\mathbf{g}$ is taken simply as the sum of the strain energies of the x and y components:

$$J(\mathbf{g}) = J(g_x) + J(g_y) \tag{3.5}$$

Clearly, the constraints (3.1) can be broken into two independent sets, one for g_x and one for g_y. Therefore, the task of minimizing (3.5) under the complete constraints (3.1) is equivalent to minimizing separately for g_x and g_y. Consequently, to get an understanding of the underlying interpolation method, we can concentrate on the scalar case here.

3.3.2 Interpolation formula

As we will see in Chapter 6, the correspondence function $g(x, y)$ minimizing (3.3) under interpolation constraints

$$g(x_i, y_i) = z_i \tag{3.6}$$

is given by

$$\begin{aligned} g(x,y) &= \sum_{i=1}^{N} \lambda_i \varrho(\|\mathbf{x} - \mathbf{x}_i\|) + a_0 x + a_1 y + a_2 \\ \text{with} \quad \|\mathbf{x} - \mathbf{x}_i\| &= \sqrt{(x - x_i)^2 + (y - y_i)^2} = r \end{aligned} \tag{3.7}$$

where $\varrho(r)$ is a *radial basis function*

$$\varrho(r) = r^2 \log r \tag{3.8}$$

It is called radial because it only depends on the Euclidean distance r to its associated data point [72].

The generating function $\varrho(x, y)$ solves the associated Euler-Lagrange (or fundamental) equation

$$\nabla^4_{x,y}\, \varrho(\sqrt{x^2 + y^2}) = \delta(x, y) \tag{3.9}$$

where ∇^4 is a two times iterated Laplacian and $\delta(x, y)$ is the Dirac distribution. The linear polynomial $a_0 x + a_1 y + a_2$ is called a *kernel term* and it appears because it does not contribute to the criterion. The unknown parameters λ_i and a_0, a_1, a_2 are determined from the interpolation constraints (3.6) and from orthogonality conditions

$$\sum_{i=1}^{N} \lambda_i = 0 \qquad \sum_{i=1}^{N} \lambda_i x_i = 0 \qquad \sum_{i=1}^{N} \lambda_i y_i = 0 \tag{3.10}$$

The method just described is called *thin-plate spline* interpolation.

Let us briefly return to the approximation case, minimizing the criterion J under the scalar variant of the inequality (3.2). Here also, the unknowns can be determined from a set of linear equations composed of the orthogonality conditions (3.10) and the following regularized version of the interpolation constraints (3.6)

$$g(x_i, y_i) + \gamma^{-1} \lambda_i = z_i \tag{3.11}$$

where γ is a constant depending on ε in (3.2).

3.3.3 Solution justified

Let us hint at the proof that (3.7) is really a solution. We will study the effect of how the criterion $J(g)$ is affected by adding to g a small perturbation ψ. For simplicity, we use the Laplacian form (3.4) of the criterion. The perturbation is chosen so that $\psi(x_i, y_i) = 0$, in order not to disturb the interpolation conditions. Furthermore, we choose ψ to vanish identically at infinity and to be sufficiently differentiable; in other words, ψ is going to be a *test function*. If g is really a minimum, then the variation

$$\widetilde{J} = J(g + \psi) - J(g) \tag{3.12}$$

must be zero to a first order of approximation. The change is given by

$$\widetilde{J} = 2 \int \nabla^2 g \nabla^2 \psi \, \mathrm{d}x\mathrm{d}y = 2 \sum_{i=1}^{N} \lambda_i \int \nabla^2 \varrho(r) \nabla^2 \psi \, \mathrm{d}x\mathrm{d}y \tag{3.13}$$

Note that we have left our the polynomial part of the solution because it does not contribute to the criterion. Using the identity $\int \nabla^2 f \, \nabla^2 \psi \, \mathrm{d}\mathbf{x} = \int \nabla^4 f \, \psi \, \mathrm{d}\mathbf{x}$, equation (3.9), and the translational invariance of ∇, we rewrite the integral as

$$\int \nabla^4 \varrho(r) \, \psi \, \mathrm{d}x\mathrm{d}y = \int \delta(x - x_i, y - y_i) \, \psi \, \mathrm{d}x\mathrm{d}y = \psi(x_i, y_i) = 0 \tag{3.14}$$

We conclude that the integral and therefore the first order change $\widetilde{J}$ are both zero. Consequently, the formulas (3.7) and (3.8) give the solution to the minimization problem.

3.4 Fractional splines

Although the thin-plate splines have been known to work well, in many applications we might benefit from a wider choice of interpolation functions, while keeping the general spirit and the invariance properties (affine geometrical transformations including scaling) we are interested in. The straightforward way to do it is to consider minimizing different criteria, namely fractional derivatives (in 1D) and fractional Laplacian (in multiple dimensions).

3.4.1 Fractional derivatives

The Laplacian in the space domain is defined by

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \tag{3.15}$$

In the Fourier domain, this corresponds to

$$\widehat{\nabla^2 f} = \omega_x^2 \hat{f} + \omega_y^2 \hat{f} = \|\boldsymbol{\omega}\|^2 \hat{f} \tag{3.16}$$

Similarly, a twice-iterated Laplacian will (through the convolution property $\widehat{f * g} = \hat{f}\,\hat{g}$) correspond to

$$\widehat{\nabla^4 f} = \|\boldsymbol{\omega}\|^4 \hat{f} \tag{3.17}$$

A natural extension of (3.16) is to consider other exponents of $\|\boldsymbol{\omega}\|$, which leads to the *fractional Laplacian*

$$\widehat{\nabla^\alpha f} = \|\boldsymbol{\omega}\|^\alpha \hat{f} \tag{3.18}$$

The same reasoning in the univariate case leads to the concept of fractional derivatives, studied by Liouville [73]. The basic form of fractional derivative corresponds to a multiplication by $(j\omega)^\alpha$ in the Fourier domain

$$\widehat{\frac{\partial^\alpha f}{\partial x}} = (j\omega)^\alpha \hat{f} \tag{3.19}$$

There is also a symmetric version, in the spirit of (3.18), where multiplication by $(j\omega)^\alpha$ is replaced by multiplication by $|j\omega|^\alpha$, see [74].

Note that the behavior of fractional derivatives for nonintegral orders α is significantly different from clasical derivatives. In particular, they are no longer local operators [74].

3.4.2 The fractional criterion

In line with the definition (3.18), it is possible to generalize the Laplacian based criterion (3.4) to fractional orders as follows.

$$J(g) = \int \|\nabla^\alpha g(\mathbf{x})\|^2 \mathrm{d}\mathbf{x} \propto \int \|\boldsymbol{\omega}\|^{2\alpha} |\hat{g}(\boldsymbol{\omega})|^2 \mathrm{d}\boldsymbol{\omega} \tag{3.20}$$

Note that this extended criterion remains scale and translation invariant. To appreciate better the meaning of (3.20), we look at the corresponding expression for the univariate case

$$J(g) = \int \left(\frac{\partial^\alpha g}{\partial x}\right)^2 \mathrm{d}x \propto \int |\omega|^{2\alpha} \, |\hat{g}(\omega)|^2 \, \mathrm{d}\omega \tag{3.21}$$

We see that the criterion measures the amplitude of the α-th derivative of g.

3.4.3 Fractional Brownian motion

There is an interesting relationship between fractional Brownian motion and the fractional derivatives just described.

Brownian motion is a stochastic process $B(t)$. Its particular characteristic is that its increment $B(t+\tau) - B(t)$ is stationary Gaussian noise with variance

$$\mathrm{E}\left[\left(B(t+\tau) - B(t)\right)^2\right] \propto |\tau| \tag{3.22}$$

which corresponds to an average power spectrum proportional to $|\omega|^{-2}$.

Fractional Brownian motion [75] is a generalization to a family of processes $B_H(t)$ described by a single parameter $0 < H < 1$. The covariance structure of $B_H(t)$ is

$$\mathrm{E}\left[B_H(t)\, B_H(s)\right] \propto |t|^{2H} + |s|^{2H} - |t-s|^{2H} \tag{3.23}$$

Consequently, its increment is also stationary and has a variance

$$\mathrm{E}\left[\left(B_H(t+\tau) - B_H(t)\right)^2\right] \propto |\tau|^{2H} \tag{3.24}$$

with an average spectrum proportional to $|\omega|^{-2H-1}$.

Consider applying the derivative (3.19) to the process B_H described by (3.23). The covariance function of the resulting process $\frac{\partial^\alpha}{\partial t^\alpha} B_H(t)$, which is stationary, becomes

$$\mathrm{E}\left[\frac{\partial^\alpha}{\partial t^\alpha} B_H(t)\, \frac{\partial^\alpha}{\partial s^\alpha} B_H(s)\right] = -\frac{\partial^\alpha}{\partial t^\alpha}\frac{\partial^\alpha}{\partial s^\alpha}|t-s|^{2H} = \rho(t-s) \tag{3.25}$$

In the Fourier domain, we get

$$\hat{\rho}(\omega) \propto \frac{|j\omega|^{2\alpha}}{|\omega|^{2H+1}} \tag{3.26}$$

In particular, taking a fractional derivative of order $\alpha = H + \frac{1}{2}$ of a fractional Brownian motion process B_H gives $\hat{\rho}(\omega) = 1$, and thus effectively yields an uncorrelated Gaussian white noise.

3.4.4 Bayesian interpretation

When $\mathbf{x}$ is a random vector with Gaussian probability density function (p.d.f.) $p_{\mathbf{x}} \propto \mathrm{e}^{-\mathbf{x}^T \mathsf{C}^{-1} \mathbf{x}}$, then maximizing its log-likelihood function ($\log p_{\mathbf{x}} \propto -\mathbf{x}^T \mathsf{C}^{-1} \mathbf{x}$) under some constraints is equivalent to finding the most likely solution to our problem given the *a priori* knowledge that $\mathbf{x}$ is normally distributed. Equivalently, we can factorize $\mathsf{C}^{-1} = \mathsf{Q}^T \mathsf{Q}$ which gives an alternative expression for the log-likelihood, $\log p_{\mathbf{x}} \propto -\|\mathsf{Q}\mathbf{x}\|$. The

technique is called *whitening*, because the new vector $\mathbf{Qx}$ is white, that is to say that its components are uncorrelated (independent).

Similarly, if we assume that $g(x)$ is fractal Brownian motion-like process, then we can apply to it the whitening filter $\frac{\partial^\alpha}{\partial x^\alpha}$, as we have shown in the previous section. Therefore, the criterion (3.21) can be interpreted as a Bayesian *fractal prior*. Some comments on Bayesian priors in the non-fractal case can be found in Poggio [76]. We assume that the underlying true function is close to the fractional Brownian motion model and find the solution to our interpolation problem combining this knowledge with the information given by the constraints.

Completely analogous reasoning justifies the choice of the criterion (3.20) in the bivariate case.

Note that even though we have started from fractional Brownian motion to derive the criterion, the orders α higher than the limit of 1.5 corresponding to the fractal Brownian motion are still completely meaningful as parameters for our deformation model. (For $\alpha < 0.5$ the resulting interpolation is point-wise unstable and has singularities.) The α translates into the assumed smoothness of the deformation and the best way of measuring it. The higher the α, the smoother the deformation.

3.4.5 Fractional landmark interpolation

The Euler-Lagrange equation corresponding to (3.20) is $\nabla^{2\alpha}\varrho = \delta$. The solution for non-special α (read: non-integer) is of the form

$$g = \sum_{i=1}^{N} \lambda_i \varrho(\underbrace{||\mathbf{x} - \mathbf{x}_i||}_{r}) \tag{3.27}$$

where

$$\varrho(r) = r^{2\alpha-2} \tag{3.28}$$

The polynomial kernel term does not appear due to the restrictions of the Fourier domain definition of the criterion (3.20).

The choice of α has obviously an influence on the interpolation results. Here, we present an example of this effect for the landmark (bivariate) case. We refer the reader to Chapter 6 for some additional examples in the univariate case.

Figure 3.1 demonstrates how the deformation results change with α. We have chosen two images from a four-chamber ultrasound sequence of a heart[1] and declared one of them reference and the other a test. We have manually identified six pairs of conforming

[1] *Acknowledgements:* Images and landmark placement are the courtesy of María J. Ledesma, Universidad Politécnica de Madrid, and Laboratory of Echocardiography, Hospital General Universitario Gregorio Marañón, Madrid, Spain.

points in both images. We have also put stable landmarks in the corners of the image. Then we warped the test image onto the reference image varying the parameter α. Note that only non-integer values of α were used, for which the formula (3.27) remains valid; integer values need a special treatment.

How do we choose the best α? As we have seen in the previous section, the statistically optimal α can be determined directly, when the characteristics of the stochastic process generating the deformation are known. However, this is never the case in practice. Therefore, α must be found experimentally. We observe that small α yields more localized and abrupt changes in the deformation field, while higher α gives rise to smoother and more global changes.

When a sufficient number of test images and landmarks are available, a suitable α for a given application can be determined by the *leave-one-out* technique: One or several landmarks are not taken into consideration when calculating the correspondence function. Their real position is then compared to their position predicted by the interpolation. Finally, the α yielding the smallest average error is selected. In the present case, we found the values of $\alpha = 1.5 \sim 2.5$ to be the most suitable.

3.5 Localization

We have seen that the basis function given by our interpolation formulas are typically polynomially growing towards infinity. We will now show how to replace them by an equivalent set of compactly supported or fastly decaying basis functions. The process is called *localization*.

The motivation of localization is threefold. First, a function expressed with compactly supported or fastly decaying basis functions is computationally less expensive to evaluate. Second, due to the smaller amplitude range of the basis functions the evaluation is more precise. Third, thanks to smaller overlap between basis functions, the coefficients are less interdependent and the corresponding linear system of equations is better conditioned. Localization corresponds to a change of basis and in this particular context it is a powerful preconditioning method.

3.5.1 Localizing $|x|^3$

Consider the task of univariate scalar interpolation minimizing the norm of the second derivative ($\int (f'')^2 \mathrm{d}x$) with uniform sampling. The Euler-Lagrange equation is

$$\frac{\partial^4}{\partial x^4} \varrho(x) = \delta \tag{3.29}$$

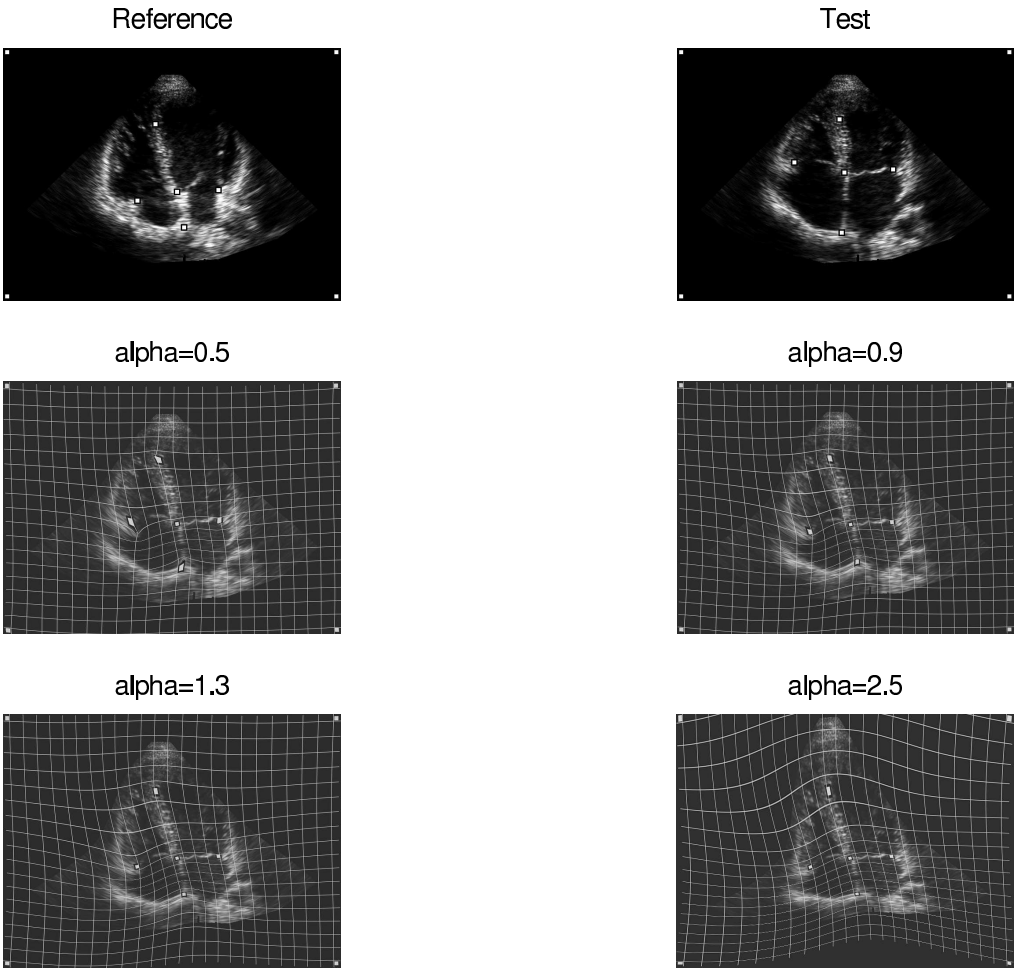


Figure 3.1: The reference *(top left)* and test *(top right)* images. The test image warped by landmark warping for $\alpha = 0.5$ *(middle left)*, $\alpha = 0.9$ *(middle right)*, $\alpha = 1.3$ *(bottom left)*, and $\alpha = 2.5$ *(bottom right)*. The landmark positions are marked with white squares and were identical in all cases.

One of its solutions corresponds to $\varrho(x) = |x|^3$ which yields the interpolation formula

$$f(x) = \sum_i \lambda_i |x - i|^3 + a_0 + a_1 x \tag{3.30}$$

with infinitely growing basis functions ϱ. We see that between each two integers, $f(x)$ is a cubic polynomial. It is also evident that its second derivative f'' is continuous everywhere. Therefore, by definition, $f(x)$ is a *cubic spline*. We know that a cubic spline can be expressed as a linear combination of compactly supported B-splines β_3 [77]. How do we go from the basis functions ϱ to basis functions β_3? By localization.

We note that β_3 can be expressed as the following linear combination of the shifted basis functions ϱ:

$$\beta_3(x) = \frac{\Delta^4}{4!}|x - k|^3 = \frac{\begin{bmatrix} 1 & -4 & \underline{6} & -4 & 1 \end{bmatrix}}{12} * |x - k|^3 \tag{3.31}$$

where Δ^4 is the four times iterated finite difference operator. The original and localized basis functions are shown in Figure 3.2. An interesting observation is that the localizing operator Δ^4 is a discretized version of the continuous operator $\frac{\partial^4}{\partial x^4}$ from (3.29). This is no coincidence as, by virtue of the discretization, the low-frequency parts of the two operators behave in the same way:

$$\mathscr{F}\left\{\frac{\partial^4}{\partial x^4}\right\} = \omega^4 \tag{3.32}$$

$$\mathscr{F}\left\{\frac{\Delta^4}{4!}\right\} = \frac{\left(1 - \mathrm{e}^{j\omega}\right)^4}{4!\,\mathrm{e}^{2j\omega}} \underset{\omega \to 0}{\sim} \omega^4 \tag{3.33}$$

Therefore, the result of the localization (3.31) is in some sense a low-frequency version of the Dirac in (3.29).

In the case of non-uniform sampling, the identical basis functions ϱ are put on the non-uniform sampling points:

$$f(x) = \sum_i \lambda_i |x - x_i|^3 + a_0 + a_1 x \tag{3.34}$$

The corresponding new set of basis functions can also be localized using the fourth *divided difference* operator [77] instead of the finite difference operator. The divided difference operator is the discrete equivalent (discretization) of the fourth derivative on the non-uniform grid. It provides the highest order coefficient of the fourth-order polynomial passing through given five points.

3.5.2 Localizing general variational problem solution

Further development of this idea shows that when solving the interpolation problem minimizing a norm $||\mathsf{L}f||$ of some operator L, the corresponding fundamental solution

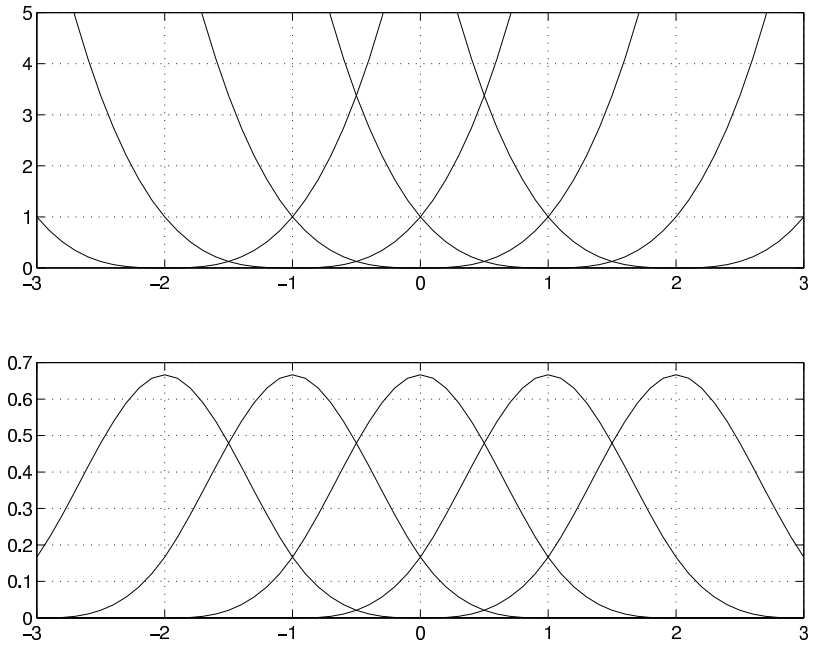


Figure 3.2: The original $|x|^3$ basis functions *(top)* and their localized version, cubic B-splines β_3 *(bottom)*

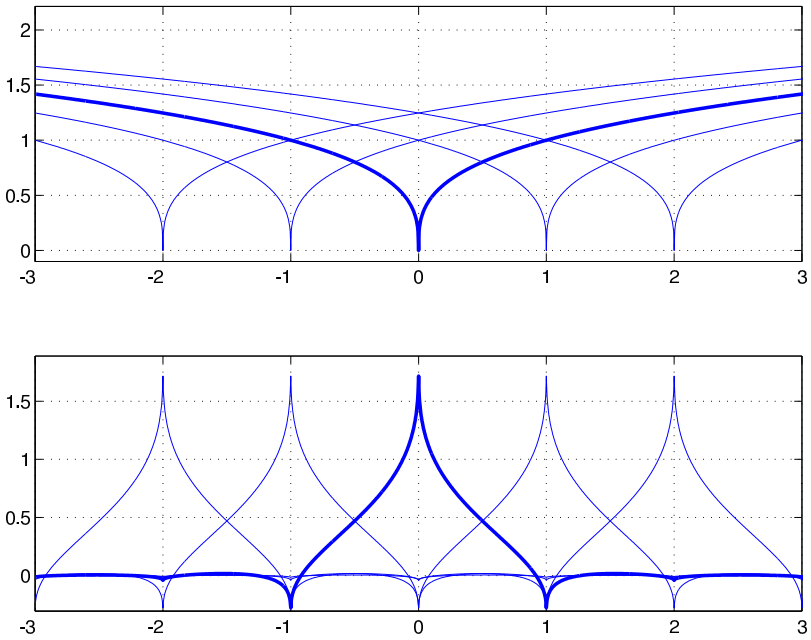


Figure 3.3: The original $|x|^{1/\pi}$ basis functions *(top)* and their localized version, fractional symmetric B-splines $\beta^*_{1/\pi}$ *(bottom)*

(Green function) ϱ satisfies the Euler-Lagrange equation $\mathsf{L}^T\mathsf{L}\,\varrho = \delta$. Then the discretized version of the same operator $\mathsf{L}^T\mathsf{L}$ can be used to localize the basis functions ϱ.

Of course, the discretization is straightforward only when the sampling points are distributed on a uniform grid. On a nonuniform grid, each basis function has to be localized separately.

For example, interpolation in the univariate scalar case minimizing the norm of a fractal derivative (3.21) leads to basis functions $|x - x|^{2\alpha-1}$ which can be localized by fractional symmetric finite difference operator $\Delta_*^{2\alpha}$ to yield rapidly decaying fractional B-splines $\beta^*_{2\alpha-1}$ [74], see Figure 3.3.

Along similar lines, in the bivariate case, the radial basis function $r^2 \log r$ can be localized by the discretized version of the twice iterated Laplacian ∇^4. The result of

$$\left\{\mathbf{h}\,\mathbf{h}^T\right\}_{j,k} * \varphi(x-j, y-k) \quad \text{with} \quad \varphi(x,y) = \frac{x^2+y^2}{2}\log(x^2+y^2) \tag{3.35}$$

and where $\mathbf{h}$ is the four-times iterated finite difference operator

$$\mathbf{h} = \Delta^4 = \begin{bmatrix} 1 & -4 & \underline{6} & -4 & 1 \end{bmatrix} \tag{3.36}$$

is shown in Figure 3.4.

Note that other localizations are indeed possible, depending on the precise definition of localization.

3.6 Conclusions

We have presented the landmark registration technique with focus on the second step, the problem of landmark interpolation. This problem can be formulated very concisely in the variational setting. We choose the variational criterion to impose useful properties on the interpolation process, such as rotational, translational, and scale invariance. Most notably, when the criterion is quadratic, the solution is expressed as a linear combination of translated generating (Green) functions. The coefficients of this linear combination are determined from a linear system of equations.

The *a priori* non-local generating functions can be localized for more efficient and more stable calculation. In some cases this localization leads to B-splines which gives an additional justification for using splines to solve this kind of problems.

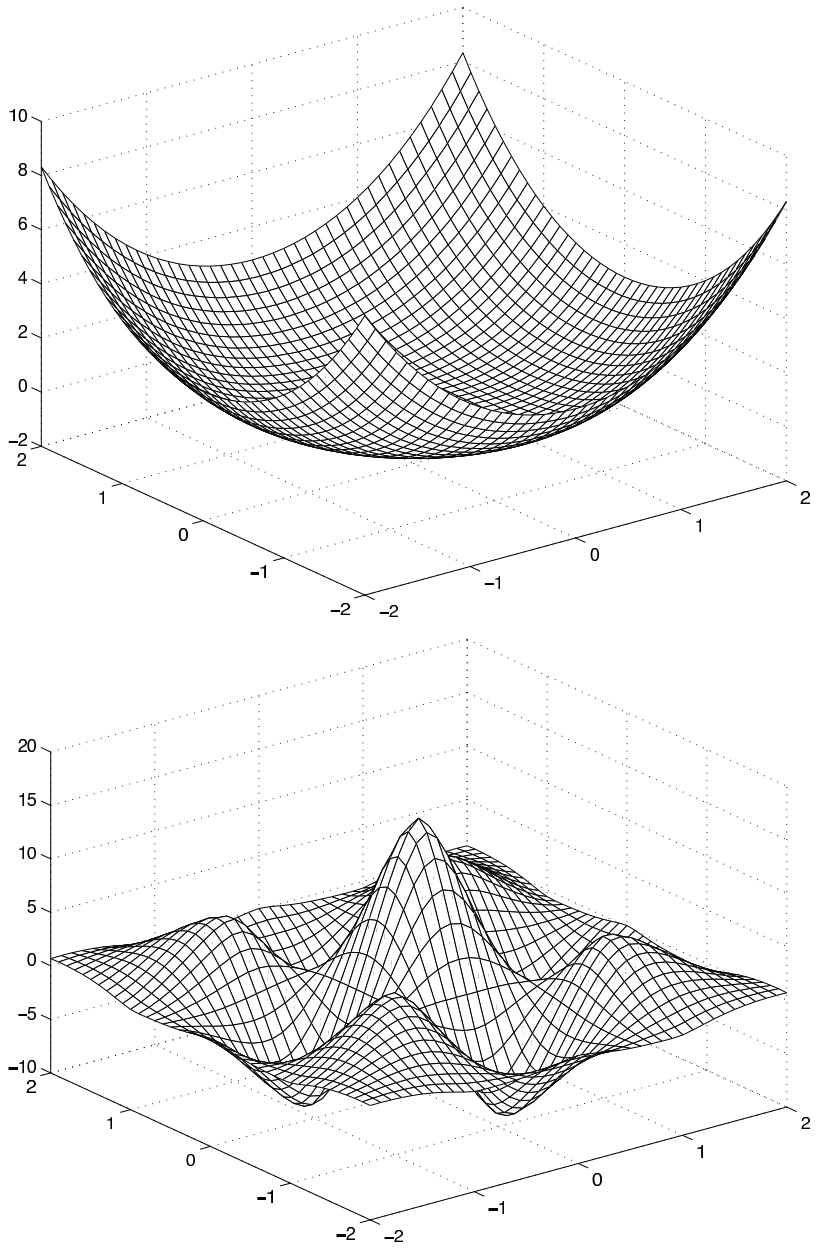


Figure 3.4: The $r^2 \log r$ basis function *(top)* and its localized version *(bottom)*.

Chapter 4

Unwarping of Unidirectionally Distorted EPI Images

This chapter is based on our article [78].

4.1 Abstract

Echo-planar imaging (EPI) is a fast nuclear magnetic resonance imaging method. Unfortunately, local magnetic field inhomogeneities induced mainly by the subject's presence cause significant geometrical distortion, predominantly along the phase-encoding direction, which must be undone to allow for meaningful further processing. So far, this aspect has been too often neglected.

In this paper, we suggest a new approach using an algorithm specifically developed for the automatic registration of distorted EPI images with corresponding anatomically correct MRI images. We model the deformation field with splines, which gives us a lot of flexibility while comprising the affine transform as a special case. The registration criterion is least-squares. Interestingly, the complexity of its evaluation does not depend on the resolution of the control grid. The spline model gives us good accuracy thanks to its high approximation order. The short support of splines leads to a fast algorithm. A multiresolution approach yields robustness and additional speed-up.

The algorithm was tested on real as well as synthetic data, and the results were compared with a manual method. A wavelet-based Sobolev-type random deformation generator was developed for testing purposes. A blind test indicates that the proposed automatic method is faster, more reliable, and more precise than the manual one.

Keywords: image registration, splines, geometrical distortion, unwarping

4.2 Introduction

4.2.1 EPI features

Echo planar imaging (EPI) [79] is a fast magnetic resonance imaging (MRI) technique permitting an acquisition of a two-dimensional slice using a single excitation, which leads to very short scan times. It is used mainly for functional imaging (fMRI), the *in vivo* non-invasive study of the temporal, spatial and behavioral dependencies of brain activations. The basis of fMRI lies in the fact that deoxyhemoglobin (the hemoglobin without a bound oxygen molecule) is paramagnetic. Neural activation in the cerebral cortex leads to an increase of blood flow, hence to a decrease of deoxyhemoglobin concentration.[1] This results in a measurable alteration of the magnetic field and in a consequent increase of signal intensity in the appropriately weighted MRI images (blood oxygen-level dependent, BOLD). It is therefore difficult to compensate for the unwanted magnetic field inhomogeneities induced mainly by the spatially varying magnetic susceptibility of the subject [80]. In contrast to conventional MRI, where the number of excitations per slice is equal to the number of scan lines, in EPI the magnetic field gradients have to encode two coordinates simultaneously in one excitation. As one of the gradients (the so-called phase-encoding gradient) is several orders of magnitude weaker than the other, the inhomogeneous magnetic field will manifest itself mainly as a geometrical distortion of the 2D slice image along the direction of this gradient. This effect is clearly visible in Figure 4.1. Since the stronger gradient is less affected, the distortion is essentially unidirectional. Letting g be the unknown warping (deformation) function, we have

$$f^o(g(x,y),y) \simeq f^u(x,y) \tag{4.1}$$

where f^o is the observed EPI image and f^u is the hypothetical ideal undistorted EPI image. We can consider each slice separately, as the shift in the z axis due to patient's movement is insignificant because his head is attached. Should there be such a displacement, it can by readily corrected by existing algorithms [31].

4.2.2 The reasons to unwarp

The amplitude of the deformation g can be as large as 3–5 mm [81] (confirmed by our own observations), which typically amounts to several pixels. In some cases, as in Figure 4.1, specifically intended to illustrate EPI distortion, the deformation can be even more pronounced. Moreover, g can vary significantly from slice to slice and from acquisition to acquisition. For localization applications like stereotactic surgery, this

[1] This effect prevails over the increase of oxygen consumption.

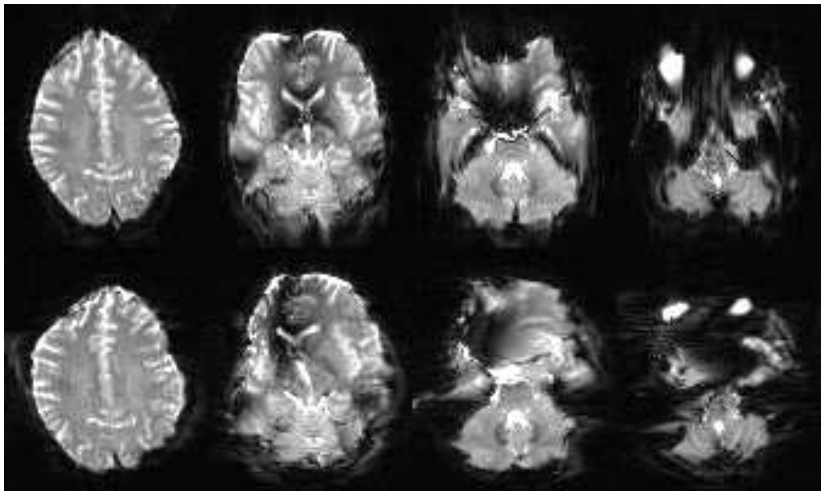


Figure 4.1: Demonstration of EPI distortion. Selected EPI brain slices taken with two different phase-encoding gradient orientations: anterior-posterior for the upper row and left-right for the bottom row. The vertical, resp. horizontal deformation of the upper, resp. bottom row images is clearly visible.

inaccuracy is much larger than the required limit of 1 mm and therefore EPI cannot currently be used to this end. It also severely hinders the performance of the statistical processing of sets of fMRI images used to obtain activation information. Since the task-induced signal changes represent typically only 5–10% of the mean signal intensity in fMRI [79, 82], they will not stand out clearly unless the perturbations caused by the deformation g are undone.

4.2.3 Existing distortion correction techniques

One approach consists in changing the acquisition procedure [80, 81, 83]. However, this is often not practical due to technical or organizational limitations, for example lack of support or approval. Furthermore, while the alternative acquisition sequences reduce the distortion, the distortion is never removed completely, and the methods usually sacrifice either sensitivity or acquisition speed.

The second group of methods uses a two-step procedure [81, 84]. First, a field map or a deformation map is obtained, e.g., from an image of a phantom. In the second step, this information is used to compensate for the deformation on real images. The major drawback of these methods is that it is impractical to build a phantom that would exactly duplicate the biological system being imaged [81], which limits the compensation only to field distortions other than those caused by the individual subject. Moreover, these other distortions are most likely already compensated for by the scanner manufacturer or operator.[2]

4.2.4 Unwarping by registration

We propose a third approach which, to the best of our knowledge, has never been applied to this particular problem. It consists of registering the distorted EPI image with a corresponding geometrically correct anatomical MRI image. In this way, we can recover the deformation g from a single EPI slice obtained by an unaltered, standard procedure. The registration can be performed manually [15], but this is tedious, time consuming, and prone to errors. An automatic procedure is advantageous, because it is faster, more precise, and does not require an expert.

Our goal in this Chapter is to present an automatic registration algorithm we have developed specifically for this problem; i.e., identifying a non-linear unidirectional two-dimensional warping. More precisely, given the observed EPI slice f^o and the corresponding undistorted anatomical MRI reference slice f^r, the task is to find a warping g so that the warped test image $f^o(g(x, y), y)$ matches as well as possible (in a sense defined later) the reference image f^r.

[2] This procedure is called shimming and is generally repeated before each series of acquisitions.

4.3 Proposed algorithm

We categorize registration algorithms according to the *warp* space used. In general, a deformation function g_V from a warp space V is described by a finite set of parameters c_k by means of a *warping model*.

At one end of the scale we have *non-parametric, local methods*. These methods are formulated either as *variational*, defining a scalar criterion to minimize, or (more generally) using PDEs. The continuously defined correspondence function that minimizes a given criterion (resp. that solves a given PDE) is sought for in a very large and unrestrictive function space, e.g., the Sobolev space W_2^2. The essence of these methods is *entirely in the criterion* (resp. PDE). The PDE come from the optical flow approach (gradient methods) [22], viscous fluid model [23–25], elastic deformations with physical analogs [3, 26] or without [27]. Some elastic deformations can also be modeled as potential fields [28].

At the other end, we have *parametric, global methods* that describe the correspondence function using a global model with a relatively small number of parameters [30]. The model mostly consists of expressing the warping function in a linear [31], global polynomial [32, 85], or harmonic basis [33]. For these methods, the deformation model corresponding to a specific warp space is as important as the criterion being minimized.

In this article, we consider mainly *intensity-based* registration methods, which directly take into account the voxel values, c.f. [86]. Other methods are based on matching *surfaces* [86–88], *curves* [89], or interpolating landmarks using radial basis functions, especially thin-plate splines [14, 38, 42, 86].

4.3.1 Semi-local model

The model proposed in this article is situated between the above-mentioned local and global methods, combining the advantages of both.

We parametrize the warp space by a *scale parameter* h and denote it V_h. The scale parameter corresponds loosely to the density of knots or landmarks. By changing h, we can approach either of the two limit cases or choose a compromise offering the best tradeoff. Big h yields a global model which has just a few parameters. Such a model is rather constrained, which is approximatively equivalent to strong explicit *regularization*. On the other hand, a small h gives a local model with many parameters, which generally leads to more complicated optimization. In exchange, a small h generally permits one to approximate any given g in V_h well, because the space V_h is big. (Arbitrarily small precision can be achieved as $h \to 0$.) This roughly corresponds to weak explicit regularization. Thus, the scale parameter can partly assume the role of an explicit regularization factor, unlike in local methods where the regularization is a part of the criterion.

In the next section, we give our motivation for the algorithm and for our particular choice of the warp model—uniform *cubic splines* represented by a linear combination of cubic *B-splines* [4, 5].

4.3.2 Univariate case

For the sake of explanation, let us begin with the one-dimensional case; i.e., with the task of recovering an univariate warping function $g : \mathbb{R} \to \mathbb{R}$. An example of such a function could be the affine map $g(x) = \alpha x + \beta$. Such linear dependence is rather frequently encountered in practice; for example, it arises when the acquisition techniques use different coordinate systems or when there is a movement between acquisitions.

The univariate equivalent of our matching problem becomes $f^o(g(x)) \simeq f^r(x)$. The landmark method would consist of (either manually or automatically) identifying a set of landmark pairs $\{(x_l^r, x_l^o)\}_{l=1}^L$, so that a feature found at location x_l^r in the reference image can be found (or associated with) a location x_l^o in the observed test image. For *interpolation*, this gives a set of L constraints for g: $g(x_l^r) = x_l^o$. To get a well-posed problem, we shall require g to minimize some criterion J. We do not want J to penalize linear dependencies, i.e., we want J to be zero for linear g. We want the resulting $g = \arg\min J$ to be linear with respect to the landmark coordinates $\{x_l^o\}$ and invariant with respect to a linear (affine) transformation of $\{x_l^r\}$, which means that if we take a linear combination of two sets of landmark coordinates, then the resulting g should be the same linear combination of the solutions corresponding to the two sets of landmarks. In other words, we want a solution that is invariant to the choice of a particular coordinate system, or to the choice of units. The simplest criterion satisfying these requirements is $J = \int |g''(x)|^2 \mathrm{d}x$, which is compatible with the elasticity theory, as it corresponds to 'strain' or 'bending' energy [14].

Besides interpolation, other approximation schemes can be applied, the most popular being *least-squares fitting*. It consists of minimizing an extended criterion $J' = J + \gamma \sum_l (g(x_l^r) - x_l^o)^2$. This has the advantage of accommodating uncertainty (noise) in landmark positions.

For both interpolation and least-squares fitting, as well as for *any other* criterion of the general form $J' = J + \sum_l \mathcal{L}(g(x_l^r), x_l^o)$ (with arbitrary function $\mathcal{L}$), the function g that minimizes the criterion can be shown to be a *cubic spline* [90, 91]; i.e., a piecewise cubic function that is twice continuously differentiable, with knots (the boundaries between polynomial pieces) at points $\{x_l^r\}$. In Appendix 4.6.1, we show that the solution can be expressed as a linear combination of *radial basis functions* $|x - x_l|^3$. However, these functions are not convenient to work with because of their instability and global support. Fortunately, it is possible to localize the $|x - x_l|^3$ functions using divided differences, which yields a base made of *cubic B-splines* with local support that generates the same space [74].

The automatic landmark method is rather difficult to apply because there is no automatic landmark detection algorithm available that would be sufficiently robust and precise, especially for our class of medical images which typically exhibit only a few distinct features. Moreover, it is difficult to automatically find common features in both anatomical MRI and EPI modalities. For this reason, instead of trying to work with landmarks, we introduce a data criterion E (defined in section 4.3.4) taking into account

the entire image and measuring the discrepancy between the warped version of the observed image $f^o(g(x))$ and the reference image $f^r(x)$. Then, we seek a deformation g such that the two images are as similar as possible, i.e., when E is minimized.

For the reasons mentioned above, we choose to search the warping g also in the cubic spline space, like in the landmark case. However, there are now no explicit landmarks available to put the knots on. We also do not know, how much useful information each part of the image can provide. Therefore, we will distribute the knots uniformly over the image. It follows that the function g will be a uniform cubic spline, which can be uniquely represented as a linear combination of uniformly-spaced cubic B-splines [4, 5]:

$$g(x) = \sum_{k \in K} c_k \, \beta_3(x/h - k) \tag{4.2}$$

where K is the set of the indices of the spline functions, the support of which intersects with the image; h is the knot spacing (the B-splines will be centered at points kh for $k \in K \subseteq \mathbb{Z}$). Working with uniform splines is also significantly faster with respect to non-uniform splines. See Appendix 4.6.2 for a definition of a cubic B-spline. Note that, in order to get a complete control over g, it is useful to put some spline knots 'outside' the image. For cubic splines, we need to put one such exterior knot at each side. Consequently, for an image size N and knot spacing h, we have $N/h + 2$ knots.

We have thus transformed the registration task into a non-linear finite-dimensional optimization problem: Find a set of coefficients $\mathbf{c}$ minimizing some criterion E.

4.3.3 Splines—a perfect fit

Let us now show several important properties of the spline model.

(a) *Good approximation properties*—The error of a cubic spline approximation decreases asymptotically as h^4 (measured by any L_p or l_p norm, $p \in \{1, 2, \ldots, \infty\}$). Quantitative analyses indicate that splines perform well in comparison with other wavelet-like basis functions [92].

(b) *Speed*—Cubic splines have a short compact support of length 4. They are symmetric and piecewise cubic. To evaluate $\beta_3(x)$ at one particular point, only 5 arithmetic operations (additions or multiplications) and 3 comparisons are needed. In multiple dimensions, where we will use tensor products of cubic splines as basis functions (see Section 4.3.5), the computational complexity stays low thanks to separability. The number of operations needed to evaluate g also does not depend on the total number of basis functions (and thus the number of parameters c_k).

(c) *Plausibility*—The spline model corresponds to a wide range of physical situations where the restoring force can be approximated as being linearly dependent on the displacement. In such situations, the generated deformation is physically plausible.

It is also a good approximation for cases when a better model is not known, such as the deformation of EPI images, even though it is not of mechanical origins.

(d) *Simplicity* — The model is linear in the parameters c_k and polynomial with respect to the position x. It is thus possible to truncate the Taylor expansion such that it is exact in some neighborhood of x with a typical diameter of $h/2$.

(e) *Scalability*—Thanks to the *q-scale relation* $\beta_3(x/q) = \sum_k \zeta(k)\beta_3(x-k)$, where $q \in \mathbb{N}$, we have the embedding $V_{hq} \subseteq V_h$; i.e., the transition from a coarse space V_{hq} to a finer space V_h is exact [93].

4.3.4 Data criterion

A reasonable and most often ([31]) used way to measure the discrepancy between two images is the *sum of squared differences* (SSD) criterion

$$E = \sum_{i \in I} (f^o(g(i)) - f^r(i))^2 \tag{4.3}$$

where the sum is over all pixels in the image. Note that minimizing this criterion is equivalent to calculating the maximum likelihood estimate of the unknown parameters assuming that the difference is an i.i.d. Gaussian noise, and that the true test image is indeed a geometrically distorted version of the reference one. Moreover, the SSD criterion is also algorithmically advantageous because it is easy to evaluate (including its derivatives) and because it depends smoothly on the parameters. Interpolation (cf. Section 4.3.7) is needed to evaluate this criterion, as it calls for values of f^o on generally non-integer coordinates.

4.3.5 Bivariate case

The transition to the bivariate case is straightforward. The criterion becomes

$$E = \sum_{\mathbf{i} \in I} (f^o(\mathbf{g}(\mathbf{i})) - f^r(\mathbf{i}))^2 \tag{4.4}$$

where we have taken the convenient notation

$$\mathbf{g}(\mathbf{i}) = \mathbf{g}(i,j) = \begin{bmatrix} g(i,j) \\ j \end{bmatrix} \tag{4.5}$$

and where the warping function is now described by a two-dimensional array of coefficients

$$\begin{aligned} g(\mathbf{x}) = g(x,y) &= \sum_{\mathbf{k}\in K} c_{\mathbf{k}}\, \boldsymbol{\beta}_3(\mathbf{x}/\mathbf{h} - \mathbf{k}) = \\ &= \sum_{(k_x,k_y)\in K} c_{k_x,k_y}\, \beta_3(x/h_x - k_x)\beta_3(y/h_y - k_y) \end{aligned} \tag{4.6}$$

Tensor products of splines were also used in [94].

4.3.6 Optimization algorithm

The criterion is minimized with respect to the coefficients c_k using a regularized version of the Newton method [31, 50], inspired by the Marquardt-Levenberg algorithm. This algorithm smoothly varies between the gradient-descent and the Newton approach, which gives it robustness and quadratic convergence near the optimum.

The algorithm uses first two derivatives of the criterion E with respect to c_k; i.e., $\nabla_{\mathbf{c}} E$ and $\nabla^2_{\mathbf{c}} E$. Thanks to the spline representation (4.2), the derivatives can be calculated exactly and at a small cost. As the number of components each pixel contributes to remains constant, the cost of evaluating E, $\nabla_{\mathbf{c}} E$ and $\nabla^2_{\mathbf{c}} E$ does *not depend* on the number of coefficients c_k (or, equivalently, the spacings h_x, h_y) used to describe the deformation. See Appendix 4.6.3 for explicit formulas.

At each step we update the vector of all coefficients $\mathbf{c}$ to $\mathbf{c} + \Delta\mathbf{c}$ by taking

$$-\left(\nabla^2_{\mathbf{c}} E(\mathbf{c}) + \lambda \mathbf{I}\right) \Delta\mathbf{c} = \nabla_{\mathbf{c}} E(\mathbf{c}) \tag{4.7}$$

where the regularization factor λ is divided by a constant λ_f if the previous step resulted in a decrease in E, otherwise it is multiplied by the same amount.[3] We iterate as long as the relative and absolute change of E stay above *a priori* given thresholds.

When the number of coefficients exceeds a certain limit, the Hessian matrix $\nabla^2_{\mathbf{c}} E$ gets too big for the linear equation set (4.7) to be efficiently solvable and this algorithm ceases to be practicable. Note that the theoretical computational complexity of solving (4.7) increases as h^{-6}. The asymptotical memory requirements grow as h^{-4}. While iterative linear equation solvers generally speed up the solution of (4.7), it is at the cost of a loss in precision, and the overall gain in our case is insignificant.

When the size of the Hessian matrix exceeds our computational capacities, we replace the Marquardt-Levenberg optimizer by a *conjugate gradient* method [50], which converges quadratically too without explicitly calculating and storing the Hessian matrix. Even though the conjugate gradient method needs more iterations for smaller problems, it outperforms Marquard-Levenberg for bigger ones.

[3] We use $\lambda_f = 10$.

4.3.7 Image interpolation model

To calculate the derivatives, as well as to evaluate the criterion (4.4), an *image interpolation* model is needed to get a continuous form ${}^c f(\mathbf{x})$ from a discrete image $f^o(\mathbf{i})$. Because of their good approximation properties, simple analytic form and effective algorithms available, we use cubic B-splines here as well

$$ {}^c f(x,j) = \sum_i b_{ij}\, \beta_3(x-i) $$

$$ \text{where} \quad {}^c f(i,j) = f^o(i,j),\ \forall\, (i,j) \in I \qquad (4.8) $$

The coefficients b_{ij} can be obtained prior to registration by an efficient filtering algorithm [5], which incurs negligible overhead. For the filtering, we are using mirror boundary conditions on the image; in this way we have the same number of coefficients b_{ij} as there are pixels in the original image.

4.3.8 Multiresolution

The robustness and efficiency of the algorithm can be significantly improved by a *multiresolution* approach: The task at hand is first solved at a coarse scale. Then, the results are propagated to the next finer level and used as a starting guess for solving the task at that level. This procedure is iterated until the finest level is reached.

In our algorithm, multiresolution is used twice. First, we build an *image pyramid*: a set of gradually reduced versions of the original image [95]. This pyramid is compatible with our image representation (4.8) and is optimal in the l_2-sense (i.e., in the sense of the criterion (4.4)), which ensures that the approximation made by substituting the lower resolution image is the best possible. Based on an image of size $N_x \times N_y$, we create a sequence of images with sizes $\mathcal{M}_I = \{(N_x^1, N_y^1), (N_x^2, N_y^2), \ldots, (N_x^{m_I}, N_y^{m_I})\}$, where $N_x^{m_I} = N_x$, $N_y^{m_I} = N_y$, and where $N_x^j = \lfloor N_x^{j+1}/2 \rfloor$, $N_y^j = \lfloor N_y^{j+1}/2 \rfloor$. The optimum starting size (N_x^1, N_y^1) depends on the image. We chose $16 < \min(N_x^1, N_y^1) \leq 32$, which works well in most cases.

Second, we use multiresolution also for the warping function. We start with a deformation g described with very few parameters c_k, and with a large distance h between knots. After the optimization of c_k is complete, we halve the distance between knots. This approximately corresponds to doubling the number of knots in each direction, i.e., quadrupling the number of coefficients c_k. Because of the two-scale spline relation, we can *exactly* represent the warping function from the old, coarse space, in the new, finer space. More precisely, the sequence of knot spacings is going to be $\mathcal{M}_W = \{(h_x^1, h_y^1), (h_x^2, h_y^2), \ldots, (h_x^{m_W}, h_y^{m_W})\}$, where $h_x^1 = N_x$, $h_y^1 = N_y$; the final element $(h_x^{m_W}, h_y^{m_W})$ corresponds to the user-chosen target grid size. The sequence obeys $h_x^{j+1} = h_x^j/2$, $h_y^{j+1} = h_y^j/2$. The process starts with g being identity.

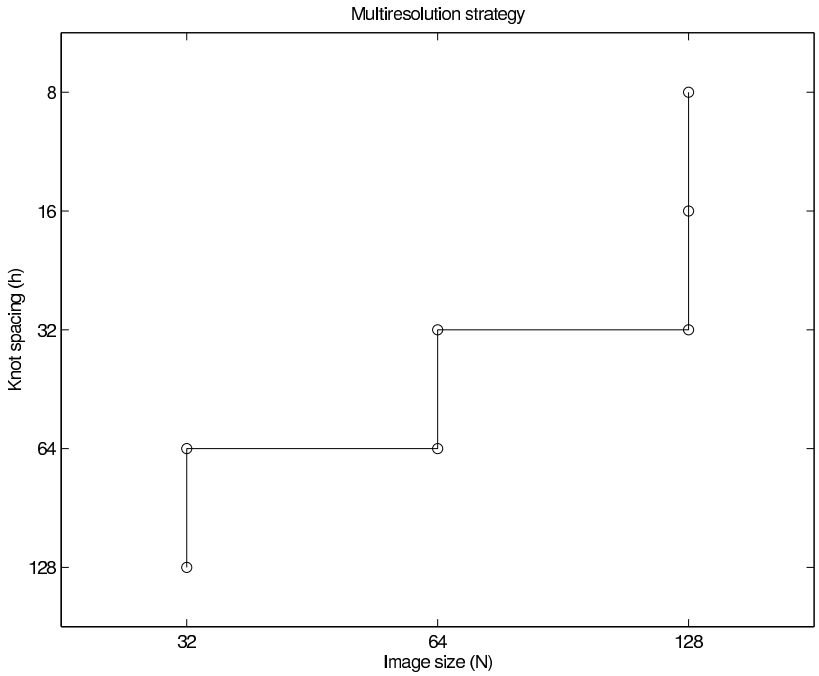


Figure 4.2: The multiresolution strategy starts with a small version of the image (horizontal axis) and a small warping control grid (vertical axis). After several steps, and by augmenting resolution alternatively in the two domains, we reach the original image resolution and the desired size of the control grid.

The global strategy that combines the two multiresolutions is depicted in Figure 4.2. Symbolically, to construct the double multiresolution $\mathcal{M}$ we first extend the shorter of $\mathcal{M}_I$, $\mathcal{M}_W$ to the length $m = \max(m_I, m_W)$ by repeating the last element. Then, $\mathcal{M}$ consists of pairs of elements from $\mathcal{M}_I$, $\mathcal{M}_W$ in alternate progression: $\mathcal{M} = \{(\mathcal{M}_I^1, \mathcal{M}_W^1), (\mathcal{M}_I^1, \mathcal{M}_W^2), (\mathcal{M}_I^2, \mathcal{M}_W^2), (\mathcal{M}_I^2, \mathcal{M}_W^3), \ldots, (\mathcal{M}_I^m, \mathcal{M}_W^m)\}$.

The consequence of using multiresolution is that the algorithm works best for images and deformations that follow the multiresolution model; i.e., when a low resolution version is a good approximation of the finer resolution version.

4.3.9 Invertibility

We implicitly assumed that the deformation $\mathbf{g}$ is recoverable and recoverability requires invertibility. In case the EPI artifacts cause the signal from adjacent pixels to blend

due to the fold-over effect, the invertibility condition might not be satisfied and the deformation is not recoverable. However, this does not happen in practice. To ensure the stability of the registration, it is necessary to enforce the invertibility also on the trial solutions during the optimization process, i.e., to perform constrained optimization. If we assume that the deformation conserves orientation, the sufficient condition ensuring local invertibility is the positivity of the Jacobian, $\det(\nabla_{\mathbf{x}}\mathbf{g}) > 0$, everywhere. In our case, the invertibility condition reduces to $\forall\,(x,y) \in S;\ \partial g/\partial x > 0$, where S is the image domain.

Although an iterative algorithm can be found verifying this condition exactly, it is not practical to apply it for performance reasons. Instead of checking that $\partial g/\partial x > 0$ on a continuous domain, we sample this condition on pixel coordinates. At this scale, the two conditions are essentially equivalent.

There is little hope of finding an algorithm capable of solving a constrained optimization problem of our complexity (highly non-linear criterion, hundreds of parameters and tens of thousands of constraints) in a reasonable time. We have therefore chosen to convert the constrained search into an unconstrained one using an exponential penalty cost function. A set of constraints

$$0 < u_{\mathbf{p}} = \left.\frac{\partial g}{\partial x}\right|_{(x,y)=\mathbf{p}} \tag{4.9}$$

is replaced by a penalty function E_p

$$E_p = \gamma \sum_{\mathbf{p}} \mathrm{e}^{-\alpha u_{\mathbf{p}}} \tag{4.10}$$

Consequently, we will minimize the combined criterion

$$E_c = E + E_p \tag{4.11}$$

The choice of the constants α and γ is a tradeoff between speed and precision. As α increases, the penalty function gets steeper, which improves precision in the vicinity of the constraints. At the same time, the criterion becomes highly non-linear, which slows down the optimization. We set α and γ so that $E_p \approx E$ on the boundary of the permissible space and $E_p \approx 10^{-5}E$ for the initial configuration, when g is an identity.

4.3.10 Preprocessing

To apply the SSD criterion, we need to make the test and reference images more similar, so that their difference after warping is as close as possible to white noise. We choose therefore to apply a preprocessing step that consists of high-pass filtering and histogram equalization. The effect can be seen in Figure 4.3.

The preprocessing makes it unnecessary to add a special parameter accounting for differences in intensity profiles of the two images. It also helps to compensate for the intensity distortion due to in-plane dephasing, which is also caused by the magnetic field inhomogeneity. The dephasing cannot be compensated completely, as it is impossible to differentiate between effects of dephasing and warping on a single pixel.

4.4 Experiments

4.4.1 General comments

We tested the performance of our algorithm on several hundreds of images. In addition, we compare to the automatic method the results of the registration of thirty image pairs by three different people, including one experienced practitioner. For the manual registration, we use the standard thin-plate spline method [14, 15, 45].

Unless stated otherwise, the tests are performed on 128×128 pixel spin-echo anatomical MRI images of the brain, on a SUN Ultra workstation, and the published numbers are arithmetical means of the results of experiments made on all of the 30 horizontal slices of a brain volume. A typical pair consisting of corresponding anatomical and EPI images is shown in the upper row of Figure 4.3.

4.4.2 Error measurement

As a main measure to compare different solutions to the registration problem, we use the *warping index* defined in [31] as $\varpi = \frac{1}{\|R\|} \sum_{\mathbf{i} \in R} |g(\mathbf{i}) - g^*(\mathbf{i})|$, where g^* is the reference solution (ideally, the true deformation), and $R \subseteq I$ is the region of interest (in our case, the interior of the brain). This corresponds to the mean per-pixel precision of the result.

We shall also use the *sum of squared differences* (SSD), as defined by (4.3). It measures the similarity of the reference and warped test images as perceived by the algorithm, and corresponds to the quantity minimized.

4.4.3 Sources of error

There are several reasons why we cannot expect a perfect registration (indicated by $\varpi = 0$).

1. *Different images*—Despite the assumptions made when deriving our criterion, the images we are asked to register are not geometrically deformed versions of each other. To this we must add the effects of discretization, quantization, and noise. Any of them may result in spurious minima of the criterion E, misleading the algorithm. This problem can be alleviated by preprocessing, or by choosing a different criterion, but it can never be completely removed. The situation is more favorable

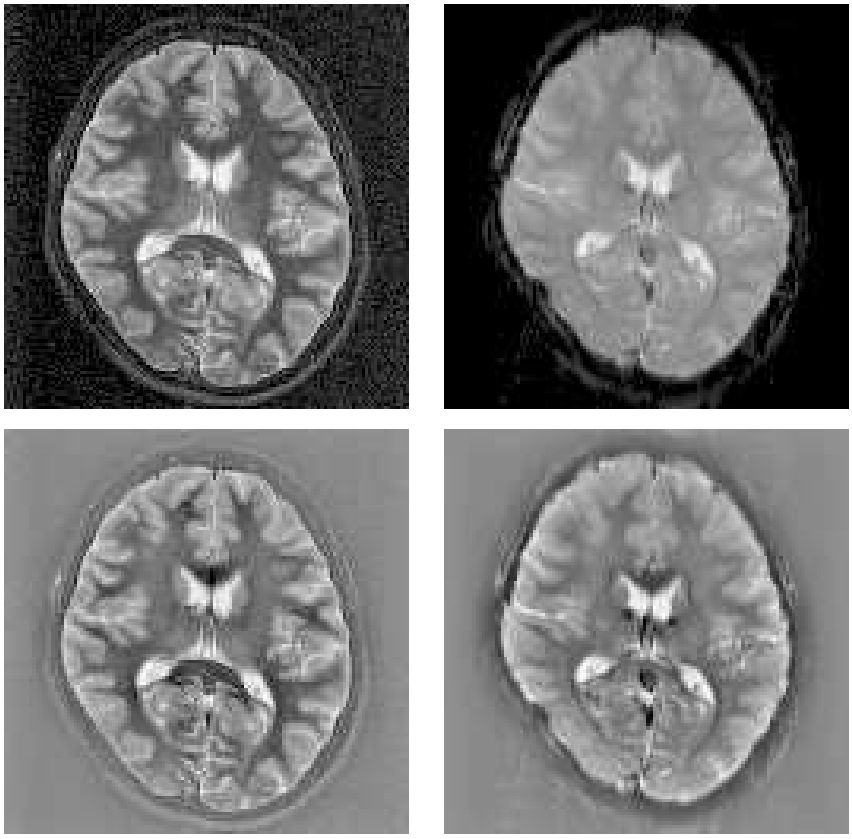


Figure 4.3: An anatomical MRI image (top left) and a corresponding EPI image (top right). Both images represent a horizontal slice through the middle of a human brain. The bottom line shows the same images after preprocessing.

when a small number of parameters is sought, for the random effects tend to cancel out.

2. *Local minima*—The algorithm with its local vision cannot distinguish between local and global minima of the criterion and can therefore get trapped in a local minimum. Multiresolution improves significantly this aspect, but it does not solve it entirely. Again, the situation is much more favorable for a small number of parameters.

3. *Warp space*—The true warping does not necessarily belong to the warp space V_h. We define $\varpi_{\min}$ as the minimum achievable registration error, which is equal to the distance between the true warping and the closest point in the warp space V. We call this closest point (which is the best possible approximation of g in V) a *projection* $\mathcal{P}_V g$. In the L_2 or l_2 sense, it is equal to $\sum_k \mu_k \varphi_k$, where the coefficients μ_k verify $\langle g, \varphi_l \rangle = \sum_k \mu_k \langle \varphi_k, \varphi_l \rangle$, and where we have denoted the basis functions by φ_k in order to simplify the notation. For other norms, iterative procedure must be used. In a controlled experimental environment, we can thus calculate $\varpi_{\min}$ exactly and compare it with the results of our algorithm.

4. *Lack of details*—The precision is limited by the local resolution of the images. When two corresponding sharp edges occur in both images, sub-pixel registration precision is often attainable. On the other hand, in homogeneous, texture-less regions, there is little hope of recovering any information whatsoever.

5. *Criterion surface complexity*—Depending on the images being registered, the dependence of the criterion on the parameters can be non-linear, non-convex, and generally very complicated. In such a situation, the convergence is extremely slow. If the time is limited, we must stop the algorithm before any significant improvement has been realized.

6. *Numerical precision*—Insufficient numerical precision can hinder the performance of the optimizer. We have encountered this problem only rarely, when iterative methods were used to solve the normal equation set (4.7).

We have tried to design the experiments to separate the effect of the above-mentioned factors whenever possible; however, this will rarely be the case in a real-life situation.

4.4.4 Deformation generator

To test our algorithm, we have implemented a wavelet-based deformation generator. We want to generate a random Sobolev-type deformation—a deformation lying in a prescribed Sobolev space W_2^r. The parameter r refers to the regularity. It is equal to the number of derivatives in the L_2-sense and it is also strongly related to point-wise differentiability. The higher the r, the more regular are the functions from W_2^r. Wavelets are

known to be good bases for functions lying in Sobolev spaces and the decay of wavelet coefficients across scales is directly related to a Sobolev-type regularity. Let $\theta_{j,k}$ denote the coefficients of a wavelet expansion[4]

$$g(x) - g_I(x) = \sum_{j,k} \theta_{j,k}\, 2^{j/2} \psi(2^j x - k) \tag{4.12}$$

where g_I is an identity transform and ψ is an orthogonal wavelet. Then the displacement $g - g_I$ belongs to a Sobolev space W_2^r if and only if

$$\sum_{j,k} 2^{2jr} \left|\theta_{j,k}\right|^2 < \infty \tag{4.13}$$

provided that the regularity of ψ is greater than r [69]. It follows that, for (4.13) to hold, the necessary condition is

$$\exists\, C \in \mathbb{R}; \forall\, j,k; \left|\theta_{j,k}\right|^2 < C\, \xi^j \quad \text{with} \quad \xi = 2^{-2r} \tag{4.14}$$

Practically, we generate our deformations using zero-mean, normally distributed coefficients with variance

$$\mathbf{E}\left\{\theta_{j,k}^2\right\} = \sigma_0^2\, \xi^j \tag{4.15}$$

where σ_0^2 governs the total energy of the deformation. Such a deformation will be denoted $W_H(\sigma_0, r)$, where the relation between r and ξ is given by (4.14). Note that the generated displacements are white noise for $r = 0$, and become progressively smoother as r increases. The regularity of the deformation converges to that of the generating wavelet ψ for $r \to \infty$. For moderate to large r, we get a *hierarchical* warping: a deformation comprising displacements at several scale levels with gradually decreasing amplitudes, from important large-level deformation towards progressively smaller finer-level details. The algorithm should work well for such deformations which are compatible with the multiresolution strategy.

Finally, the deformation can be projected onto V_h if needed, in which case we denote it $W_H(\sigma_0, r, h)$. Typically, we use Battle-Lemarié wavelets of order 4, $\sigma_0 = 5$ and $r = 0.5 \sim 1.6$, depending of what aspect of the algorithm we want to highlight. For some experiments, we have also added an affine component. Examples of generated deformations are shown in Figure 4.4.

4.4.5 Ideal case

We begin our series of experiments with an ideal case: the test (f^o) and reference (f^r) images are identical except for a known transformation, no noise is present. We use

[4]For brevity, we shall deal here with the unidimensional case only.

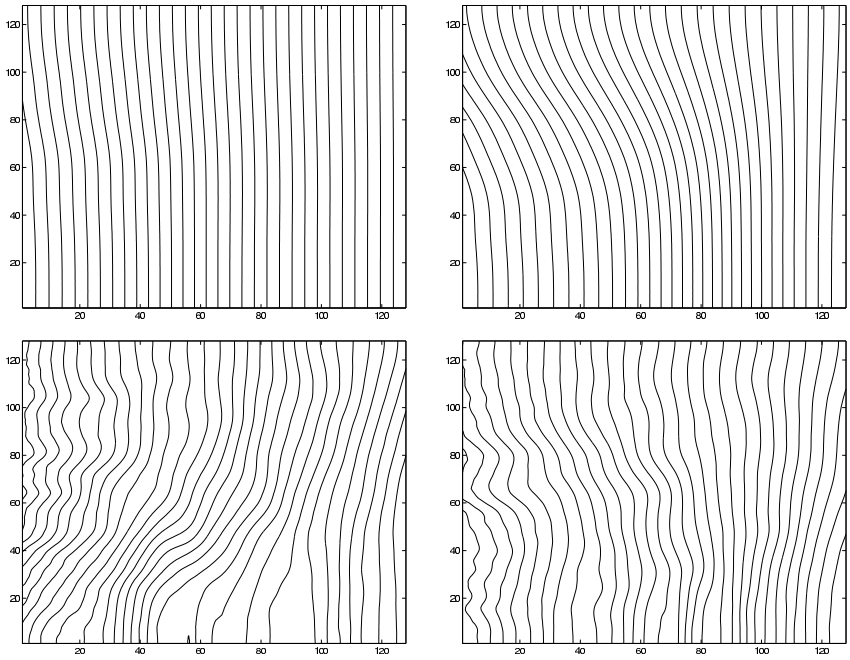


Figure 4.4: Examples of deformation presented as contour plots. The top row ones were generated with $W_H(10, 1.6)$, the bottom ones with $W_H(10, 0.6)$.

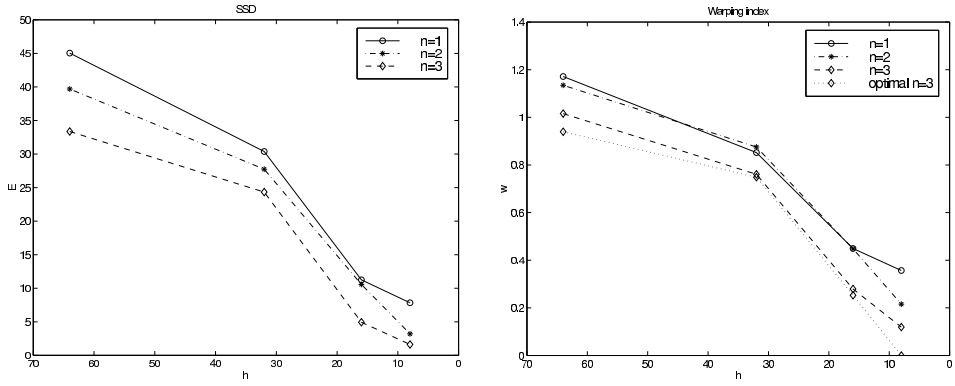


Figure 4.5: The quality of the registration as a function of the warp spline degree and the knot spacing. The initial values ranges (prior to the registration) were $E = 150$ and $\varpi = 3.5$. Each point shown is an average of thirty experiments using the anatomical image from Figure 4.3.

a random Sobolev deformation $W_H(5, 1.6, 8)$ as defined in the previous section. We show how the criterion E and the warping index ϖ decrease with the knot spacing h. Moreover, we want to demonstrate the advantage of using cubic splines to represent the warping, as opposed to linear and quadratic ones[5] (Linear splines are sometimes used for motion estimation [35, 53].)

Figure 4.5 displays E and ϖ as a function of the degree n of the splines used to represent the deformation g, and the knot spacing h. It clearly shows the benefit of using a high degree n in high-precision applications. The minimum achievable error $\varpi_{\min}$ (cf. Section 4.3.3b) is shown by a dotted line for the cubic case and marked *optimal*.

4.4.6 Image interpolation order

As many other registration algorithm use only linear interpolation on the image, we want to show in this experiment that higher-order image interpolation is advantageous from the point of view of both precision and speed.

Table 4.1 demonstrates the dependency of the registration accuracy and speed on the degree n_i of the splines used for image interpolation. As a deformation, we have chosen $W_H(5, 0.5, 64)$. The stopping criterion was identical in all cases.

Besides providing a less accurate approximation, linear splines are penalized by not

[5] For splines of even degrees, there is an additional shift term of $1/2$ in (4.2), e.g., $\beta_2(x/h - k + 1/2)$, which minimizes the number of knots needed. Other changes are straightforward.

Table 4.1: The quality of the registration as a function of the image interpolation spline degree. The warping function was interpolated by cubic splines.

h		32			16		
n_i	before	1	2	3	1	2	3
E	113.4	2.53	1.84	1.50	1.12	1.00	0.58
ϖ	2.93	0.130	0.111	0.098	0.074	0.067	0.057
time [s]		1010	766	69	1983	1500	103

providing a meaningful second derivative estimate of the image, which forces us to use the slower conjugated gradient algorithm in place of the Marquardt-Levenberg optimizer.

The table shows that, although all results are accurate to a fraction of a pixel, the convergence speed varied a lot. This proves that the benefit from better approximation properties of higher-order splines with respect to linear, or even quadratic ones, indeed overweights the increase in computational complexity per iteration. This is consistent with other findings in the literature [31]. Being able to better estimate the image derivatives $\nabla_{\mathbf{x}}{}^c f$, one is able to better estimate the criterion derivatives $\nabla_{\mathbf{c}} E$ and $\nabla_{\mathbf{c}}^2 E$. This permits the optimization algorithm to acquire a more precise local model of $E(\mathbf{c})$, which in turn leads to more efficient optimization steps and faster convergence. The ability to precisely represent an image from a few samples is crucial at coarse levels of the image pyramid. The iterations there are relatively inexpensive compared to finer levels because we process much less data; it is therefore beneficial to get as close as possible to the optimum. In this manner we provide a good starting estimate for the next finer level, where each iteration costs at least 4 times more, and thus saving the overall effort.

4.4.7 Noisy case

Figure 4.6 demonstrates the dependence of the registration accuracy on the signal-to-noise ratio. For this series of experiments, the test images were obtained from a known transformation of a reference image with a various level of white Gaussian noise added. Here, an in-space deformation $W_H(5, 1.16, 32)$ was used. We observe that the degradation of the algorithm's performance is graceful for SNR $> 10\,\mathrm{dB}$.

4.4.8 Out-of-space deformation

This experiment illustrates the behavior of the algorithm as the deformation progressively moves out of the warp space. Specifically, we have used $W_H(5, 0.5, 32) + \alpha W'_H(5, 0.5)$ with $0 \leq \alpha \leq 1$, where the first term corresponds to the in-space part of the deformation

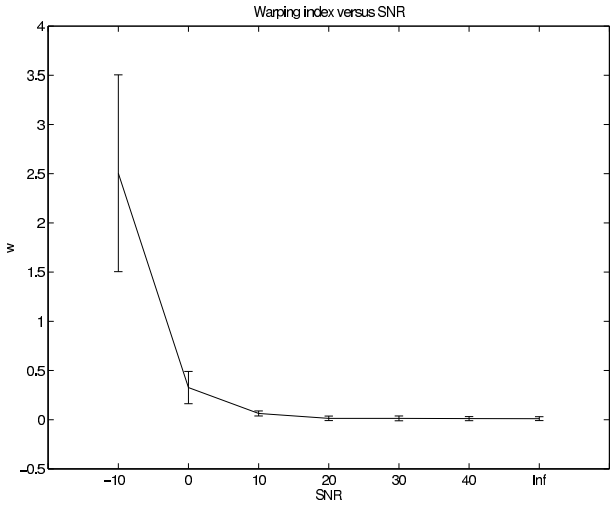


Figure 4.6: The quality of the registration as a function of the SNR in dB. $h = 32$, $\varpi_{\text{before}} = 3.34$. The error bars mark one standard deviation.

and W'_H is the displacement associated with W_H, i.e., $W_H - g_I$. Figure 4.7 compares the attained and theoretically attainable warping index and the corresponding SSD.

It can be seen that, the experimentally recovered deformation is almost as close to the true one as theoretically possible, with the exception of $\alpha = 0$; i.e., a deformation lying exactly in the warp space. On the other hand, we obtain mostly a smaller SSD than what corresponds to the projection. This demonstrates that the relationship between the criterion we optimize (E) and the true error we make (ϖ) is far from straightforward: the minimum of the difference (E) does not correspond exactly to the true solution as measured by ϖ. We will see this sort of behavior in Section 4.4.10, too.

4.4.9 Multiresolution strategy

In Table 4.2, we compare three multiresolution strategies: the strategy actually used, a strategy without multiresolution in the image size, and a strategy without multiresolution in the warping grid size. The results show that both multiresolutions improve accuracy as well as speed of the algorithm because they reduce the amount of data to be treated and provide a smoothed version of the problem.

Multiresolution in the warp grid size significantly improves accuracy because it avoids that the algorithm is trapped in a local minimum. The multiresolution in the image size reduces the amount of data to be treated and consequently also the execution time,

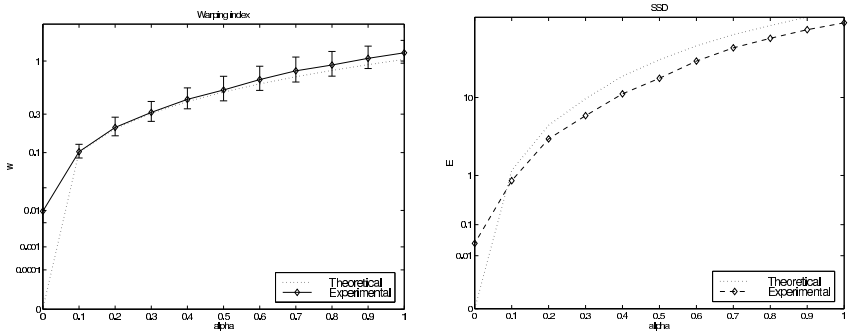


Figure 4.7: Warping index (left) and SSD (right) for an out-of-space deformation. The mean initial values (prior to the registration) ranged between $E = 146$, $\varpi = 2.8$ for $\alpha = 0$ and $E = 181$, $\varpi = 3.47$ for $\alpha = 1$. Error bars show one standard deviation.

mainly for coarse control grids.

The joint strategy combines the advantages of both multiresolutions and yields the best results.

4.4.10 Artificial EPI images

By filtering and performing histogram modification of the anatomical MRI images, we obtain images that are visually equivalent to the corresponding EPI images (compare Figures 4.8 and 4.3) and we use them as test images. We warp the anatomical images with a known deformation $W_H(5, 0.9)$ and use them as reference images.

The reference/test image pairs are registered automatically as well as manually using the landmark method by 3 people, including one experienced practitioner. The results are presented in Table 4.3. For the manual case, the best results obtained (in the sense of ϖ among all the attempts of all the participants) are shown. The column marked *ideal* shows the minimum attainable warping index $\varpi_{\min}$, given the fact that the target deformation does not belong to the space searched by the algorithm. The results demonstrate that the automatic method is vastly superior to the manual one.

4.4.11 Real EPI images

Figure 4.9 shows a typical pair of corresponding anatomical and EPI images with superimposed contours of the anatomical image before and after manual and automatic registration. It illustrates that the automatic procedure leads to subjectively comparable or better results than the manual one.

Table 4.2: Different multiresolution strategies. Image size is denoted by N, grid spacing by h. From top to bottom: joint, image size only, and warp grid size only multiresolutions.

Warping index before registration $\varpi = 9.3$.

h	128	64	64	32	32	16	
N	32	32	64	64	128	128	
ϖ	0.43	0.40	0.15	0.15	0.0038	0.0029	total
time	2.2	2.0	29.0	1.2	20.0	23.2	78

h	16	16	16	
N	32	64	128	
ϖ	1.40	0.86	0.69	total
time	14.0	27.1	129.8	171

h	128	64	32	16	
N	128	128	128	128	
ϖ	0.145	0.0043	0.0038	0.0031	total
time	164.9	101.6	24.6	34.4	326

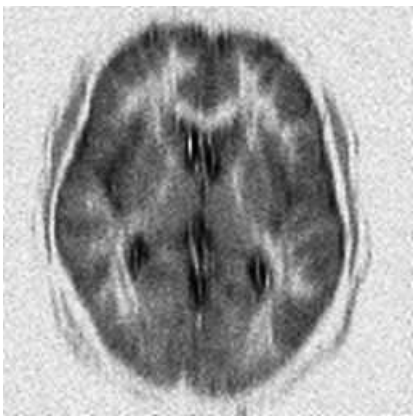


Figure 4.8: Artificial EPI image

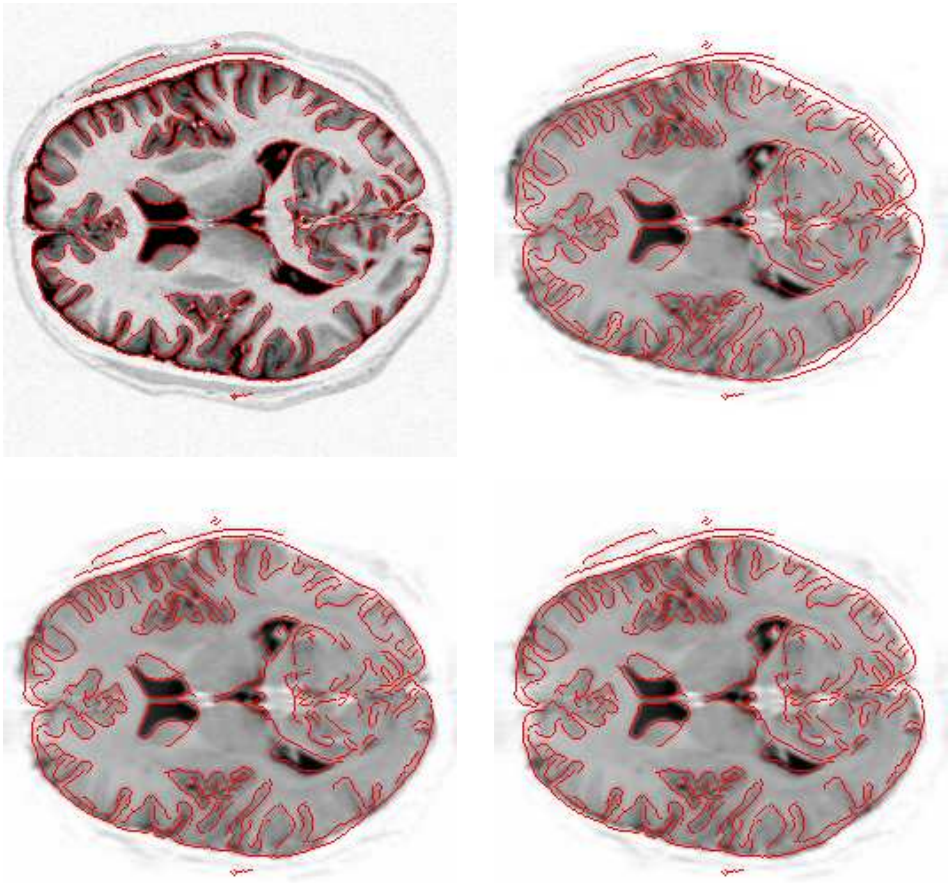


Figure 4.9: Anatomical (top-left) and EPI (top-right) images before registration, with superimposed contours from the anatomical images. EPI image after automatic (bottom-left) and manual (bottom-right) registration.

Table 4.3: Manual versus automatic registration. ϖ is the warping index, E is the mean-square difference.

	before	manual	automatic	ideal	true
ϖ	2.60	1.01	0.44	0.19	0
E	246	178	161	163	162
time		$\sim 5\,\text{min}$	$\sim 30\,\text{s}$		

4.5 Conclusions

We have suggested a new approach for undoing non-linear unidirectional deformations in EPI images. We proceed by registering them with corresponding geometrically correct anatomical MRI images. We have developed a fully automatic image registration algorithm specialized for this task. Our technique increases the geometrical precision of EPI images and thus improves the quality of information obtainable from these images. This will allow the use of EPI images in many clinical and diagnostic applications where they could not have been used previously, as well as to increase their usefulness in existing applications.

As an additional benefit, our method can be extended to compensate for other causes of geometrical distortion of EPI images besides imperfect magnetic field, such as heart beat and respiration. (In this case, we would look for bidirectional warping.)

The novelty of our registration algorithm stems from a high-order spline model for the warping. It has good approximation properties and lends itself well to a multiresolution approach, while permitting an efficient implementation. We have also taken advantage of a spline model for the image being warped, leading to a second dimension of the multiresolution strategy, and yielding additional computational savings. Finally, we have replaced the customary regularization criterion by a scale parameter of the search space.

We have also presented many experiments to demonstrate the performance of our algorithm.[6] We plan to carry out more clinical experiences to prove conclusively the potential of our method in a real-world setting.

4.6 Acknowledgment

We thank the anonymous reviewers for bringing to our attention a recent conference report [36], which describes a similar approach to ours and that was published after this manuscript had been submitted, at about the same time as our own conference report [2].

[6] An online demonstration of our algorithm is available on our WEB page *'http://bigwww.epfl.ch/'*.

Appendix

4.6.1 Optimality of the Cubic Spline Model

The landmark fitting from Section 4.3.2 is mathematically equivalent to the *smoothing spline* problem frequently encountered in statistics [96]. We present here an informal derivation of the variational property of cubic splines. We use an original Fourier-based technique which is instructive and concise.

Consider the following *approximation problem*: Given a set $\{(x_i, y_i) \in \mathbb{R}^2\}$ and a function $\mathcal{L} : \mathbb{R}^2 \to \mathbb{R}$, find a function $g : \mathbb{R} \to \mathbb{R}$ minimizing the functional criterion

$$J'(g) = J(g) + \sum_i \mathcal{L}(g(x_i), y_i) \quad \text{with} \quad J(g) = \int |g''(x)|^2 \mathrm{d}x \tag{4.16}$$

It is not difficult to show that the solution of this problem belongs to the same class as a solution of the *interpolation problem*, that would consist of finding g minimizing $J(g)$ under the constraints $g(x_i) = z_i$, equivalent to landmark interpolation from Section 4.3.2. We will therefore concentrate on the interpolation case here.

Using the Lagrange multiplier method, we construct an augmented criterion

$$J_\lambda = \int |g''(x)|^2 \mathrm{d}x - 24 \sum_i \lambda_i \left(g(x_i) - z_i\right) \tag{4.17}$$

We then express J_λ in terms of $\hat{g}(\omega)$, the Fourier transform of g

$$J_\lambda = \frac{1}{2\pi} \int \omega^4 |\hat{g}(w)|^2 \mathrm{d}\omega - 24 \sum_i \lambda_i \left(\frac{1}{2\pi} \int \overline{\hat{g}(\omega)}\, \mathrm{e}^{-j\omega x_i} \mathrm{d}\omega - z_i \right) \tag{4.18}$$

where we have used $\hat{g}(\omega) = \overline{\hat{g}(-\omega)}$. To find an optimal $\hat{g}(\omega)$, we differentiate J_λ with respect to $\overline{\hat{g}(\omega)}$ and we impose the first order change to be zero:

$$\Delta J_\lambda = 2\,\mathrm{Re}\left(\frac{1}{2\pi} \int \overline{\Delta \hat{g}} \left(\hat{g}\, \omega^4 - \, -12 \sum_i \lambda_i \mathrm{e}^{-j\omega x_i} \right) \mathrm{d}\omega \right) = 0 \quad \text{for all } \overline{\Delta \hat{g}(\omega)} \tag{4.19}$$

This gives

$$\hat{g} = \sum_i \lambda_i \frac{12\, \mathrm{e}^{-j\omega x_i}}{\omega^4} \tag{4.20}$$

By interpreting $\mathrm{e}^{-j\omega x_i}$ as a phase shift, we get

$$g(x) = \sum_i \lambda_i\, \varrho(x - x_i) \tag{4.21}$$

where ϱ is the inverse Fourier transform of $12/\omega^4$, which gives a general form of all solutions to our interpolation or approximation problems as

$$g(x) = \sum_i \lambda_i |x - x_i|^3 \tag{4.22}$$

The basis function $|x|^3$ is twice continuously differentiable and piecewise cubic. Hence, g is a cubic spline. In order to be complete, this solution needs to be augmented by a linear term generating the null space of J. For a rigorous treatment and generalization to multiple dimensions, we refer to [45].

4.6.2 B-Splines

A B-spline β_r of degree r is recursively defined as

$$\beta_r = \beta_{r-1} * \beta_0 \quad \text{for } r > 0$$
$$\beta_0(x) = \begin{cases} 1 & \text{if } x \in (-\frac{1}{2}, \frac{1}{2}) \\ 0 & \text{otherwise} \end{cases}$$

B-splines as defined above have a compact support $(-r/2 - 1/2, r/2 + 1/2)$, are symmetric, and are $(r-1)$-times continuously differentiable. Specifically,

$$\beta_3(x) = \begin{cases} 2/3 - (1 - |x|/2)x^2 & \text{if } 0 < |x| \leq 1 \\ (2 - |x|)^3/6 & \text{if } 1 < |x| < 2 \\ 0 & \text{otherwise} \end{cases} \tag{4.23}$$

4.6.3 Explicit Derivatives

Given E by (4.4),(4.6), and (4.8), let us express the components of $\nabla_{\mathbf{c}} E$ and $\nabla^2_{\mathbf{c}} E$.

$$\frac{\partial E}{\partial c_{\mathbf{k}}} = 2 \sum_{\mathbf{i} \in I} e_{\mathbf{i}} \frac{\partial e_{\mathbf{i}}}{\partial c_{\mathbf{k}}} \quad \text{where} \quad e_i = {}^c f(\mathbf{g}(\mathbf{i})) - f^r(\mathbf{i}) \tag{4.24}$$

$$\frac{\partial e_{\mathbf{i}}}{\partial c_{\mathbf{k}}} = \left. \frac{\partial \, {}^c f}{\partial x} \right|_{\mathbf{x}=\mathbf{g}(\mathbf{i})} \cdot \left. \frac{\partial g}{\partial c_{\mathbf{k}}} \right|_{\mathbf{x}=\mathbf{i}} = \tag{4.25}$$

$$= \boldsymbol{\beta}_3 \left(\mathbf{i}/h - \mathbf{k} \right) \sum_q b_{qj} \beta_3'(g(\mathbf{i}) - q) \tag{4.26}$$

$$\tag{4.27}$$

where $\mathbf{i} = (i, j)$, and the summation across q only needs to be performed within the support of β_3' of length 4. The second derivatives are

$$\frac{\partial^2 E}{\partial c_{\mathbf{k}} \partial c_{\mathbf{l}}} = 2 \sum_{\mathbf{i} \in I} \left(\frac{\partial e_{\mathbf{i}}}{\partial c_{\mathbf{k}}} \frac{\partial e_{\mathbf{i}}}{\partial c_{\mathbf{l}}} + e_{\mathbf{i}} \frac{\partial^2 e_{\mathbf{i}}}{\partial c_{\mathbf{k}} \partial c_{\mathbf{l}}} \right) \tag{4.28}$$

$$\frac{\partial^2 e_{\mathbf{i}}}{\partial c_{\mathbf{k}} \partial c_{\mathbf{l}}} = \left. \frac{\partial^2 \, {}^c f}{\partial^2 x} \right|_{\mathbf{x}=\mathbf{g}(\mathbf{i})} \frac{\partial g}{\partial c_{\mathbf{k}}} \frac{\partial g}{\partial c_{\mathbf{l}}} + \frac{\partial {}^c f}{\partial x} \underbrace{\frac{\partial^2 g}{\partial c_{\mathbf{k}} \partial c_{\mathbf{l}}}}_{0} \tag{4.29}$$

$$\frac{\partial^2 e_{\mathbf{i}}}{\partial c_{\mathbf{k}} \partial c_{\mathbf{l}}} = \boldsymbol{\beta}_3 \left(\mathbf{i}/h - \mathbf{k} \right) \boldsymbol{\beta}_3 \left(\mathbf{i}/h - \mathbf{l} \right) \sum_q b_{qj} \beta_3''(g(\mathbf{i}) - q) \tag{4.30}$$

which is easy to calculate because many terms can be reused from the calculation of the first derivatives.

Chapter 5

Fast Multidimensional Elastic Image Registration

5.1 Abstract

We present an algorithm for fast elastic multidimensional intensity-based image registration. It is fully automatic in its default mode of operation. In the case of hard real-world problems, it is capable of accepting expert hints in the form of soft landmark constraints. Much fewer landmarks are needed and the results are far superior compared to pure landmark registration.

Particular attention has been paid to the factors influencing the speed of this algorithm. The B-spline deformation model is shown to be computationally more efficient than other alternatives. We also present the concept of subspace optimization. As particular case, we consider representing the deformation in a sparse wavelet basis.

We demonstrate the evolution of the registration process for the task of registering 2D slices of anatomical MRI images of the brain warped with a known deformation. We present an application of the algorithm for registering ECD SPECT images for realignment of corresponding Xenon inhalation SPECT images. Another application involves analysing heart motion from a sequence of 2D ultrasound images.

5.2 Proposed algorithm

The algorithm presented here is a combination of ideas from preceding chapters. First, it is a generalization of the unidirectional registration algorithm from Chapter 4 to multiple dimensions. The deformation model has not only been generalized but also re-engineered for faster execution. The optimization algorithm has been changed to yield faster conver-

gence in the initial part of the optimization. Second, we recognize that for some difficult problems the fully automatic registration may fail. This is to say that it converges to a solution that minimizes the criterion but which does not correspond to the true correspondence of the underlying physical objects shown in the images. Hence, using the idea of landmarks from Chapter 3, we ask an expert to identify a small number of corresponding points in both images. We introduce virtual springs connecting the identified points and add a corresponding term (spring potential energy) to the data part of the criterion. This steers the algorithm towards the correct solution which would not have been found otherwise.

5.2.1 Organization of this Chapter

In the remainder of this section, we describe the concept of registration by minimization, the B-spline image model, the structure of the deformation model, and the difference measure. In Section 5.3 we justify our choice of B-splines as basis functions for the deformation model. We present the optimization method in Section 5.4, where we also describe the multiresolution strategy. Section 5.5 is devoted to the semi-automatic mode incorporating landmark information into the global criterion. We deal with implementation issues in Section 5.6 and present experiments and applications in Section 5.7.

5.2.2 Registration as minimization

The input images are given as two N-dimensional discrete signals $f_r(\mathbf{i})$ and $f_t(\mathbf{i})$ where $\mathbf{i} \in I \subset \mathbb{Z}^N$ and I is an N dimensional discrete interval representing the set of all pixel coordinates in the image. We will call them *reference* and *test* images, respectively. We suppose that the test image is a geometrically deformed version of the reference image, and vice versa. This is to say that the points with the same coordinates in the reference image $f_r(\mathbf{x})$ and in the warped test image $f_w(\mathbf{x}) = f_t^c(\mathbf{g}(\mathbf{x}))$ should correspond. Here f_t^c is a continuous version of the test image and $\mathbf{g}(\mathbf{x})$ is the deformation function to be identified. The two images f_r, f_w will not be identical because of noise and also because the assumption that there is a geometrical mapping between the two images is not necessarily correct. Therefore, we define the solution to our registration problem as the result of the following minimization:

$$\mathbf{g} = \arg\min_{\mathbf{g} \in G} E\big(f_r, f_t(\mathbf{g}(\mathbf{x}))\big) \tag{5.1}$$

where G is the space of all admissible deformation functions $\mathbf{g}$ and E the criterion being minimized. Graphically, the flowchart of our algorithm is shown in Figure 5.1.

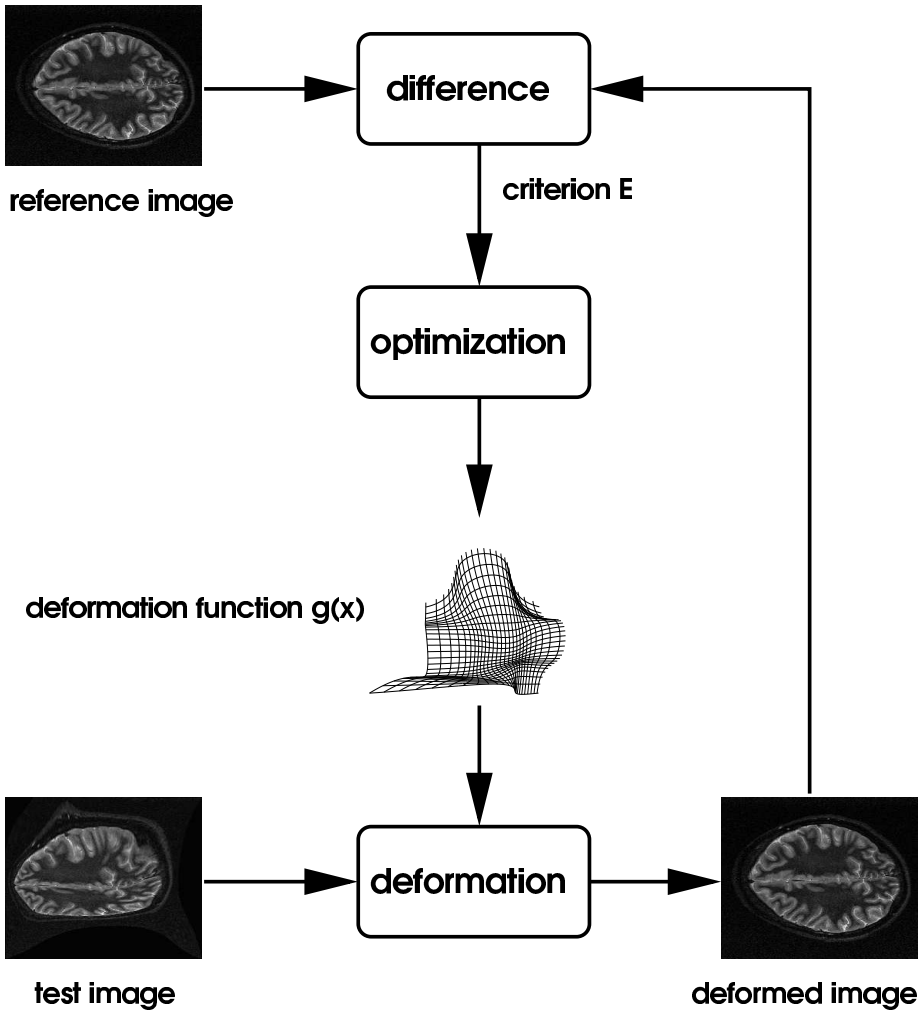


Figure 5.1: The flowchart of the registration algorithm. A warping function **g** is sought minimizing a difference between a reference image and a deformed version of the test image.

5.2.3 Image model

We suppose that the discrete test image is a sampled version of a continuously-defined image $f_t^c(\mathbf{x})$. That is to say

$$f_t(\mathbf{i}) = f_t^c(\mathbf{i}) \quad \text{for all} \quad \mathbf{i} \in I \subset \mathbb{Z}^N \tag{5.2}$$

We need to specify an image model to reconstruct the continuous images. In accordance with Chapter 4, we choose to interpolate the image using uniform B-splines:

$$f_t^c(\mathbf{x}) = \sum_{\mathbf{i} \in I_b \subset \mathbb{Z}^N} b_{\mathbf{i}} \beta_{n_i}(\mathbf{x} - \mathbf{i}) \tag{5.3}$$

where β_n is a tensor product of centered B-splines of degree n:

$$\beta_n(\mathbf{x}) = \prod_{k=1}^{N} \beta_n(x_k) \tag{5.4}$$

with $\mathbf{x} = [x_1, \ldots, x_n]$. In this way, we are representing by B-spline coefficients $b_{\mathbf{i}}$ the image initially given by its samples $f_t(\mathbf{i})$.

5.2.4 Deformation model structure

So far, we have considered the deformation function $\mathbf{g}$ to be an arbitrary admissible function $\mathbb{R}^N \to \mathbb{R}^N$. We will restrict it now to a family of functions described by a finite number of parameters $\mathbf{c}$. This transforms a variational problem (5.1) into a much easier finite-dimensional minimization problem. Moreover, the restriction of the family G of all possible functions $\mathbf{g}$ can already guarantee some useful properties, such as the regularity (smoothness) of the solution.

For the moment, we describe the deformation by a general linear model

$$\mathbf{g}(\mathbf{x}) = \mathbf{x} + \sum_{\mathbf{j} \in J} \mathbf{c}_{\mathbf{j}} \varphi_{\mathbf{j}}(\mathbf{x}) \tag{5.5}$$

where J is a set of parameter indexes and $\varphi_{\mathbf{j}}$ are the corresponding basis functions. Adding $\mathbf{x}$ in the above equation makes the set of zero parameters correspond to identity, which seems to be a reasonable behaviour. The choice of the functions $\varphi_{\mathbf{j}}$ will be discussed in Section 5.2.4.

5.2.5 Difference measure

Various measures E can be used to evaluate differences between the warped test image f_w and the reference image f_t. The fundamental requirement is that E should be minimal

when the images are aligned. This is generally impossible to guarantee unless the images are the same or known. Nevertheless, if we consider the images to be realizations of the same stochastical process with known properties, we can often find a criterion which is minimal in a well-defined statistical sense when the images are aligned. For example, if we suppose that f_r is a deformed (warped) version of f_t with i.i.d. (independent and identically distributed) Gaussian noise added to each pixel, then the criterion giving the optimal solution $\mathbf{g}$ in the ML (maximum likelihood) sense is the SSD (sum of squared differences) criterion

$$E = \frac{1}{\|I\|} \sum_{\mathbf{i} \in I} e_{\mathbf{i}}^2 = \frac{1}{\|I\|} \sum_{\mathbf{i} \in I} \big(f_w(\mathbf{i}) - f_r(\mathbf{i})\big)^2 \tag{5.6}$$

where the summation is taken over all pixels in the reference image. We have used this criterion for all our applications because it is fast to evaluate and yields smooth criterion surface which lends itself well to optimization. It proved to be robust enough, especially if preprocessing was used to equalize the image values (see Chapter 4 for an example).

In principle, there is no difficulty in extending our method for more sophisticated pixel-based similarity measures, such as information-based measures [97], especially mutual information [41], or weighted ℓ_p norms.

5.3 Choosing the deformation basis

The purpose of this section is to motivate our choice of (cubic) B-splines (Chapter 4) as the most adequate basis functions $\varphi_{\mathbf{j}}$ to represent the deformation in model (5.5). The alternative possibilities that come to mind are polynomials [32], harmonic functions [33, 34], radial basis functions (Chapter 3), and wavelets [55, 56, 98].

It is highly desirable to have as few basis functions as possible to contribute at each particular point, while keeping the approximation quality, of course. First, short basis functions have small overlap. This reduces the interdependency between the coefficients (parameters) and consequently makes the minimization problem easier to solve. Small overlap also makes the Hessian more sparse and therefore potentially faster to invert.

Second, the size of the support of the basis functions directly influences the speed of the calculation. To evaluate a function at N_{pix} points costs $O(N_{\text{pix}} N_{\text{supp}})$ operations, where N_{supp} is the number of functions contributing to a single point.[1] The cost of evaluating the gradient of the criterion with respect to the coefficients is higher but asymptotically equivalent. The cost of evaluating the Hessian is $O(N_{\text{pix}} N_{\text{supp}}^2)$ operations. (See also Section 5.6.1.)

[1] We assume that the cost of evaluating the basis function itself is constant since those can be precalculated.

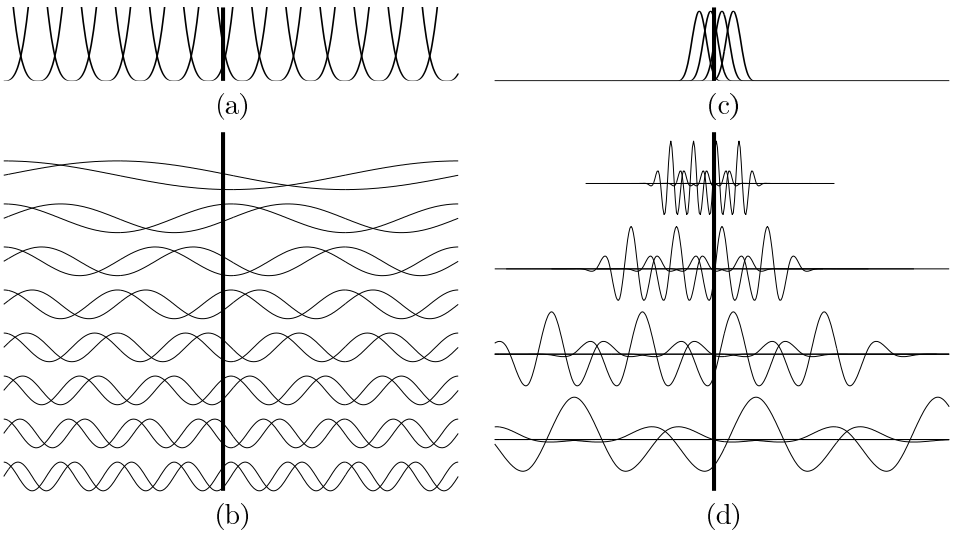


Figure 5.2: The basis functions involved in evaluating of the value of a 1D function at one point (denoted by a vertical line): radial basis functions *(a)*, harmonic functions *(b)*, B-splines *(c)*, wavelets *(d)*.

We depict the various candidate generating functions and their support in Figure 5.2. Except for the Fourier basis, we choose basis functions of the same degree (cubic), generating the same space. We see clearly that the least number of contributing functions (4) is in the B-spline case. This effect turns out to be even more dramatic in higher dimensions.

The reasoning above rules out the polynomials and radial basis functions because no fast algorithm is known for their evaluation and the brute-force evaluation is slow due to their long support. We decided against the harmonic (Fourier) basis functions because of their lack of localization (the fact that any two of them overlap). Note that this is the case even though the Fourier base is orthogonal in the classical (L_2) sense; here we weight the scalar product with arbitrary image dependent weights. (See Section 5.6.1 for a formula.) Another argument against the Fourier basis is that it cannot express linear functions (affine deformations). The only two remaining candidate basis are therefore B-splines and wavelets.

5.3.1 Splines versus wavelets

To make a fair comparison between B-spline and wavelet bases, we consider compactly supported cubic B-spline wavelets [99] spanning the same space as cubic B-splines.

First, let us analyze the task of evaluating the deformation at a single point. There are four participating B-splines alltogether while there are four participating B-spline wavelet at each level. Given that the complexity of one generating function evaluation is comparable (both are piecewise polynomial), B-splines win.

Second, to evaluating the deformation at a regular grid, we can use an iterative filterbank algorithm (such as Mallat's FWT [69]) in both cases. The number of iterations needed is the same, but the filters are longer in the wavelet case. Therefore, the B-spline representation is better again.

The complexity of evaluation of the gradient of the criterion corresponds to the complexity of the evaluation of the deformation because the same type of formula is involved (see Section 5.6.1) plus a constant cost independent on the deformation model.

In multiple dimensions, the differences are even more striking. This comparison can be refined but the overall conclusion remains the same: B-spline basis is much more computationally efficient than the equivalent wavelet basis.

5.3.2 B-spline deformation model

The B-spline deformation model is obtained by substituting a scaled version of the B-spline in (5.5)

$$\mathbf{g}(\mathbf{x}) = \mathbf{x} + \sum_{\mathbf{j} \in I_c \subset \mathbb{Z}^N} \mathbf{c}_\mathbf{j} \beta_{n_m} (\mathbf{x}/\mathbf{h} - \mathbf{j} + \mathbf{d}) \tag{5.7}$$

where n_m is the degree of splines used, $\mathbf{h}$ is the knot spacing and $\mathbf{d}$ the knot offset. This corresponds to placing the knots on a regular grid over the image. Similar to the image case, β_{n_m} here represents an N-dimensional tensor product of B-splines (see (5.4)). The principal difference from our previous work (Chapter 4) is that here we require the node spacing $\mathbf{h}$ to be integer. This restricts the generality of the model somewhat and also limits the liberty of scale changes we need for the multiresolution approach. However, the big advantage is that the B-spline values involved in the interpolation (5.7) repeat with period $\mathbf{h}$ when evaluating for integer values of $\mathbf{x}$. Therefore, the values of $\mathbf{g}$ are a discrete convolution of the $\mathbf{h}$-upsampled coefficient values $\mathbf{c}$ with the values of a B-spline evaluated with a sampling step $1/\mathbf{h}$. Moreover, the convolution can be performed in a separable manner.

To evaluate $\mathbf{g}$ at one point we would normally need to evaluate a B-spline for $N(n_m+1)$ different values and calculate a linear combination of the $(n_m+1)^N$ resulting products. However, as $\mathbf{h}$ is now integer, we can precalculate the B-spline values in advance. Thanks to the separability, we can evaluate $\mathbf{g}$ with the cost of only $N(n_m+1)$ multiplications per pixel, and this irrespective of the grid size.

The B-spline model has good approximation properties and is fast to evaluate. It is also physically plausible and scalable in the sense that a coarse level deformation can be represented exactly at finer levels (see also Section 5.6.4).

A remarkable feature of the B-spline model (5.7) is that the complexity of evaluating the deformation, and the gradient and the Hessian of the criterion, does not depend on the sampling step $\mathbf{h}$, or equivalently, on the number of parameters. This can be seen from the fact that one pixel in the image always contributes to a fixed number of gradient (or Hessian) components.

5.3.3 Subspace optimization

One of the potential advantages of a wavelet deformation model is the ability to selectively add finer basis functions in specific regions. In this way the deformation model can be made more flexible where enough data is available while remaining constrained elsewhere. To achieve the same effect, we propose to work with a B-spline representation at the finest level. The deformation is described using B-spline coefficients $\mathbf{c}$ that we constrain to live in some subspace, parametrized by a different set of coefficients $\mathbf{d}$. We have a linear relationship between the two sets of coefficients:

$$\mathbf{c} = \mathsf{W}\mathbf{d} \tag{5.8}$$

(Note that $\mathbf{c}$, $\mathbf{d}$ are here considered as 1D vectors of all parameters.) The matrix W is just a symbol, it should not actually be stored explicitely, because full matrix multiplication would destroy the efficiency we are after.

The gradient with respect to $\mathbf{d}$ is

$$\nabla_{\mathbf{d}} E = \mathsf{W}^T \nabla_{\mathbf{c}} E \tag{5.9}$$

The advantage of our approach is computational efficiency. Note that if W is a wavelet transform, its transpose W^T can be calculated by reverting the dataflow in the filterbank algorithm, using the same filters. After calculating the gradient using (5.26), we only need to perform W^T on the gradient $\nabla_{\mathbf{c}} E$ which is much smaller than the image.

The relation (5.8) is very general and comprises much more than just wavelets. All cases of optimization in linear subspaces can be cast into this framework. In principle, we could use tools like principal component analysis (PCA) to analyze most likely deformation modes and search only among them [100]. We may also use (5.8) to remedy for what may be perceived as a limitation of our spline model: the uniformity of the grid. With the subspace approach, we can selectively coarsen the deformation model, while keeping the low-level uniform control grid for computation. In many cases, this will be the most efficient way to approach the task.

5.4 Optimization strategy

5.4.1 Optimization algorithms

Given our deformation model (5.5), the minimization (5.1) becomes

$$\mathbf{c} = \arg\min_{\mathbf{c}} E_c \tag{5.10}$$

where E_c is the complete criterion (see e.g., (5.33)) to be optimized and $\mathbf{c}$ are the parameters describing the deformation function (5.7). This is a classical finite-dimensional non-linear optimization problem for which numerous algorithms exist [50]. We have almost exclusively considered local iterative algorithms, which is well justified thanks to the smoothness of our criterion. (One example of an algorithm which is not local is the genetic algorithm.)

All these algorithms can be cast in a common framework. They start with an initial estimate $\mathbf{c}^{(0)}$. Then a sequence of steps is repeated until convergence. At each step we take the actual estimate $\mathbf{c}^{(i)}$ and calculate a proposed update $\Delta\mathbf{c}^{(i)}$. We then evaluate the criterion at the new proposed point, $E_c(\mathbf{c}^{(i)} + \Delta\mathbf{c}^{(i)})$, and test if the criterion value has improved. If the step is successful, that is, if $E_c(\mathbf{c}^{(i)} + \Delta\mathbf{c}^{(i)}) < E(\mathbf{c}^{(i)})$, then the proposed point is accepted, $\mathbf{c}^{(i+1)} = \mathbf{c}^{(i)} + \Delta\mathbf{c}^{(i)}$. Otherwise, new, more conservative update $\Delta\mathbf{c}^{(i)}$ is calculated, and the test is repeated.

We tested the following four algorithms:

Gradient descent with feedback step size adjustment

The simplest of this class of algorithms is *gradient descent* (GD). Its update rule is

$$\Delta\mathbf{c}^{(i)} = -\mu\nabla_{\mathbf{c}} E_c(\mathbf{c}^{(i)}) \tag{5.11}$$

where μ is the step size which needs to be adjusted. We used a simple automatic adjustment method. After a successful step, μ is multiplied by μ_f, otherwise it is divided by μ'_f. We start with $\mu = 1$ and use $\mu_f = 10$, $\mu'_f = 15$.

Gradient descent with quadratic step size estimation

The single most critical point of the preceding algorithm is the determination of the step size μ. We have devised a formula yielding an estimation of the optimal step size which performs better than the crude adjustment method described above. We approximate the criterion around $\mathbf{c}^{(i)}$ as

$$E(\mathbf{c}^{(i)} + \mathbf{x}) = E(\mathbf{c}^{(i)}) + \mathbf{x}^T\nabla_{\mathbf{c}} E_c(\mathbf{c}^{(i)}) + \alpha\|\mathbf{x}\|^2 \tag{5.12}$$

The minimum of this model along $\mathbf{c}^{(i+1)} = \mathbf{c}^{(i)} - \mu\nabla_{\mathbf{c}} E_c(\mathbf{c}^{(i)})$ (see (5.11)) is achieved for the step size

$$\mu^* = \frac{1}{2} \frac{||\mathbf{c}^{(i)} - \mathbf{c}^{(i-1)}||^2}{E(\mathbf{c}^{(i-1)}) - E(\mathbf{c}^{(i)}) + \left\langle \mathbf{c}^{(i)} - \mathbf{c}^{(i-1)}, \nabla_{\mathbf{c}} E_c(\mathbf{c}^{(i)}) \right\rangle} \tag{5.13}$$

As a fallback strategy, if μ^* does not yield an improvement over $E(\mathbf{c}^{(i)})$, we use the previous step size divided by μ'_f, as above.

Conjugated gradient

The *conjugated gradient* algorithm chooses its descent directions to be mutually conjugate so that moving along one does not spoil the result of previous optimizations. Its update rule is

$$\Delta\mathbf{c}^{(i)} = -\mu \left(\frac{\left\langle \nabla_{\mathbf{c}} E_c(\mathbf{c}^{(i)}) - \nabla_{\mathbf{c}} E_c(\mathbf{c}^{(i-1)}), \nabla_{\mathbf{c}} E_c(\mathbf{c}^{(i)}) \right\rangle}{||\nabla_{\mathbf{c}} E_c(\mathbf{c}^{(i-1)})||^2} \nabla_{\mathbf{c}} E_c(\mathbf{c}^{(i-1)}) + \nabla_{\mathbf{c}} E_c(\mathbf{c}^{(i)}) \right) \tag{5.14}$$

To work well, the μ has to be chosen optimally. Therefore, at each step, we need to run another internal one-dimensional minimization routine which finds the optimal step length μ in the direction given by the above formula. We implemented the internal minimization using Brent's algorithm [50], essentially using parabolic interpolation and golden-ratio search.

Marquardt-Levenberg

The most effective algorithm in the sense of the number of iterations was a regularized Newton method inspired by the *Marquardt-Levenberg* algorithm (ML), as in Chapter 4. The update rule is

$$\Delta\mathbf{c}^{(i)} = -\left(\nabla^2_{\mathbf{c}} E_c(\mathbf{c}^{(i)}) + \lambda \mathsf{I}\right)^{-1} \nabla_{\mathbf{c}} E_c(\mathbf{c}^{(i)}) \tag{5.15}$$

where the regularization factor λ can steer the algorithm between pure Newton method (for small λ) and pure gradient descent (for big λ). The λ starts at 1 and is adjusted automatically, divided by $\lambda_f = 10$ after a successful step and multiplied by $\lambda'_f = 15$ after a failed step. It might not always be economical, feasible, or even desirable, to calculate the true and complete Hessian matrix $\nabla^2_{\mathbf{c}} E_c$. The algorithm will converge, albeit perhaps more slowly, even if we use just an approximation of it. See also Section 5.6.1.

5.4.2 Choosing the best optimizer

Ultimately, we want the optimizer which yields the optimal $\mathbf{c}$ within desired precision in the least amount of time. The choice is dependent on the particular problem, namely on the smoothness of the criterion surface and the computational complexity of the evaluation of the criterion, its gradient and Hessian. Finally, it depends also on the desired precision of the result.

We performed a series of experiments to determine which of the four optimizers and their variants works best for our task. We quickly ruled out the conjugated gradient algorithm. As many evaluations are needed for its internal minimization to converge, the conjugated gradient algorithm is only advantageous if the evaluation of E_c is at least $10 \sim 100$ faster than the evaluation of the gradient ∇E_c. This is not the case, so consequently, the unidimensional line searches are wasteful in the terms of the number of iterations. We also quickly determined that quadratic step size estimation for gradient descent minimization was almost always better than the feedback adjustment.

Among Marquardt-Levenberg (ML) algorithms, we found the performance to be superior when using the full Hessian. All approximations of the Hessian included a performance penalty which was normally not compensated by the reduced computation cost.

This leaves us with two alternatives: the gradient descent with optimal step-size estimation or the Marquardt-Levenberg (ML) like algorithm. This choice is rather delicate. As we pointed out in Chapter 4, the ML algorithm is converging quadratically near the optimum. This advantage can be offset by the additional expense of evaluating the Hessian matrix and its inverse. Nevertheless, when a small number of parameters is sought for, the criterion is smooth, and high precision is needed, the ML algorithm is the fastest. This is demonstrated in Figure 5.3.

However, in some cases these assumptions are not fulfilled. Often the deformation is described by several hundreds of parameters and the calculation of the Hessian matrix takes of the order of magnitude more time than the calculation of the gradient. See Table 5.1 for a typical relationship between the computation times.

The complexity of Hessian matrix inversion (the solution of the equation (5.15) for $\Delta\mathbf{c}^{(i)}$) grows with the cube of the number of parameters and becomes dominant for large number of parameters (several hundreds). We also realize that the intrinsic resolution of the images registered is limited and that there is no use trying to find the deformation function with higher accuracy. Thus, we need an algorithm which performs well in the first part of the optimization process, far from the optimum. Here the Hessian information is not precise enough to improve the convergence and experiments show that the ML and gradient descent algorithms perform about equally in the terms of the number of iterations. For an example, see Figure 5.4. As the time needed for one iteration of the gradient descent is much smaller with respect to one ML iteration, we decided that the gradient descent (GD) algorithm (with the quadratic step size estimation) was the most suitable for our applications, mostly because of its speed. One additional pleasant property of the GD algorithm in our context is its tendency to leave uninfluential

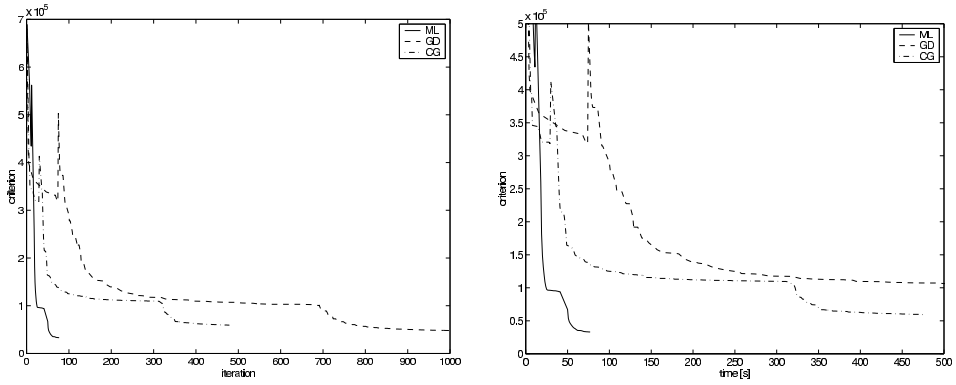


Figure 5.3: Comparison of gradient descent, conjugated gradient, and Marquardt-Levenberg optimization algorithm performances when registering SPECT images. The graphs give the value of the finest-level SSD criterion of all successful (i.e., criterion-decreasing) iterations as a function of the number of criterion evaluations and execution time. The peaks are caused by transitions between resolution levels.

Table 5.1: Relative times to evaluate the criterion E, its gradient ∇E, and Hessian $\nabla^2 E$, for a volume of $64 \times 64 \times 17$ voxels approximated by cubic splines, as a function of the spline degree n_m used to model the deformation and the size of the parameter grid n_c. (The time to evaluate E is currently around 1 s.)

n_m	2	3	3
n_c	$3 \times 6 \times 6 \times 6$	$3 \times 6 \times 6 \times 6$	$3 \times 4 \times 4 \times 4$
E	1.0	1.4	1.4
E, ∇E	2.3	2.9	2.9
E, ∇E, $\nabla^2 E$	10.9	48.7	48.8

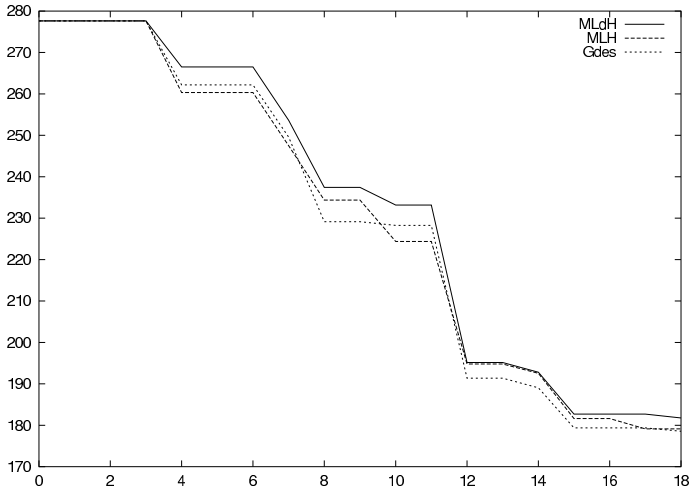


Figure 5.4: The evolution of the SSD criterion during the first 18 iterations when registering Lena image artificially deformed with $2 \times 4 \times 4$ cubic B-spline coefficients and maximum displacement of about 30 pixels, without multiresolution. The optimizers used were: Marquardt-Levenberg with full Hessian (MLH), Marquardt-Levenberg with only the diagonal of the Hessian taken into account (MLdH), and gradient descent (Gdes). The deformation was recovered in all cases with accuracy between 0.1 and 0.01 pixels.

coefficients intact, unlike the ML algorithm which in its quest for the most gain in each step often changes the uninfluential coefficients excessively. This leads to unplausible solutions. Therefore, less regularization is often needed for the GD algorithm.

However, the choice of the algorithm might need to be revised if a different class of problems is to be solved.

5.4.3 Multiresolution

As in Chapter 4, we use multiresolution for both image model and the deformation model. We start with the coarsest resolution versions of both and alternatively refine the image and the deformation model every time convergence is reached, until the finest level. The second possibility consists in refining both the image and the deformation model at the same time. This saves some transition costs but the inter-level changes are bigger and the overall performance is therefore usually worse.

The coarse versions of images are generated using reduction. Conversely, coarse level solutions are extrapolated to finer levels using expansion. See Section 5.6.4 for

implementation details.

5.5 Semi-automatic registration

We realize that although the multiresolution leads to a very robust registration algorithm, there are still cases when it is mislead by an apparent similarity of features which do not correspond physically. Therefore, we developed an extension of the algorithm which can use expert hints to be put on track towards the right solution. The hints come in the form of a set of landmarks. The landmarks are manually identified pairs of corresponding points in the two images (see also Chapter 3).

As an example, we tried to register two ultrasound images from different parts of the cardiac cycle. (See Section 5.8.2 for details.) The unsupervised registration correctly registered some of the structures but missed others, see Figure 5.5, top. We asked an expert to identify several landmarks in both images (Figure 5.5, middle part). Using this minute hint, the semi-automatic algorithm could recover a plausible deformation, even though the landmark information alone (using thin-plate splines) would not have been enough (Figure 5.5, bottom). The warped version of the test image (on the right in the middle row of Figure 5.5) and the deformation function is shown in Figure 5.6

5.5.1 Virtual springs

The landmark information is incorporated in the automatic process using the concept of virtual strings, tying each pair of corresponding points together. We augment the data part of the criterion E with a term E_s, corresponding to the potential energy of the springs, and minimize the sum of the two $E_c = E + E_s$. The spring term is:

$$E_s = \sum_{i=1}^{S} \alpha_i \left\| \mathbf{g}(\mathbf{x}_i) - \mathbf{z}_i \right\|^2 \tag{5.16}$$

where S is the number of springs, α_i are weighting factors corresponding to their stiffnesses, and $\mathbf{x}_i$, resp. $\mathbf{z}_i$, are the landmark positions in the reference, resp. test images. The contribution of E_s to the gradient of E_c can be calculated from (5.7):

$$\left[\frac{\partial E_s}{\partial c_{\mathbf{j},1}} \cdots \frac{\partial E_s}{\partial c_{\mathbf{j},N}} \right]^T = 2 \sum_{i=1}^{S} \alpha_i \left(\mathbf{g}(\mathbf{x}_i) - \mathbf{z}_i \right) \beta_{n_m}(\mathbf{x}/\mathbf{h} - \mathbf{j} + \mathbf{d}) \tag{5.17}$$

The spring factors α_i control the influence of the particular landmark pair. We propose to start with all $\alpha_i = 1.0$ and adjust them experimentally to get the most satisfactory results. In the example in Figure 5.6, we used $\boldsymbol{\alpha} = (1.0, 0.1, 0.1, 0.1)$, with the most weight (1.0) given to the rightmost landmark.

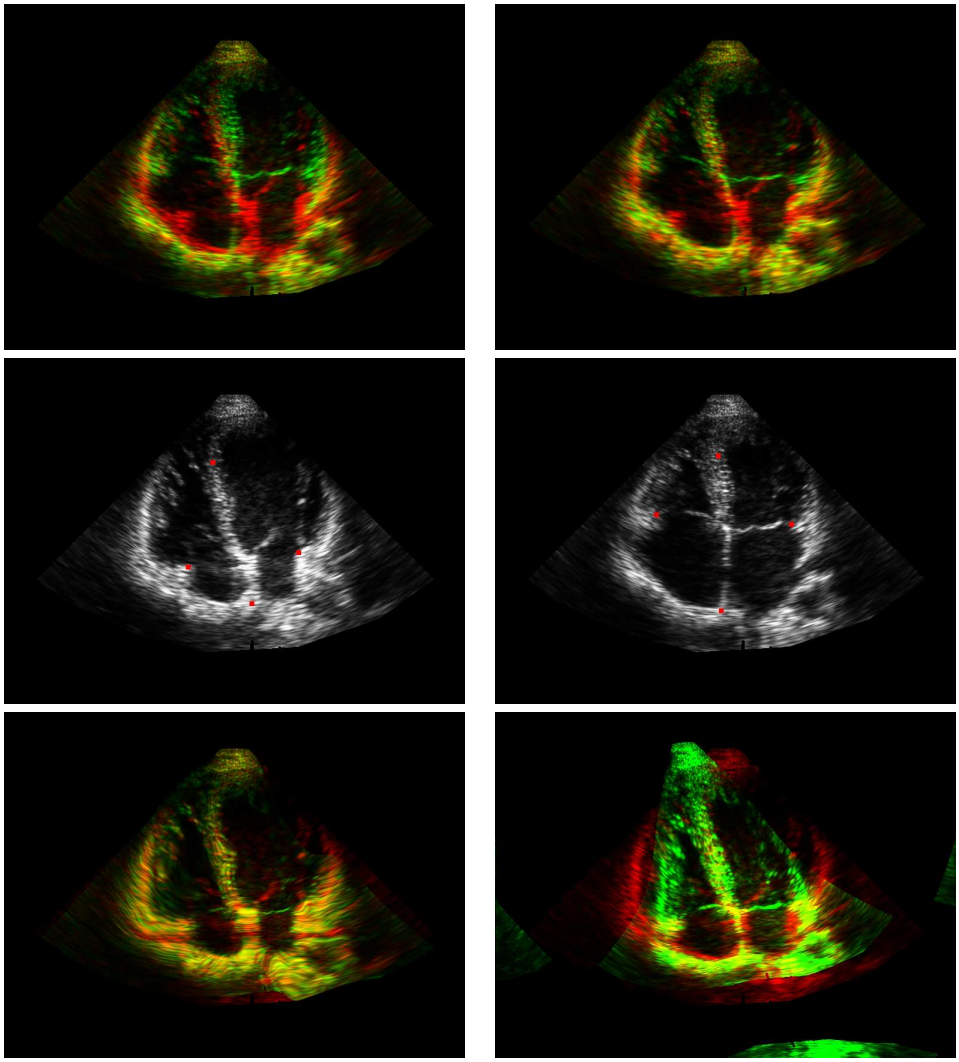


Figure 5.5: The difference between two ultrasound images from different parts of the cardiac cycle. One is shown in green, the other in red, yellow areas indicate perfect overlap. The original difference before registration *(top left)* is only partially corrected by the unsupervised registration *(top right)*. Misalignment of several structures is clearly visible. The middle row shows the two images with superimposed landmarks. The precision of the semi-automatic registration result can be easily appreciated *(bottom left)* in comparison with the purely landmark-based thin-plate spline registration *(bottom right)*.

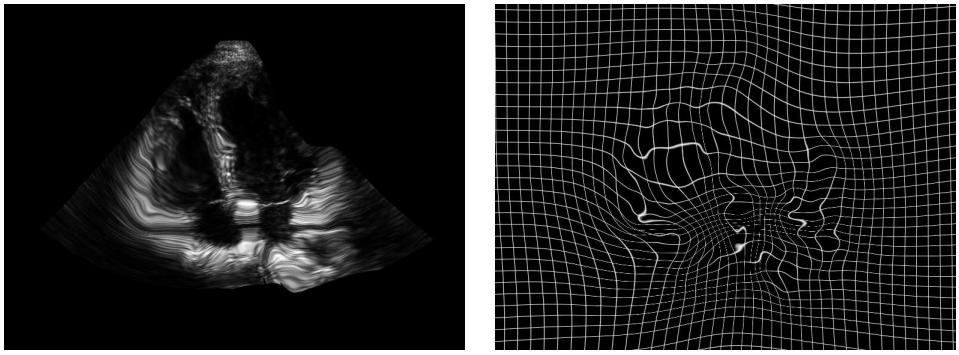


Figure 5.6: The deformed test image from Figure 5.5 *(left)* and the correspondence function *(right)* as found by the semi-automatic algorithm.

5.6 Implementation issues

The purpose of this section is to describe some specific aspects of our implementation. These are mostly independent of the main philosophy of the algorithm but can have a major impact on its performance.

5.6.1 Explicit derivatives

For the optimization algorithm, we need to calculate the partial derivatives of E_c from (5.33), as they form the gradient vector $\nabla_{\mathbf{c}} E_c(\mathbf{c}^{(i)})$ and the Hessian matrix $\nabla^2_{\mathbf{c}} E_c(\mathbf{c}^{(i)})$. Starting from equation (5.6), we obtain

$$\frac{\partial E}{\partial c_{\mathbf{j},m}} = \frac{1}{\|I\|} \sum_{\mathbf{i} \in I_b} \frac{\partial e_{\mathbf{i}}}{\partial f_w(\mathbf{i})} \left. \frac{\partial f_t^c(\mathbf{x})}{\partial x_m} \right|_{\mathbf{x}=\mathbf{g}(\mathbf{i})} \frac{\partial g_m(\mathbf{i})}{\partial c_{\mathbf{j},m}} \tag{5.18}$$

and the second derivatives

$$\frac{\partial^2 E}{\partial c_{\mathbf{j},m} \partial c_{\mathbf{k},n}} = \frac{1}{\|I\|} \sum_{\mathbf{i} \in I_b} \left(\frac{\partial^2 e_{\mathbf{i}}}{\partial f_w(\mathbf{i})^2} \frac{\partial f_t^c}{\partial x_m} \frac{\partial f_t^c}{\partial x_n} + \frac{\partial e_{\mathbf{i}}}{\partial f_w(\mathbf{i})} \frac{\partial^2 f_t^c}{\partial x_m \partial x_n} \right) \frac{\partial g_m}{\partial c_{\mathbf{j},m}} \frac{\partial g_n}{\partial c_{\mathbf{j},n}} \tag{5.19}$$

From (5.6) defining the SSD criterion, we get

$$\frac{\partial e_{\mathbf{i}}}{\partial f_w(\mathbf{i})} = 2\big(f_w(\mathbf{i}) - f_r(\mathbf{i})\big) \tag{5.20}$$

$$\frac{\partial^2 e_{\mathbf{i}}}{\partial f_w(\mathbf{i})^2} = 2 \tag{5.21}$$

The derivative of the deformation function (5.7) is simply

$$\frac{\partial g_m}{\partial c_{\mathbf{j},m}} = \beta_{n_m}\left(\mathbf{x}/\mathbf{h} - \mathbf{j}\right) \tag{5.22}$$

The deformation model is linear and all its second derivatives are therefore zero; that is the reason for the simplicity of (5.19). The partial derivatives of f_t^c in (5.18,5.19) can be calculated from (5.3) as a tensor product

$$\frac{\partial f_t^c}{\partial x_m} = \sum_{\mathbf{k}\in I} b_{\mathbf{k}} \beta'_{n_i}(x_m) \prod_{\substack{l=1 \\ l\neq m}}^{N} \beta_{n_i}\big(g_l(x_l)\big) \tag{5.23}$$

Second-order partial derivatives of f_t^c are obtained in a similar fashion.

5.6.2 Hessian approximation

Because the evaluation of the Hessian matrix from (5.19) is costly, several modifications have been devised. The Marquardt-Levenberg approximation assumes that the term (5.20) is negligibly small or that it sums to zero on the average. This reduces (5.19) to

$$\frac{\partial^2 E}{\partial c_{\mathbf{j},m} \partial c_{\mathbf{k},n}} = \frac{2}{||I||} \sum_{\mathbf{i}\in I_b} \frac{\partial f_t^c}{\partial x_m} \frac{\partial f_t^c}{\partial x_n} \frac{\partial g_m}{\partial c_{\mathbf{j},m}} \frac{\partial g_n}{\partial c_{\mathbf{j},n}} \tag{5.24}$$

Another simplification is to consider only diagonal terms $\partial^2 E / \partial c_{\mathbf{j},m}^2$. Obviously, this diagonal Hessian approximation only makes sense if the basis functions $\varphi_{\mathbf{j}}$ do not overlap too much. This is another argument for the B-spline model. Each such approximation makes the evaluation faster at the expense of precision which may result in slower convergence. Whether it is advantageous to use some approximation depends on many factors, including the size and the character of the data.

5.6.3 Gradient calculation as convolution

Here too the use of an integer step size $\mathbf{h}$ leads to computational savings. Substituting (5.22) into (5.18) gives

$$\frac{\partial E}{\partial c_{\mathbf{j},m}} = \sum_{\mathbf{i}\in I_b} \frac{\partial e_{\mathbf{i}}}{\partial f_w(\mathbf{i})} \left.\frac{\partial f_t^c(\mathbf{x})}{\partial x_m}\right|_{\mathbf{x}=\mathbf{g}(\mathbf{i})} \beta_{n_m}\left(\mathbf{i}/\mathbf{h} - \mathbf{j}\right) \tag{5.25}$$

which can be transformed into a discrete convolution

$$\left\{\frac{\partial E}{\partial c_{\mathbf{j},m}}\right\}_{\mathbf{j}} = \sum_{\mathbf{i}} w(\mathbf{i}) b(\mathbf{j}\cdot\mathbf{h} - \mathbf{i}) = (w * b)_{\downarrow \mathbf{h}} \tag{5.26}$$

where we have substituted w for the first two terms in (5.25) and $b(\mathbf{q}) = \beta_{n_m}(-\mathbf{q}/\mathbf{h})$. The convolution kernel b is separable and the convolution (5.26) can be calculated as a sequence of N unidimensional convolutions $\left((w * b_1)_{\downarrow h_1} * \ldots b_n\right)_{\downarrow h_N}$. Because of the downsampling, calculating one output value at step k consists of a scalar product with a filter b_k of effective length $(n_m+1)h_k$ (the support of a B-spline) and shifting this filter by h_k.

5.6.4 Multiresolution spline representation

To deploy the multiresolution strategy (see Section 5.4.3), we need to specify expansion and reduction operators. We will use the same approach for both the deformation model and the image model.

Let us consider here a 1D signal represented in a B-spline space

$$f(x) = \sum_i c_i \beta_n(x - i) \tag{5.27}$$

The expansion operator $\mathcal{E}$ yields a twice expanded version of f which is also a spline

$$f_e = \mathcal{E}f, \quad f_e(x) = f(x/2) = \sum_i d_i \beta_n(x - i) \tag{5.28}$$

with coefficients d_i given by

$$d = c_{\uparrow 2} * u^n \tag{5.29}$$

where $c_{\uparrow 2}$ denotes upsampled version of c and u^n is a symmetrical binomial filter defined in the z-domain as

$$U^n(z) = \frac{(1+z)^{n+1}}{2^n} z^{-(n+1)/2} \tag{5.30}$$

The twice reduced signal $f(2x)$ cannot be represented as a spline with knots at integers. We need to resort to approximation and we have chosen the L_2 optimality as described in [101]. The reduction operator $\mathcal{R}$ will yield a projection (denoted P_1) in the original spline space with step size 1.

$$f_r = \mathcal{R}f, \quad f_r(x) = P_1 f(2x), \quad f_r(x) = \sum_i e_i \beta_n(x - i) \tag{5.31}$$

The spline coefficients e_i are calculated as

$$e = (\mathring{h} * c)_{\downarrow 2} * b^{-(2n+1)} \tag{5.32}$$

with prefilter $\mathring{h} = b^{2n+1} * u^n$, where b^{2n+1} is the sequence of sampled values of a B-spline of degree $2n+1$, $b^n(i) = \beta_n(i)$. Finally $b^{-(2n+1)}$ is the inverse filter to b^{2n+1} and the convolution can be handled by recursive filtering [4, 5].

Because $\mathcal{R}$ is a projection complementary to $\mathcal{E}$, we have the projection identity $\mathcal{R}\mathcal{E}f = f$. Extension of both operators to multiple dimensions is trivial thanks to separability.

5.6.5 Consequences of finite support

All what we said so far about expansion and reduction holds for infinite signals. To adapt the method for finite signals, we considered the following requirements: the expansion must be exact in the continuous sense, the projection identity must hold, reduction followed by expansion must conserve the length of the signal, and as much information as feasible should be conserved. These requirements are useful to guarantee the best possible use of the coarse-grid results at the fine-grid level and are absolutely indispensable for multigrid minimization.

Traditionally, one represents the signal with exactly one coefficient per sample and assumes that the signal outside the region of interest follows some known pattern, such as periodicity, or mirror-on-boundary conditions. We take the mirror-on-boundary condition as an example, but the same kind of problems appear for other boundary conditions, too. First, the signal is forced to be symmetric and thus flat at boundaries. Second, the boundary conditions for both expansion and reduction are only conserved for odd number of samples, otherwise the mirror position needs to change. Third, varying the length of the signal by one does not change the length of the reduced version which makes it impossible to recover the original length by expansion.

Because of these considerations, we decided to dissociate the number of B-spline coefficients from the length of the interest region. Initially, we extend the signal by $\lceil (n-1)/2 \rceil$ samples at each extremity which allows us to represent any spline without constraints. We imagine that the boundaries of our signal are at the first and last of our coefficient list and we never move them when expanding, although the effective length (region of interest) might be smaller. In this way, expansion is always exact while it adds extra knots at each end. Reducing expanded signal recovers exactly the original. When reducing other signals, we need to extend them to be able to use our efficient filtering technique. We choose to use the mirror-on-boundary conditions.

5.6.6 Image size change

The only trick when expanding and reducing the images is to adapt the deformation function accordingly. This is easily accomplished by multiplying the coefficients by 2

when expanding and 0.5 when reducing. Thanks to our choice of the expansion and reduction operators, the origin of the image does not change.

5.6.7 Fast spline calculations

It is essential to take full advantage of the properties of splines. First, specialized routines are used to calculate the values of a B-spline of a specific order using a minimum number of operations. Second, as we are using tensor products of B-splines as our basis functions, many operations can be performed in a separable fashion, reducing the complexity of operations from $O(k^N)$, where N is the number of dimensions and k the size of the data, to $O(kN)$. This is the case for the prefiltering step required to find the B-spline coefficients, and also for the interpolation of values of a function given by its B-spline coefficients. A related general optimization technique called 'common subexpression elimination' calls for precomputing all expressions used more than once and for moving repeated calculations outside loops. Third, a compact support of B-splines makes many of the infinite sums in the expressions given earlier reduce to sums over just a small number of elements.

5.6.8 Stopping criterion

To get a fast optimization algorithm, particular attention has to be paid to the stopping criterion. This holds for both gradient descent and ML. Classically, the relative and absolute improvement of the criterion value is compared with a fixed threshold [50]. The comparison only has sense after the criterion decreased, that is to say, after a successful step. Consequently, after jumping close to the solution, many iterations are wasted on decreasing the step size (or increasing the regularization factor), only to confirm the convergence.

In problems of complicated topology, the converse can happen. After a long series of unsuccessful steps, the subsequent decrease of the criterion might be small only because the step size is small. In such cases it is better to stop only after several successful steps, instead of right after the first one.

For our class of problems, we found it to be advantageous to base the stopping criterion on the changes $\Delta\mathbf{c}$ of parameter values. We stop when the step size falls below an *a priori* given threshold. The size of a step that fails gives an indication of the accuracy of the result. We might assume that the minimum is not much farther than this step size. The threshold is easy to set as we can relate it directly to the pixel size. Typically, we would use the threshold of $10^{-1} \sim 10^{-3}$ pixels.

5.6.9 Masking

A substantial gain in speed comes from considering only important pixels when calculating the data criterion (5.6) and its derivatives. In a typical image, there are many

Figure 5.7: Example of a mask selecting 10 % of the pixels with the highest gradient values for the Lena image.

background pixel that do not contribute significantly to the criterion. It is possible to determine an *a priori* mask of significant pixels, for example $10 \sim 50\,\%$ of the total number of pixels, and to consider only those pixels in subsequent calculations. Inspecting (5.18) reveals that the contributions of individual pixels to the change of the criterion is directly proportional to the amplitude of the directional derivatives at the respective points. Therefore, a reasonable strategy is to construct the mask by thresholding the gradient of the image. An example of such a mask is given in Figure 5.7. We recalculate the mask to each level; this is analogous to using a smoothed version of the gradient.

5.6.10 Invertibility and regularization

In some applications, it is useful to add an extra regularization term E_r to the difference measure E, and to look for a minimum of the combined criterion

$$E_c = E + \gamma E_r \tag{5.33}$$

The factor E_r is used to make the solution well-posed, or to privilege likely solutions based on our *a priori* knowledge.

First we consider a penalty term designed to enforce the invertibility of the deformation, generalizing the concept from Chapter 4. Its motivation comes from the fact that if the Jacobian $\det(\nabla_{\mathbf{x}}\mathbf{g})$ is positive everywhere, then the deformation $\mathbf{g}$ is locally invertible. Evaluating this constraint at pixel coordinates and converting the strict constraints into soft ones using a barrier function yields the following penalty term

$$E_p = \sum_{\mathbf{i}\in I} \mathrm{e}^{-\alpha \det(\nabla_{\mathbf{x}}\mathbf{g}(\mathbf{i}))} \tag{5.34}$$

Experience shows, that for typical data, this term is never important at the solution point (to which the optimization converges). It mostly becomes useful at the beginning of the optimization process when the trial points vary a lot, especially with some optimizers. In such cases, the penalty term forces the algorithm to stay in the region of invertible deformations.

Depending on the particular task and the expected properties of the solution, various regularization terms can be used. We investigated for example stabilizer penalizing non-linear deformations

$$E_t = \int \sum_{k,l,m=1}^{N} \left(\frac{\partial^2 g_k}{\partial x_l \partial x_m} \right)^2 \mathrm{d}\mathbf{x} \tag{5.35}$$

and a very simple norm measuring the distance of $\mathbf{g}$ from identity through the coefficients $\mathbf{c}$

$$E_d = \sum_{\mathbf{j}} \|\mathbf{c_j}\|^2 \tag{5.36}$$

When the corresponding weight γ is small, the regularization mainly smoothes the deformation function in places where little information is present in the images. As it gets bigger, the regularization gradually overrides the data term and the deformation tends towards a smooth function in the sense of the particular regularization.

An alternative to regularization is the virtual spring mechanism described in Section 5.5, which we found to be preferable for our applications.

5.7 Experiments

This section presents a series of experiments in controlled environment to help in assessing the accuracy, speed, and robustness of our algorithm.

We show the SSD criterion E we minimize, and also a *warping index*

$$\varpi = \sqrt{\frac{1}{||R||} \sum_{\mathbf{i} \in R} ||g(\mathbf{i}) - g^*(\mathbf{i})||^2} \tag{5.37}$$

a mean geometric error between the true (g^*) and the recovered (g) deformation. The mean is only calculated over the part (R) of the image containing useful data (the object); the corresponding mask R can be seen in Figure 5.8, bottom left.

5.7.1 Registration of MRI brain slices

To get an idea of the working of the algorithm, we show its performance when recovering an artificially generated deformation of a 2D slice of an anatomical spin-echo MRI volume of the brain.[2] We use here synthetic images (artificially deformed) because the knowledge of the ground truth permits us to better judge the performances of the algorithm. However, registering real MRI image pairs has many practical applications, including generating atlases, intersubject or intrasubject comparisons, see also Chapter 1.

The original image of size 256×256 pixels is shown in Figure 5.8, top left. We use a spline grid controling the deformation with one knot for every 32 pixels. We warp the image with a deformation belonging to the warping space and consisting of a global vertical shift of 3 pixels, additional vertical shift of 10 pixels in one region of the image, and a horizontal shift of about 6 pixels in another region. The warped image is superimposed on the original in Figure 5.8, top right. Then the automatic registration algorithm is run. We use cubic splines to represent the deformation. The stopping threshold is set to 0.5 pixels for all levels except the last, where we set it to 0.1 pixels. The deformation recovered was used to warp again the original image. Its warped version is shown superimposed on the images warped with the true deformation in Figure 5.8, bottom right. We note that the deformation was well recovered, there is no perceptible difference.

The spatial distribution of the resulting geometrical error is shown in Figure 5.10. We see that the error is concentrated in areas with little detail in the image, the maximum error there is 1.5 pixels. In other regions, subpixel accuracy is achived; the mean error (warping index) over all brain is about 0.4 pixels. This is important when interpreting the warping index; it is a mean over a large region, including some parts with little texture. In general, high-contrast regions such us edges are resolved much more precisely than indicated by the value of ϖ, while the agreement in the zones with low-contrast will

[2] Author's brain. Images courtesy of Arto Nirkko from Inselspital Hospital, Bern, Switzerland.

be worse and often only coincidental, since there is little or no information to guide the algorithm.

The evolution of the optimization can be studied from the graphs in Figure 5.9. We observe the steady and correlated descent of the observable criterion being optimized (E) and of the warping index (ϖ), the quantity measuring the quality of the registration. The steep descent of the criterion in the initial phase of the optimization justifies our choice of the optimization algorithm, while the slow progress close to the end of the optimization illustrates the difficulty of the problem. The jumps in the curves are caused by the transitions between levels of the multiresolution progression and their relative insignificance manifests the accuracy of the spline model.

The warping index value changes from 4.62 at the beginning of the optimization until 0.39 at the end, while the criterion E descends from 695 at the beginning to 25.1 at the end. Note that, the final values of both E and ϖ depend strongly on the preset stopping threshold, which in turn influences the optimization time. The threshold value is a subjective compromise between the accuracy and computation time. It is perfectly possible to stop optimizing only after 7 s and skip the finest resolution level altogether, if the precision of $\varpi = 0.7$ pixels is acceptable. On the other hand, after about 4 more minutes of iteration, the error ϖ descends to less than 10^{-4} pixels. However, in this author's opinion, such super subpixel accuracy is almost never achievable on real images, because of the noise and the unknown characteristics of the acquisition process.

5.7.2 Out-of-space deformation

The true deformation is not guaranteed to lie in the space where we are looking for it and can therefore never be recovered exactly. This experiment compares the resulting approximation error with the overall registration error. We generated a random hierarchical deformation using the wavelet methodology from Chapter 4. It is a superposition of deformations belonging to B-spline wavelet spaces with knot spacings from $h = 8$ to $h = 256$ pixels. This way we generate a deformation with a controled smoothness which is not in the search space. At each level, the expected amplitude of the deformation is increased by two with respect to the preceding finer level. We deform the MRI image (the same as in the preceding experiment, see Figure 5.8) with this random deformation and try to recover the deformation in spaces with knot spacings $h = 8 \sim 256$. Figure 5.11 presents the original deformation represented, while Figure 5.12 shows the recovered deformations and the residual differences between the reference image and the warped test images for different values of the knot spacing h.

The resulting warping index is compared with the best achievable one in a given space in Figure 5.13. We see that although the ideal values are not attained, the difference is within the range of half a pixel, which is comparable to the essential registration error of $\varpi = 0.2$ for the case when the 'correct' space is used.

In real situations, the true deformation space is not known. However, thanks to

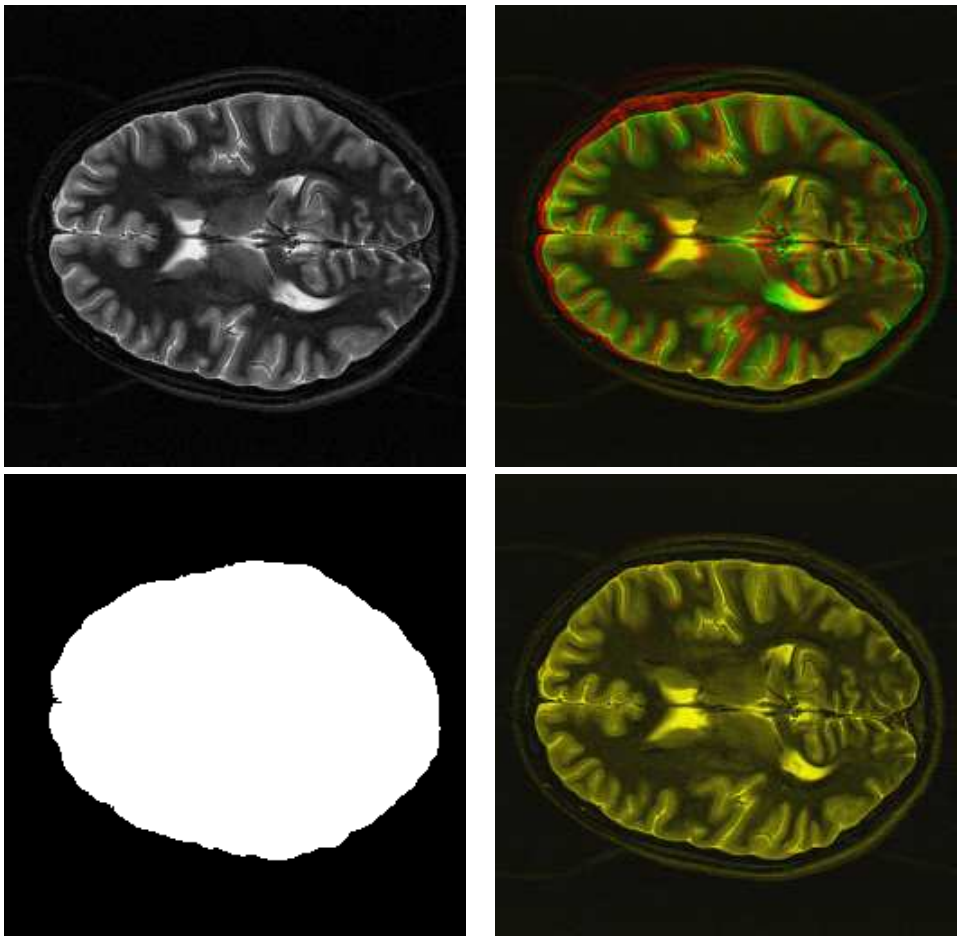


Figure 5.8: The original slice of anatomical MRI brain image *(top left)*, original superimposed over the true deformation *(top right)*, the recovered deformation versus the true deformation *(bottom right)*, and the mask used to calculate the warping index *(bottom left)*.

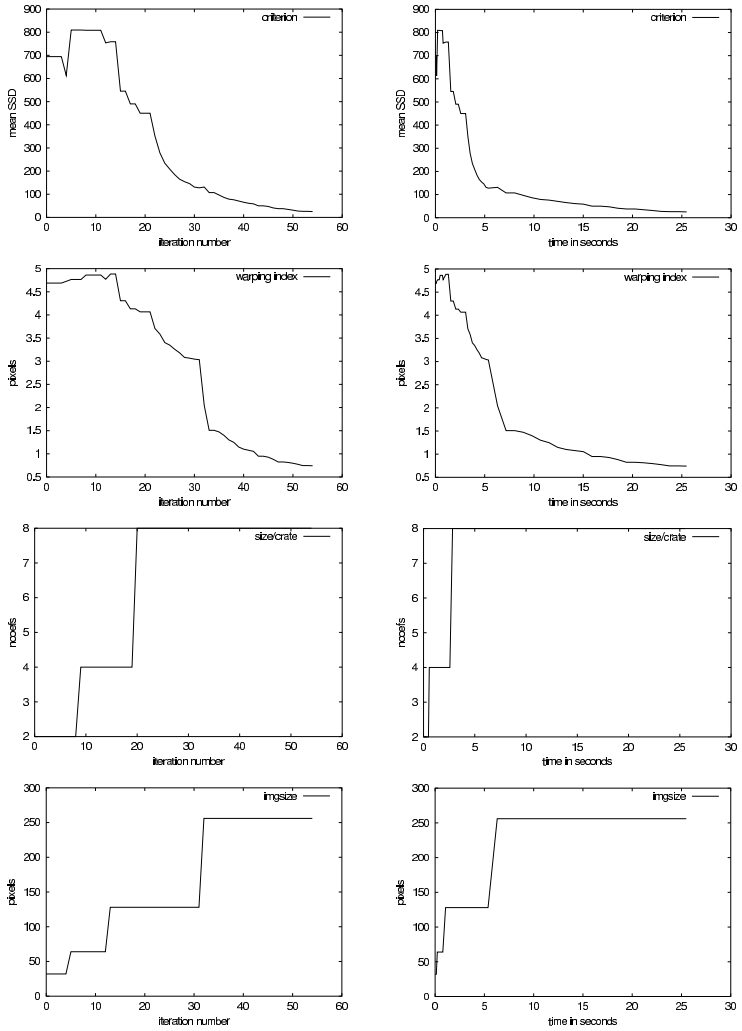


Figure 5.9: The evolution of the optimization process. The left column contains graphs with respect to the number of iterations, the same quantity is graphed with respect to time in the right column. The first row shows the SSD criterion E, the second row the warping index ϖ. The third and fourth row relate the multiresolution aspect of the optimization, showing the effective number of coefficients (in one dimension) and the image size, respectively.

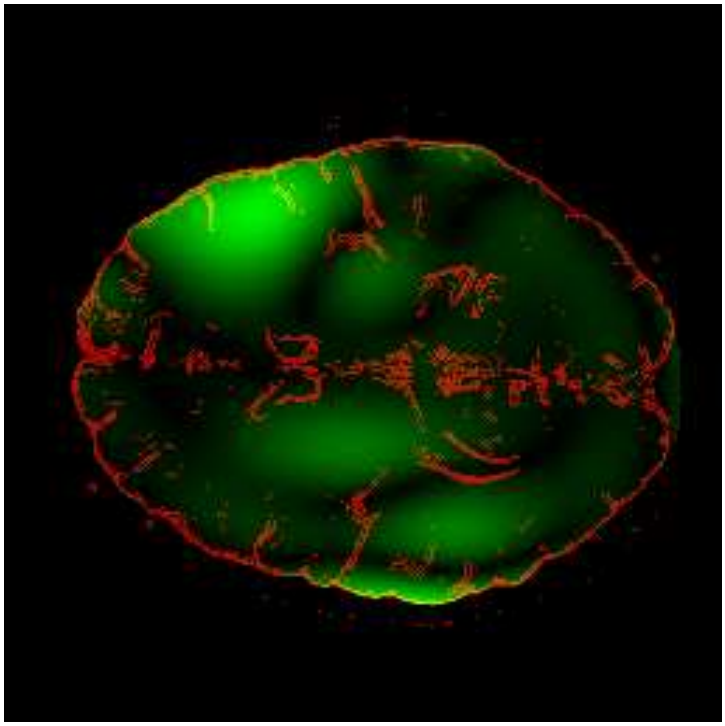


Figure 5.10: The geometrical error after registration (green) with superposed contours of the original MRI image (red). The maximum (green) intensity corresponds to the error of 1.5 pixels.

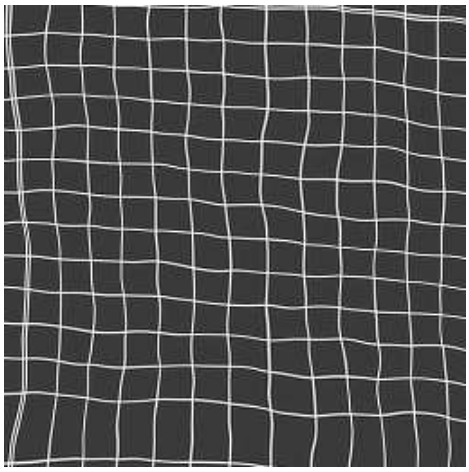


Figure 5.11: The original randomly generated deformation.

the good approximation properties of splines, we can reasonably expect that by using a sufficiently small value of h, we can reduce the approximation error to acceptable values. And the experiment we have just described indicates that the overall registration error should not be much bigger than the superposition of the approximation error and the essential registration error.

5.7.3 Choosing the spline degree

The choice of the spline degree for the image and deformation models is a trade-off between the accuracy and speed. Here too we generated hierarchical random deformations (see Section 5.7.2) and we recover them in a coarser space ($h = 64$). We observe that in this particular case the differences are rather small and it would be justifiable to use lower degree splines to gain speed. The differences would be more important for smoother deformations and/or images, which are better approximated by high order splines.

Note that, theoretically, the complexity of calculation should be approximately proportional to the support of the splines, and thus (in 2D) to the square of their order. However, the observed gain is much smaller. This is due to some fixed costs like pixel accesses, which are independent of the spline orders, and also due to the fact that the generally less precise low-order spline models usually require more iterations of the optimizer.

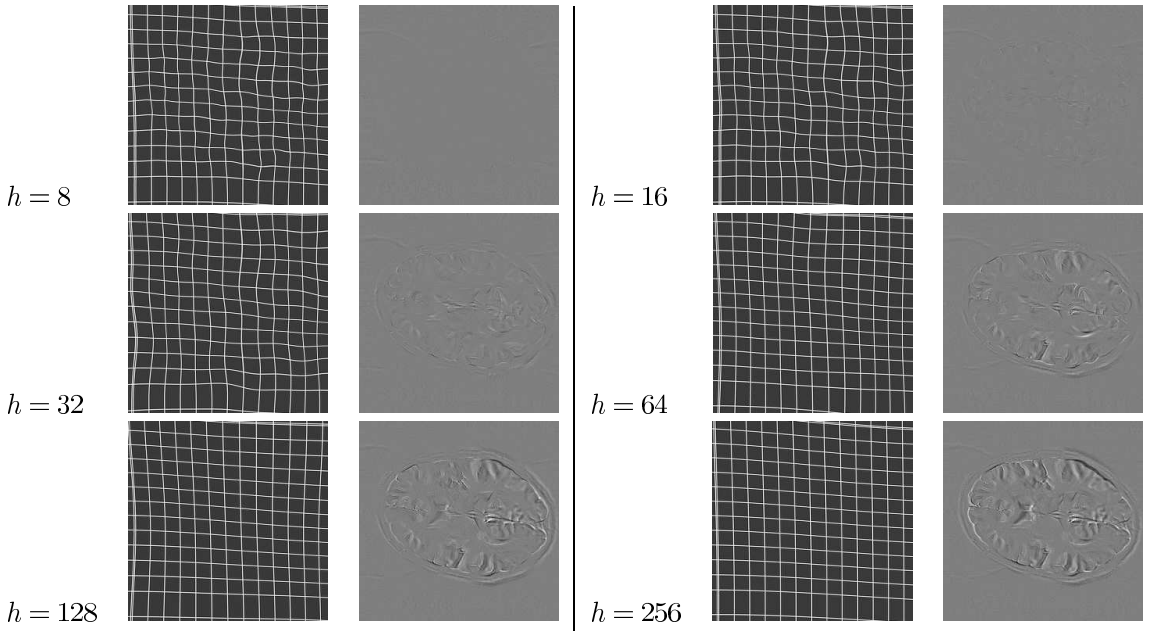


Figure 5.12: The deformations from Figure 5.11 recovered using progressively smaller (coarser) deformation spaces *(left column)*, and the corresponding residual error *(right column)*.

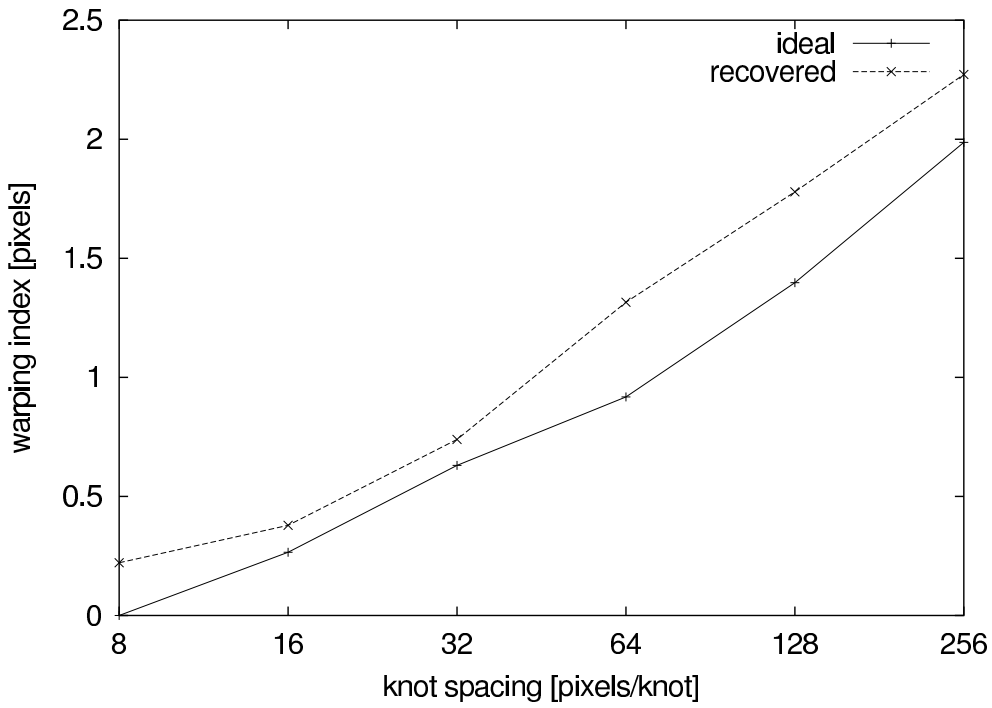


Figure 5.13: The ideal (best achievable) versus attained warping index when recovering the randomly generated hierarchical deformation applied on a MRI image, as a function of the warping space.

Table 5.2: The warping index (ϖ) and elapsed time as a function of the spline degrees in the image model (n_i) and the deformation model (n_m).

n_i	n_m	ϖ	time
3	1	3.067	20.1
3	2	2.880	26.7
3	3	2.837	48.9
1	3	3.020	23.4
2	3	2.838	43.2
3	3	2.837	48.9

5.7.4 Starting point

The following experiment evaluates the robustness of the algorithm with respect to the starting point. We try to recover the deformation from Section 5.7.1 (MRI images) optimizing only at the finest level. We will linearly vary the starting point of the optimizer between identity and the true deformation and observe the attained warping index ϖ for stopping threshold of 0.01 pixels. Figure 5.14 shows the warping index of the deformation used as a starting point and of the recovered deformation. We observe that although the final result does depend on the starting point, it is most likely only the influence of the stopping criterion. The algorithm therefore proves to be very robust, even without the help of a multiresolution: it converges in all cases. On the other hand, the elapsed time and the number of iterations differes significantly, from 2 iterations when starting from the true solution, to several hundreds when starting from identity.

5.7.5 Statistical distribution of errors

To evaluate the behavior of the algorithm on a larger set of cases, we have generated a series of random hierarchical deformations (see Section 5.7.2), warped the MRI slice with them, and applied our registration algorithm to recover the deformation. We use the stopping threshold of 0.01 pixels and a warping space which contains the deformation. We then compare the warping index corresponding to the recovered deformation with the initial warping index, that is, the distance between the true deformation and identity. In Figure 5.15 we present the histograms of the initial and final warping indexes and the scatter plot describing the relation between the two. We observe that the algorithm gives results with accuracy consistently better than 0.1 pixels.

5.8 Further applications

5.8.1 SPECT atlas

We applied our algorithm to the registration of ECD[3] and Xenon inhalation SPECT images [102].[4]

These image modalities are used to visualize the blood flow in the brain. The Xenon method is non-invasive, and the resulting images contain very little anatomical information. On the other hand, the ECD method requires intravenous injection, and the resulting images show also anatomical structures in the brain. Both methods yield a 3D volume of pixels obtained by a tomographic reconstruction procedure.

[3] TCD (Technetium Ethylene Cysteine Diethylester) is the radiactively marked intravenously injected agent.

[4] We described this application in [103]. We are very grateful to Prof. Slosman and Dr. Chicherio from the University Hospital in Geneva for bringing this problem to our attention and for providing us with experimental data.

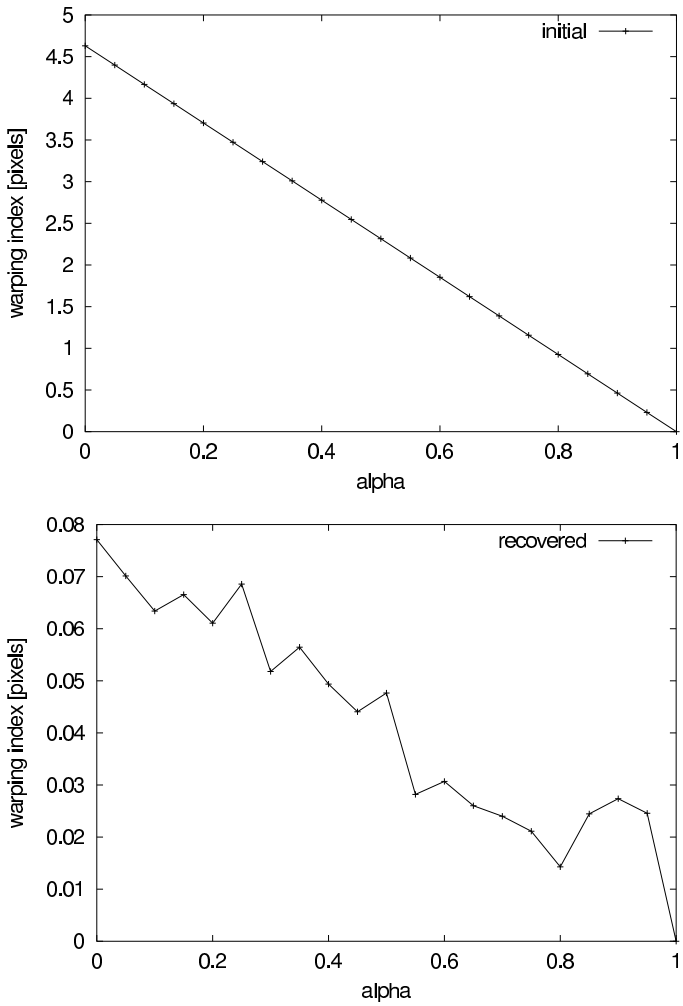


Figure 5.14: The warping index of the deformation used as a starting point *(top)* and of the recovered deformation *(bottom)*. The initial guess (starting point) varies from identity ($\alpha = 0$) to the true deformation ($\alpha = 1.0$).

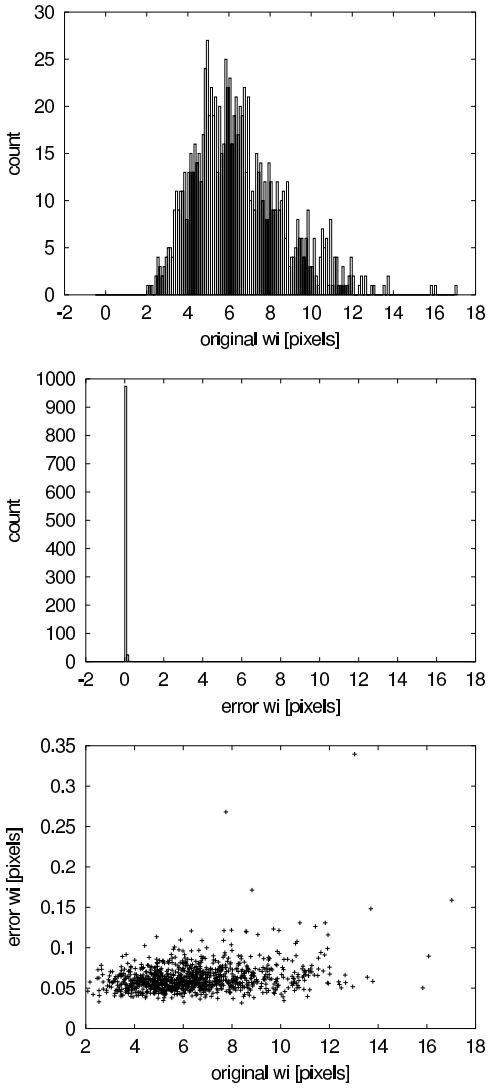


Figure 5.15: The histogram of the warping index of the true deformations *(top)*, the histogram of the geometric error (warping index) of the recovered deformation *(middle)*, and the scatter plot of the two *(bottom)*. The data is based on 142 experiments.

The position of the subjects' heads in the scanner differ as well as the head dimensions and the size of the internal structures of the brains. Therefore, in order to compare and evaluate the results of Xenon SPECT examination, the compared volumes (from different subjects) have to be registered.

We have chosen to first register the ECD SPECT images of the two subjects, and to apply the deformation found to the Xenon SPECT images acquired immediately after the ECD ones. The Xenon SPECT images cannot be registered directly because they contain too little anatomical information.

Once this method is perfected and applied to a large body of volunteers, an atlas of Xenon SPECT images will be created, permitting to use this non-invasive and fast method for diagnostical comparison of brain activities of a subject with an atlas.

As for the real application, the true correspondence between the two volumes is not known and it is therefore difficult to evaluate the performance of the registration algorithm, we have chosen to test it using artificially generated random deformation, using the methodology we described in Chapter 4.

Figure 5.16 gives an example of the SPECT images and of the difference before and after registration for artificially generated deformation. You can see that the SPECT images are rather blurred, which augments the difficulty of the registration task. Note also that the differences in the registered images are significantly reduced. Figure 5.17 shows the artificially generated deformation and the resulting deformation found by our algorithm.

We also registered real ECD SPECT images of two different subjects. Because the amplitudes in the two data sets were very different, we normalized them before performing the registration. Even better results were achieved using histogram equalization. The differences before and after registration for a selected slice are shown in Figure 5.18. We see that on the large scale the images were correctly registered, which is the first step towards their meaningful comparison.

5.8.2 Ultrasound for heart motion recovery

In this section we present an application of our algorithm to the registration of standard 2D ultrasound sequences of the heart.[5]

The registration algorithm has been used to register pairs of successive images from the sequence. The extracted deformation function effectively corresponds to the displacement field and thus also the velocity field of the points in the heart. Additional quantities can be derived from the velocity field, examples include the accumulated displacement, and strain. Figure 5.19 shows the accumulated displacement and the velocity field during the systole.

[5]The major part of this work was performed by María J. Ledesma-Carbayo during Autumn 2000 and Winter 2001, while she stayed with our group. More details can be found in our conference paper [104].

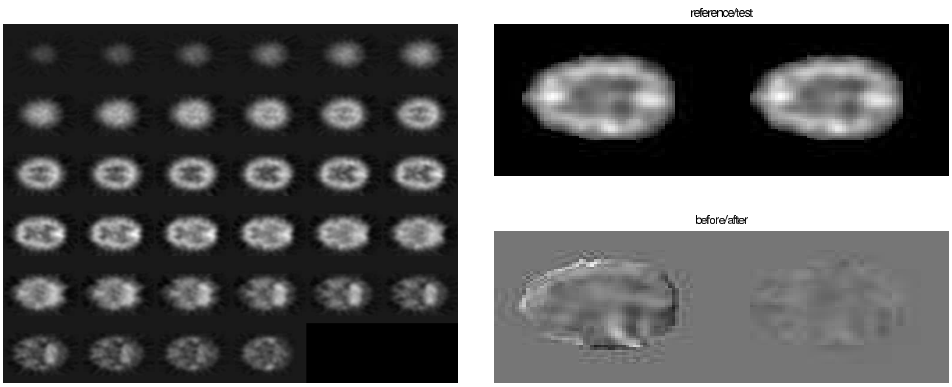


Figure 5.16: The slices of an ECD SPECT image are shown on the left. One slice of the difference between and after registration of EP reconstructed Xenon SPECT images for artificially generated deformation is shown on the bottom right. The top right image shows the corresponding slices from the two EP images before registration.

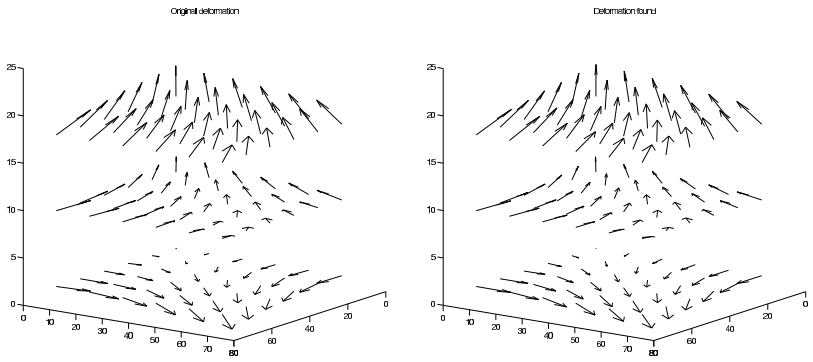


Figure 5.17: The left image shows a randomly generated deformation which was applied to an ECD SPECT image in the previous figure. The right image shows the deformation recovered using our algorithm by registering the deformed SPECT image with the original one. We observe that for this model case, the deformation is very well recovered.

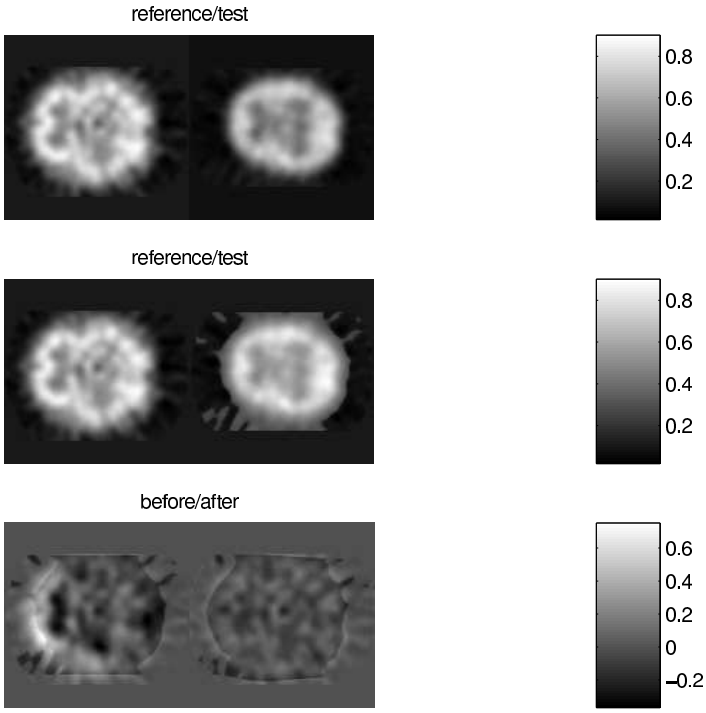


Figure 5.18: A slice of the ECD SPECT volume for two different subjects *(top)*, with normalized intensities. The same slices after histogram equalization *(middle)*. The difference between them before *(bottom left)* and after *(bottom right)* registration.

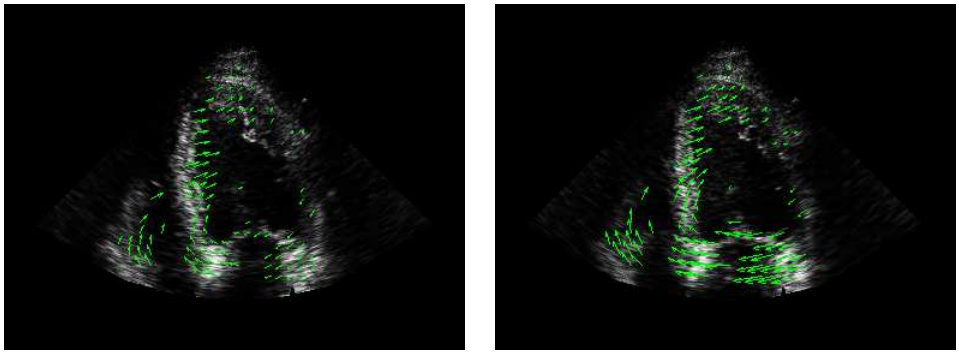


Figure 5.19: *Left:* Accumulated displacement field at the time of maximum contraction shown with arrows overlaid on the first image of the sequence. *Right:* Velocity field during the contraction.

The automatically extracted motion and derived parameters are essential for visualizing, analyzing, and quantifying heart motion and consequently heart function, with direct applications in diagnostic. So far, similar analysis has been only possible using the tagged MRI technique which is much more instrument- and patient-intensive.

5.9 Conclusions

We developed a fully automatic elastic registration algorithm. We extended the idea from Chapter 4 to multidimensional data, and streamlined the algorithm to accelerate it. We designed a new step-prediction formula for gradient descent algorithm and showed its efficiency for our application.

We introduced a concept of virtual springs, yielding a semi-automatic registration method, capable of using expert hints in the form of landmarks to solve particularly difficult problems where the fully automatic algorithm may be mislead. This is a powerful combination of the ideas of manual landmark registration of Chapter 3 and the pixel-based registration using splines from Chapter 4.

We applied the algorithm to an artificially generated problem with anatomical MRI images to demonstrate its speed and accuracy. Furthermore, we presented two medical applications in two different medical fields: SPECT imaging and ultrasound cardiology.

Our implementation still includes a lot of generality to ease experiments with different configurations of the registration problem and the optimization algorithm. We believe that by producing a specialized program taking advantage of a specific configuration, the run time can be decreased by an additional factor of 2 to 4. This will enable truly

interactive implementation of automatic and semi-automatic elastic image registration with numerous applications in medicine, biology, and any other field where deformed images need to be compared.

Chapter 6

Variational Reconstruction: Tutorial

This chapter is based on our article [105].

6.1 Abstract

We consider the problem of reconstructing a multidimensional vector function $\mathbf{f}_{\text{in}} : \mathbb{R}^m \to \mathbb{R}^n$ from irregularly sampled responses of several linear shift-invariant filters. Traditional approaches reconstruct in an *a priori* given space; e.g., the space of bandlimited functions. Instead, we have chosen to search for a reconstruction which is optimal in the sense of a quadratic plausibility criterion J. We motivate this choice and show that the variational formulation is applicable to a wide variety of problems.

First, we present the solution of the generalized interpolation problem, that is, when the reconstructed function is exactly consistent with the measurements. Later, we consider also the approximation problem, which is the continuously-regularized version of the interpolation problem, not requiring an exact consistence with the measurements. We show that both problems lead to the same class of solutions. Interestingly, thanks to the specific form of the solution, the system of equations for the approximation problem is obtained by straightforward addition of a constant diagonal matrix to the system for the interpolation case.

We present a brief review of existing reconstruction techniques, formulate a general framework of the variational approach, show the reconstruction formula and apply it to several practical examples, including new variational formulation of derivative sampling, landmark warping, and tomographic reconstruction.

6.2 Introduction

6.2.1 Perfect reconstruction

Reconstructing a signal from its samples is one of the most fundamental tasks in signal processing. The classical sampling theorem presented by Shannon [106] states that a bandlimited function f_{in} (whose frequency spectrum is limited by the Nyquist frequency $\omega_{\max} = \pi/T$) can be perfectly reconstructed from its regularly-spaced (ideal) samples $s_j = f_{\text{in}}(jT)$. Reconstruction is carried out by convolving the samples with a sinc kernel:

$$f_{\text{out}}(x) = f_{\text{in}}(x) = \sum_{j \in \mathbb{Z}} s_j \operatorname{sinc}(x/T - j)$$

$$\text{where} \quad \operatorname{sinc}(x) = \frac{\sin(\pi x)}{\pi x} \tag{6.1}$$

In 1977, Papoulis [107] showed that it was also possible to recover f_{in} from the output of q linear shift-invariant filters sampled at $(1/q)$-th the Nyquist rate. This has generalized Shannon's theory in two important ways. First, it allows for *non-ideal* sampling, closer to the reality than Shannon's ideal one. Second, it provides for *multichannel* sampling, permitting the reconstruction using several measurement devices. Such a multichannel, generalized sampling system is shown in Figure 6.1.

Papoulis theory has been further extended to a multidimensional [108] ($m > 1$) and vector [109] ($n > 1$) bandlimited functions $\mathbf{f}_{\text{in}} : \mathbb{R}^m \to \mathbb{R}^n$. Interlaced reconstruction (working at different scales) is also possible in some wavelet spaces [110].

Recent applications of generalized sampling include, among others, deinterlacing [111, 112], and super-resolution [113, 114] reconstruction.

6.2.2 Consistent reconstruction

Unser and Aldroubi [115] generalized this framework by replacing the principle of perfect reconstruction by a weaker condition of *consistency*, requiring that the reconstructed signal f_{out} provides exactly the same measurements as the original signal f_{in}, when run through the measurement system.

The reconstruction space V in which they search for the function f_{out} can be different from the input space, provided that the reconstruction is *unique*. That is, there is only one function $f \in V$ consistent with the measurements. Note that the uniqueness depends on the reconstruction space V, as well as on the measurement system and sampling points.

Unser and Aldroubi chose the assumption of square-integrable input functions, from the space L_2. They searched the reconstructed function f_{out} in a functional space $F =$

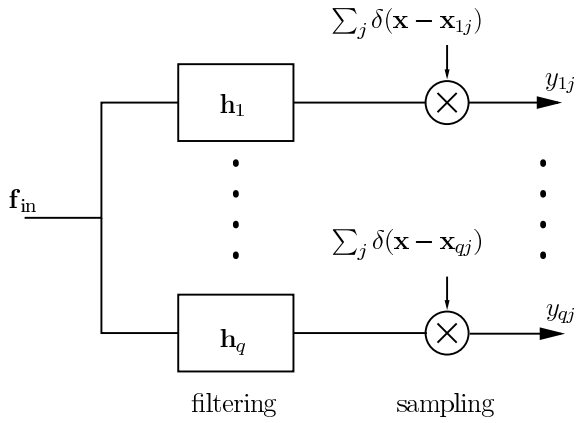


Figure 6.1: Generalized sampling. Sampling is modeled by a multiplication with a multidimensional train of Dirac's masses. In Papoulis's framework $x_{ij} = x_j = iqT$. Here, the sampling locations $\mathbf{x}_{ij}$ can be arbitrary.

$V(\varphi)$, generated by integer translates of a function φ

$$V(\varphi) = \left\{ f(x) = \sum_{k \in \mathbb{Z}} c_k \varphi(x - k);\ c_k \in \ell_2 \right\} \tag{6.2}$$

Their reconstruction formula is a generalized case of (6.1). The approach was extended to the multichannel case in [116], providing a generalized version of the Papoulis theory.

6.2.3 Non-uniform sampling

In many applications, the location of measurement points is irregular. This is often the case when human input is involved, or when some part of the sampled domain needs more attention. Typical examples include shape reconstruction from incomplete measurements [117] or interpolating an image deformation from landmarks (important points in the image), whose location is specified manually or automatically detected [14, 38, 42, 86].

The reconstruction can be done within the class of bandlimited functions [118, 119] or more general wavelet and spline-like spaces [120].

For an extensive review on sampling, see [121, 122].

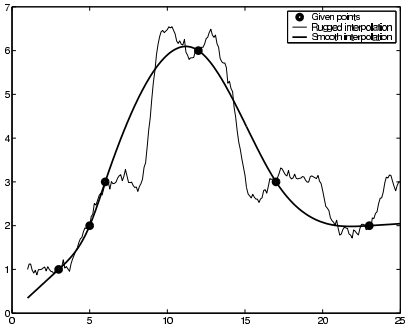


Figure 6.2: When interpolating a function from its values (circles), many solutions are possible. However, smooth interpolation (solid line) is usually preferable to a rugged one (dashed line).

6.2.4 Motivation for variational reconstruction

As an illustrative example, let us consider the task of interpolating a unidimensional function given its values at some sampling points. As illustrated in Figure 6.2, there is an infinite number of functions passing through the given points. Nevertheless, most people would probably agree that the smooth approximation curve in Figure 6.2 looks 'more correct' than the rugged noisy approximation. Similarly, we can often quantify the degree of plausibility of a function for a given application. Then, we search for the most plausible function satisfying our interpolation (consistency) conditions. This is the key concept of our approach. From now on, we will concentrate on the typical case where we want the solution to be 'smooth'. As smoothness can be measured by the amplitude of the derivatives, maximizing smoothness translates into minimizing the norm of various differential operators. In Figure 6.2, the smooth curve minimizes the L_2-norm of the second derivative $\|f''\|_{L_2}$, which is known to yield a cubic spline interpolation [90, 91].

6.2.5 Proposed variational reconstruction

The reconstruction method presented in this paper has been designed to be as general as possible from several possible viewpoints. It can handle multidimensional and vector functions. It is applicable for multichannel sampling using filters (non-ideal sampling). We put no bandlimiting restrictions on the input signal. The reconstruction is stable and unique for a large class of sampling configurations.

We retain the idea of a consistent reconstruction. However, we will not explicitely specify the reconstruction space beforehand. Rather, this space will be determined naturally from the problem at hand based on continuous *regularization*. We introduce a non-

negative *smoothness criterion* (penalty function) $J(\mathbf{f})$, which gets smaller as the function $\mathbf{f}$ gets smoother. We then seek a function $\mathbf{f}$ minimizing this criterion under the consistency constraints (introduced in Section 6.2.2); e.g., passing through given points. In other words, we replace the subspace constraint $\mathbf{f} \in V(\varphi)$ by a variational formulation. The criterion $J(\mathbf{f})$ provides the regularization needed to overcome the ambiguity of the reconstruction problem. It may also represent an *a priori* knowledge in the Bayesian framework, quantifying our confidence that a particular function $\mathbf{f}$ is close to the input $\mathbf{f}_{\text{in}}$ [76].

Another advantage of using a variational formulation is that it gives us a unique solution, except for rare pathological cases. On the other hand, for the subspace methods, the uniqueness is very difficult to ensure, especially if the distribution of the measurement points is not known *a priori*.

6.2.6 Existing work

The work presented in this paper can be seen as an extension of the theory of radial-basis function approximation [72], especially Duchon's thin-plate splines [45, 71]. Our formulation generalizes the method for vector functions, non-ideal (generalized) sampling, and generating functions that need not be radial. Moreover, we present an accessible and self-contained description of the approximation techniques yielded by this theory targeted to the engineering audience.

It is also worth noting that our regularization is completely specified in the continuous domain, unlike alternative methods that often use discretized version of the regularization operator.

Some aspects of the different reconstruction methods for sampling are presented in Table 6.1. Note that even if a method does not explicitely support vector functions, it can be applied to such functions by considering each component separately. Similarly, uni-dimensional linear reconstruction methods can be applied to multidimensional functions by considering the tensor product of the basis functions. Furthermore, a reconstruction problem with non-uniform sampling locations can be recast as a problem with generalized sampling operators acting on uniform sampling [120]. Finally, many of the cells marked 'no' could have been filled by adapting the techniques developed in other contexts (such as wavelets). In particular, one may consider non-separable wavelets [123], vector-valued wavelets [124], or box-splines [125]; however, we are not aware that these extensions had been explicitely applied to sampling.

6.2.7 Organization of the paper

Deriving the solution of the variational reconstruction problem that we briefly introduced in Section 6.2.5 is not overly difficult but requires some mathematical precautions and rigor. Accordingly, to improve the readability of this Chapter, we have chosen to present here mostly the results, in a directly applicable form. The general description of the

Table 6.1: Various reconstruction methods and some of their generic features. See discussion in Section 6.2.6

	Shannon [106]	Papoulis [107]	Wavelets, splines	RBF [72]	This paper
Sampling type	ideal	generalized	generalized	ideal	generalized
Sampling locations	uniform	uniform	uniform	non-uniform	non-uniform
Reconstruction space	bandlimited	bandlimited	splines/wavelets	implicit	implicit
Variational	no	no	no	yes	yes
Multichannel	no	yes	[110, 116]	no	yes
Vector valued	no	[109]	no	no	yes
Multidimensional	no	[108]	no	yes	yes
Input space	bandlimited	bandlimited	L_2 [115], wavelets [126]	arbitrary	arbitrary

solution is given in Section 6.3 and several examples of its applications are treated in Section 6.4.

The more general form of the theory, the tools required, and the mathematical derivation of the solution *per se* are presented in Chapter 7.

6.3 General description of the solution

6.3.1 Problem description

Let $\mathbf{f}_{\text{in}} : \mathbb{R}^m \to \mathbb{R}^n$ be the unknown multidimensional vector function (with n components in m dimensional space) which we want to reconstruct. We also have a filterbank $\mathsf{H}_{n\times q} = [\mathbf{h}_1 \dots \mathbf{h}_q]$, consisting of q filters $\mathbf{h}_i$ of size $n \times 1$, which gives us a set of filtered signals $\mathbf{h}_i^T * \mathbf{f}_{\text{in}}$. We measure (sample) each of the filtered signals at N arbitrary points $\mathbf{x}_{ij}$ which gives a set of qN real samples

$$s_{ij} = \mathbf{h}_i^T * \mathbf{f}_{\text{in}}(\mathbf{x}_{ij}) = \int_{\mathbb{R}^m} \mathbf{h}_i^T(\mathbf{x})\, \mathbf{f}_{\text{in}}(\mathbf{x}_{ij} - \mathbf{x}) \mathrm{d}\mathbf{x} \tag{6.3}$$

for $i \in \{1, \dots, q\}$ and $j \in \{1, \dots, N\}$. This corresponds to the situation depicted in Figure 6.1. Note that the scheme would also admit a different number of samples N_i for each channel. However, for the reasons of notational simplicity, we only treat the case $N_i = N$ here.

Recall the smoothness criterion $J(\mathbf{f})$ we introduced in Section 6.2.5. We can now specify it more concretely as the squared norm of a filtered version of the function $\mathbf{f}$ with a filterbank $\mathsf{L}_{n\times p} = [\mathbf{l}_1 \dots \mathbf{l}_p]$ consisting of p filters $\mathbf{l}_k$, each represented by a vector of distributions of size $n \times 1$.

$$J(\mathbf{f}) = \|\mathsf{L}^T * \mathbf{f}\|_{L_2}^2 = \sum_{k=1}^{p} \|\mathbf{l}_k^T * \mathbf{f}\|_{L_2}^2 \tag{6.4}$$

This permits us to define the reconstruction space F as a space of functions for which both the criterion $J(\mathbf{f})$ and the outputs of the measurement filters $(\mathbf{h}_i^T * \mathbf{f})$ are finite:

$$F = \{\mathbf{f} : \mathbb{R}^m \to \mathbb{R}^n;\ J(\mathbf{f}) < \infty \quad \text{and} \quad \forall\, i \in \{1, \dots, q\}, j \in \{1, \dots, N\};\ \left|\mathbf{h}_i^T * \mathbf{f}(\mathbf{x}_{ij})\right| < \infty\Big\} \tag{6.5}$$

We want to find a reconstruction operator to give us a function $\mathbf{f}_{\text{out}}$ from the reconstruction space F, given the samples s_{ij} in (6.3).

The *consistency condition* requires that the reconstructed signal $\mathbf{f}_{\text{out}}$ provides exactly the same measurements as the original signal $\mathbf{f}_{\text{in}}$ when run through our measurement

system:

$$\big(\mathbf{h}_i^T * \mathbf{f}_{\text{in}}\big)(\mathbf{x}_{ij}) = s_{ij} = \big(\mathbf{h}_i^T * \mathbf{f}_{\text{out}}\big)(\mathbf{x}_{ij}) \qquad \forall\, i \in \{1,\dots,q\}, j \in \{1,\dots,N\} \quad (6.6)$$

Let us call F_c the set of all functions of F consistent with the measurements

$$F_c = \Big\{\mathbf{f} \in F; \forall\, i,j;\ s_{ij} = \mathbf{h}_i^T * \mathbf{f}(\mathbf{x}_{ij})\Big\} \quad (6.7)$$

Finally, we can define our interpolation problem. We say that $\mathbf{f}_{\text{out}} \in F_c$ is a solution to a *generalized interpolation problem* GIP($\mathsf{L},\mathsf{H},\mathsf{X},\mathbf{s}$) iff it minimizes J, that is iff for all functions $\mathbf{f} \in F_c$ we have $J(\mathbf{f}) \geq J(\mathbf{f}_{\text{out}})$. We will write $\mathbf{f}_{\text{out}} \in \text{GIP}(\mathsf{L},\mathsf{H},\mathsf{X},\mathbf{s})$,where $\mathsf{X}_{m\times qN} = [\mathbf{x}_{1,1}\,\mathbf{x}_{1,2}\dots\mathbf{x}_{q,N}]$ is a matrix of all sampling locations, and $\mathbf{s} = [s_{1,1}\,s_{1,2}\dots s_{q,N}]^T$ is a vector of size $qN \times 1$ containing all measurements .

6.3.2 Choice of the criterion

The choice of the filterbank L defining the criterion (6.4) is to a large extent restricted by the required properties of the solution. Already by construction, the solution is translation-invariant with respect to measurement locations $\mathbf{x}_{ij}$ and linear with respect to measurements s_{ij}.

On top of it, we want the solution to be *scale-invariant*. For example, we want the solution to be independent on whether we measure in meters or inches. This can be assured if the criterion J is scale pseudo-invariant. We say that $J(\mathbf{f})$ is pseudo-invariant with respect to a geometrical transformation $\mathbf{g}_\alpha$, where α is a (vector) parameter, iff there is a function $\xi(\alpha)$ such that[1] $J(\mathbf{f}\circ\mathbf{g}_\alpha) = \xi(\alpha)J(\mathbf{f})$. In one dimension, derivative criteria such as

$$J(f) = \left\| \frac{\partial^M f}{\partial x^M} \right\|^2 \quad (6.8)$$

satisfy the scale pseudo-invariance. Note that a linear combination of derivatives of different orders does not satisfy the pseudo-invariance. On the other hand, fractional derivatives are acceptable.

For multidimensional functions ($m > 1$), we additionally impose the *rotation-invariance* of the criterion. The semi-norms introduced by Duchon [45, 71]

$$J(f) = \|f\|^2_{D_M} = \sum_{|\mathbf{k}|=M} \frac{M!}{\mathbf{k}!} \int_{\mathbb{R}^m} \left(\frac{\partial^M f}{\partial \mathbf{x}^{\mathbf{k}}} \right)^2 \mathrm{d}\mathbf{x} \quad (6.9)$$

[1] We define $(\mathbf{f}\circ\mathbf{g}_\alpha)(\mathbf{x}) = \mathbf{f}\big(\mathbf{g}_\alpha(\mathbf{x})\big)$

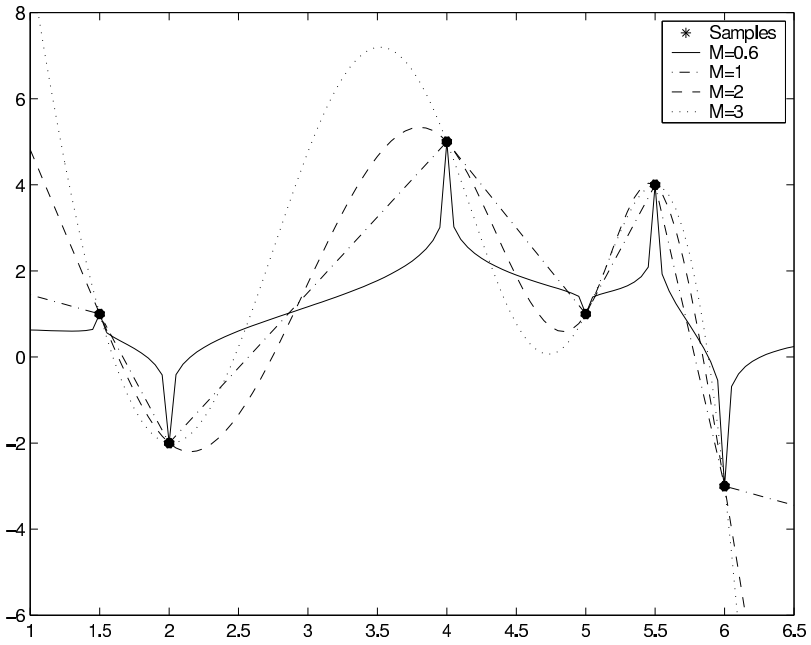


Figure 6.3: The dependence of the interpolation results on the order of the semi-norm M used. High order semi-norms tend to produce slowly varying curves with large overshoots and vice-versa.

where $k_l \in \{0, \ldots, M\}$, $\mathbf{k}! = \prod k_l!$, $|\mathbf{k}| = \sum_{l=1}^{M} k_l$, and $\partial \mathbf{x}^{\mathbf{k}} = \partial x_1^{k_1} \ldots \partial x_M^{k_M}$, satisfy the rotation-invariance. This can be seen from the Fourier form of the semi-norm

$$\|f\|_{D_M}^2 = (2\pi)^m \int_{\mathbb{R}^m} \|\boldsymbol{\omega}\|^{2M} \left\| \hat{f}(\boldsymbol{\omega}) \right\|^2 \mathrm{d}\boldsymbol{\omega} \tag{6.10}$$

which can be used for well-behaved functions, see Chapter 7 for more details.

Note, that (6.9) defines a semi-norm, not a norm. Consequently, there is a set of functions, called *kernel*, for which the value of the semi-norm is zero. For Duchon's semi-norms, the kernel consists of polynomials of degree $M - 1$.

The most often used Duchon's semi-norms are summarized in Table 6.2. Table 6.3 gives kernels for the same semi-norms.

The choice of the order of the semi-norm influences the smoothness of the interpolation, as shown in Figure 6.3.

For vector functions ($n > 1$), we choose to add a requirement of rotational invariance in the output (measurement) domain. Consequently, the criterion $J(\mathbf{f})$ treats all

Table 6.2: Most often used Duchons' semi-norms in dimensions $m = 1, 2, 3$. M is the order of the semi-norm. For other combinations of m and M see equation (6.9)

$\mathbb{R}^m$	D_M	$\|f\|^2_{D_M}$ for $f : \mathbb{R}^m \to \mathbb{R}$
$\mathbb{R}^1$	D_2	$\int \left(\frac{\partial^2 f}{\partial x^2}\right)^2 \mathrm{d}x$
$\mathbb{R}^1$	D_3	$\int \left(\frac{\partial^3 f}{\partial x^3}\right)^2 \mathrm{d}x$
$\mathbb{R}^2$	D_2	$\int \left(\frac{\partial^2 f}{\partial x^2}\right)^2 + 2\left(\frac{\partial^2 f}{\partial x \partial y}\right)^2 + \left(\frac{\partial^2 f}{\partial y^2}\right)^2 \mathrm{d}x\mathrm{d}y$
$\mathbb{R}^2$	D_3	$\int \left(\frac{\partial^3 f}{\partial x^3}\right)^2 + 3\left(\frac{\partial^3 f}{\partial x^2 \partial y}\right)^2 + 3\left(\frac{\partial^3 f}{\partial x \partial y^2}\right)^2 + \left(\frac{\partial^3 f}{\partial y^3}\right)^2 \mathrm{d}x\mathrm{d}y$
$\mathbb{R}^3$	D_2	$\int \left(\frac{\partial^2 f}{\partial x_1^2}\right)^2 + \left(\frac{\partial^2 f}{\partial x_2^2}\right)^2 + \left(\frac{\partial^2 f}{\partial x_3^2}\right)^2 + 2\left(\frac{\partial^2 f}{\partial x_1 \partial x_2}\right)^2 + 2\left(\frac{\partial^2 f}{\partial x_1 \partial x_3}\right)^2 + 2\left(\frac{\partial^2 f}{\partial x_2 \partial x_3}\right)^2 \mathrm{d}x\mathrm{d}y\mathrm{d}z$

Table 6.3: Kernels of most often used Duchons' semi-norms in dimensions $m = 1, 2, 3$. M is the order of the semi-norm.

$\mathbb{R}^m$	D_M	kernel of $\|f\|^2_{D_M}$ for $f : \mathbb{R}^m \to \mathbb{R}$
$\mathbb{R}^1$	D_2	$a_0 + a_1 x$
$\mathbb{R}^1$	D_3	$a_0 + a_1 x + a_2 x^2$
$\mathbb{R}^2$	D_2	$a_0 + a_1 x + a_2 y$
$\mathbb{R}^2$	D_3	$a_0 + a_1 x + a_2 y + a_3 x^2 + a_4 y^2 + a_5 xy$
$\mathbb{R}^3$	D_2	$a_0 + a_1 x + a_2 y + a_3 z$

components of $\mathbf{f}$ equivalently. This yields criteria of the form

$$J(\mathbf{f}) = \sum_k \|f_k\|^2 \tag{6.11}$$

where each component f_k is measured using Duchon's semi-norm (6.9).

6.3.3 Fundamental solution

As we will see in the following section, the criterion J and the sampling filterbank H determine the space containing the solution of the generalized interpolation problem. The generating (basis) functions $\boldsymbol{\varphi} : \mathbb{R}^m \to \mathbb{R}^n$ of the space are called *fundamental solutions* and they satisfy the following distributional relation:

$$\mathsf{L} * \mathsf{L}^R * \underbrace{[\boldsymbol{\varphi}_1 \dots \boldsymbol{\varphi}_q]}_{\boldsymbol{\Phi}_{n\times q}} = \mathsf{H} \tag{6.12}$$

where $\mathsf{L}^R(\mathbf{x}) = \mathsf{L}^T(-\mathbf{x})$. The easiest way to find the fundamental solutions is to first identify the set of *Green functions* [127] $\boldsymbol{\psi}$ corresponding to the filterbank L

$$\mathsf{L} * \mathsf{L}^R * \underbrace{[\boldsymbol{\psi}_1 \dots \boldsymbol{\psi}_n]}_{\boldsymbol{\Psi}_{n\times n}} = \delta(\mathbf{x})\, \mathsf{I}_{n\times n} \tag{6.13}$$

where I is the identity matrix. Table 6.4 gives the scalar Green functions for the most often used Duchon's semi-norms as well as in the general case. Note that it is enough to consider the scalar case ($n = 1$), when the convolution $\mathsf{L} * \mathsf{L}^R = u$ is a scalar. In the vector case ($n > 1$), we get $\boldsymbol{\Psi} = \psi \mathbf{I}_{n\times n}$ thanks to (6.11).

Having determined the Green functions, the fundamental solutions are obtained by convolving Green functions with the sampling operators:

$$[\boldsymbol{\varphi}_1 \dots \boldsymbol{\varphi}_q] = \boldsymbol{\Phi}_{n\times q} = \boldsymbol{\Psi}_{n\times n} * \mathsf{H}_{n\times q} \tag{6.14}$$

$\mathbb{R}^m$	D_M	$c\,\psi(\mathbf{x})$	$\hat{c}\,\hat{\psi}(\boldsymbol{\omega})$	remark
$\mathbb{R}^1$	D_2	r^3	$\|\omega\|^{-4}$	
$\mathbb{R}^1$	D_3	r^5	$\|\omega\|^{-4}$	
$\mathbb{R}^2$	D_2	$r^2 \log r$	$\|\|\boldsymbol{\omega}\|\|^{-4}$	
$\mathbb{R}^2$	D_3	$r^4 \log r$	$\|\|\boldsymbol{\omega}\|\|^{-6}$	
$\mathbb{R}^3$	D_2	r	$\|\|\boldsymbol{\omega}\|\|^{-4}$	
$\mathbb{R}^m$	D_α	$r^{2\alpha-m} \log r$	$\|\|\boldsymbol{\omega}\|\|^{-2\alpha}$	if $2\alpha - m$ is even
$\mathbb{R}^m$	D_α	$r^{2\alpha-m}$	$\|\|\boldsymbol{\omega}\|\|^{-2\alpha}$	otherwise

Table 6.4: Green functions ψ in dimension m, satisfying $u * \psi = \delta$, where $u = \mathbf{l}^T * \mathbf{l}$ corresponds to Duchon's semi-norm $\|f\|_{D_M} = \|\mathbf{l}^T * f\|$. The multiplicative constants c, $\hat{c}$ can be determined but are irrelevant for our purposes. The functions are all expressed using Euclidean distance $r = \|\mathbf{x}\|$. The last formula is valid also for non-integer α.

6.3.4 Explicit solution

The solution to our generalized interpolation problem in its most useful form is given by the following theorem, to be proven in Chapter 7.

Theorem 1 (GIP solution) *The generalized interpolation problem GIP*$(\mathsf{L}, \mathsf{H}, \mathsf{X}, \mathbf{s})$*, is solved by a function*

$$\mathbf{f}_{out}(\mathbf{x}) = \underbrace{\sum_{k=0}^{P-1} a_k \mathbf{p}_k}_{\text{kernel part}} + \underbrace{\sum_{i=1}^{q} \sum_{j=1}^{N} \lambda_{ij} \boldsymbol{\varphi}_i(\mathbf{x} - \mathbf{x}_{ij})}_{\text{fundamental part}} \tag{6.15}$$

where $\boldsymbol{\varphi}_i$ are the fundamental solutions, $\{\mathbf{p}_k\}_{k=0}^{P-1}$ is the basis of the kernel of the semi-norm J (e.g., polynomials), and the real coefficients a_k and λ_{ij} are determined in such a way that $\mathbf{f}_{out} \in F$, if and only if the following two conditions are satisfied:

(i) The solution $\mathbf{f}_{out}$ is consistent with the constraints:

$$y_{ij} = \mathbf{h}_i^T * \mathbf{f}_{out}(\mathbf{x}_{ij}) \quad \textit{for all} \quad i, j \tag{6.16}$$

(ii) The coefficients λ_{ij} satisfy the 'orthogonality' condition

$$\sum_{ij} \lambda_{ij} (\mathbf{h}_i^T * \mathbf{p}_k)(\mathbf{x}_{ij}) = 0 \quad \textit{for all} \quad k \tag{6.17}$$

The orthogonality condition (ii) ensures that the fundamental part of the solution (6.15) is orthogonal to any element of the kernel. This way we represent as much as possible of the solution $\mathbf{f}_{\text{out}}$ by the kernel term, that is, for zero cost.

Equivalently, conditions (i) and (ii) can be written in a matrix form as

$$\underbrace{\begin{bmatrix} \mathsf{A} & \mathsf{Q} \\ \mathsf{Q}^T & 0 \end{bmatrix}}_{\mathsf{B}} \begin{bmatrix} \boldsymbol{\lambda} \\ \mathbf{a} \end{bmatrix} = \begin{bmatrix} \mathbf{s} \\ \mathbf{0} \end{bmatrix} \tag{6.18}$$

where the parameters have been arranged in vectors as $\boldsymbol{\lambda} = [\lambda_{1,1} \dots \lambda_{1,N}\ \lambda_{2,1} \dots \lambda_{q,N}]^T$ and $\mathbf{a} = [a_1 \dots a_P]^T$. The components of the matrix A of size $Nq \times Nq$ are given by $\left(\mathsf{A}\right)_{iN+k,jN+l} = \mathbf{h}_i^T * \boldsymbol{\varphi}_j(\mathbf{x}_{ik} - \mathbf{x}_{jl})$ and represent the contribution of the fundamental solutions to each measurement. The components of the matrix Q are given by $\left(\mathsf{Q}\right)_{iN+j,k} = \mathbf{h}_i^T * \mathbf{p}_k(\mathbf{x}_{ij})$ and represent the kernel part of the solution as well as the orthogonality conditions.

We see that the solution (6.15) consists of two parts. The first, kernel part, does not contribute to the criterion, $J(\mathbf{f} + \mathbf{p}_k) = J(\mathbf{f})$, so we can intuitively tell that it is useful to accommodate in it as much as possible of $\mathbf{f}_{\text{out}}$. The second, fundamental part consists of a linear combination of shifted basis functions $\boldsymbol{\varphi}_j$ positioned at the sampling points. Interestingly, the fundamental part is reminiscent of a wavelet (or multi-wavelet) like expansion for it also involves shifts of some generating functions. One difference is that here the basis functions $\boldsymbol{\varphi}_i(\mathbf{x} - \mathbf{x}_{ij})$ in (6.15) are not necessarily uniformly spaced. Another difference is that wavelets are usually well localized while the functions $\boldsymbol{\varphi}_i$ (related by convolution (6.14) to the Green functions ψ from Table 6.4) are typically not, they increase as one moves away from the origin. However, the orthogonality conditions (6.17) localize the functions $\mathsf{L}^T * \boldsymbol{\varphi}_i$ which has the effect of taming the growth of the solution at infinity. Dropping the scale invariance requirement also leads to basis functions that grow more slowly [70].

The polynomials of order $M-1$ are called the *kernel* of the semi-norm $\|\mathbf{f}\|_{D_M}$, because they yield zero when measured with this semi-norm.

It can be shown that if two functions $\mathbf{f}_1, \mathbf{f}_2$ both solve the interpolation problem, then their difference belongs to the kernel (see also Chapter 7). As the dimensionality of the kernel is usually much smaller than the number of measurement points, this implies that the solution to the interpolation problem is unique, except for rare pathological cases. In particular, we refer to the results on unicity of Micchelli [128] and Powell [72], that can be adapted to special cases of our formulation.

The value of the criterion (6.4) applied to the solution can be determined easily as

$$J(\mathbf{f}_{\text{out}}) = \sum_{ij} \lambda_{ij} s_{ij} \tag{6.19}$$

Thus, the coefficients λ_{ij} can be also interpreted as the weights of the corresponding constraints.

6.3.5 Generalized approximation problem

In some applications, for example if the measurements are noisy, we do not want the reconstructed function $\mathbf{f}_{\text{out}}$ to pass exactly through the measured points. Instead, we want it to be a compromise between its smoothness (or plausibility), as measured by the criterion J, and the closeness of the fit to the sampled points, as measured for example by the sum of the squared differences. To define the *generalized approximation problem* (GAP) we introduce a combined criterion J_a. For the standard regularized least-squares approximation, J_a has the form

$$J_a(\mathbf{f}) = J(\mathbf{f}) + \gamma \underbrace{\sum_{ij} \left(\mathbf{h}_i^T * \mathbf{f}(\mathbf{x}_{ij}) - s_{ij} \right)^2}_{\text{data term } J_d} \tag{6.20}$$

where J is the regularization criterion defined by (6.4) and s_{ij} are the measured points close to which we want to pass.

We say that $\mathbf{f}_{\text{out}} \in F$ is a solution to the generalized approximation problem GAP$(\mathsf{L}, \mathsf{H}, \mathsf{X}, \mathbf{s}, \gamma)$, iff for all functions $\mathbf{f} \in F$ we have $J_a(\mathbf{f}) \geq J_a(\mathbf{f}_{\text{out}})$.

Note that this problem is equivalent to the following one: Find $\mathbf{f}_{\text{out}}$ which minimizes J_a under the constraint $J_d \leq \varepsilon$, where ε is an *a priori* given error bound. The γ should be chosen such that the error ε corresponds to the expected noise (error) in the measurements. If the measurement noise is not known, γ can be found for example using the leave-one-out technique [129].

The problem of solving the GAP is closely related to the interpolation problem presented, as demonstrated by the following Theorem.

Theorem 2 (GAP solution) *The generalized approximation problem GAP$(\mathsf{L}, \mathsf{H}, \mathsf{X}, \mathbf{s}, \gamma)$ is solved by the function $\mathbf{f}_{out}$ defined by* (6.15)*, if and only if $\mathbf{f}_{out}$ belongs to F, and the parameters a_i and λ_{ij} satisfy the matrix equation*

$$\underbrace{\begin{bmatrix} \mathsf{A} + \gamma^{-1}\mathsf{I} & \mathsf{Q} \\ \mathsf{Q}^T & \mathbf{0} \end{bmatrix}}_{\mathsf{B}} \begin{bmatrix} \boldsymbol{\lambda} \\ \mathbf{a} \end{bmatrix} = \begin{bmatrix} \mathbf{s} \\ \mathbf{0} \end{bmatrix} \tag{6.21}$$

where the symbols are the same as in (6.18)

Note the fundamental similarity of (6.21) to (6.18) and the convenient form of the regularization which amounts to simply adding a constant diagonal matrix term to the equation system. The simplicity of (6.21) comes from the continuous regularization and from the fact of using the fundamental solutions as the basis of our space.

Our variational formulation of the GAP is similar in spirit to using discrete regularization to deal with the ill-poseness of some inverse problems. Our regularization,

however, is completely specified in the continuous domain. Also, those discrete regularizations, often used in combination with nontrivial basis functions, such as the finite element method (FEM), modify the equation set in a much more complicated and less predictable way: the identity matrix $\mathbf{I}$ is replaced by some general matrix which needs to be determined on a case-by-case basis.

If the data term is not quadratic, but still depends only on the sampled values, the solution conserves the form (6.15) but the equation set (6.21) becomes generally nonlinear, c.f. Chapter 7.

6.4 Examples

We now present some examples of how the theory can be used.

6.4.1 Reconstruction from irregular samples

Let us consider the problem of finding a function $f : \mathbb{R} \to \mathbb{R}$, passing through a finite number of points (x_i, y_i) and minimizing a criterion $J(f) = \|f\|_{D_2}^2 = \|f''\|^2$ (see Table 6.2). From Table 6.4 we see that the fundamental solution corresponding to the semi-norm J is proportional to $|x|^3$. The kernel corresponding to this semi-norm is the class of all linear polynomials $a_0 + a_1 x$; i.e.; the class of functions for which $f'' = 0$ everywhere. The reconstruction is thus

$$f(x) = a_0 + a_1 x + \sum_{i=1}^{N} \lambda_i |x - x_i|^3 \tag{6.22}$$

which has $N + 2$ unknown parameters. The consistency conditions $f(x_i) = y_i$ give us N linear equations, while the orthogonality requirements $\sum \lambda_i = 0$ and $\sum \lambda_i x_i = 0$ yield the remaining two. A nice consequence of the orthogonality conditions is to make the second derivative of $f''(x) = 6 \sum_i \lambda_i |x - x_i|$ vanish after the last sampling point which ensures that $J(f) < \infty$ and thus $f \in F$. Note, that f is a piecewise cubic polynomial with continuous second derivatives; i.e., it is a *cubic spline*. This result is known, see [77, 130]. An example of a spline reconstruction (interpolation) is shown in Figure 6.4.

For uniform sampling, the basis functions $|x - x_i|^3$ can be localized using digital filtering (with iterated finite difference filter) to obtain compactly supported uniform cubic B-splines, which makes an interesting link with existing theory [4, 5]. For non-uniform sampling, the localization is also possible using divided differences [131] leading to non-uniform B-splines. In both cases, if we increase the order M of the semi-norm, the order of the splines will increase, too, and the corresponding interpolation will converge to the sinc interpolation [132, 133]. This shows the relation with the sampling theorem (6.1).

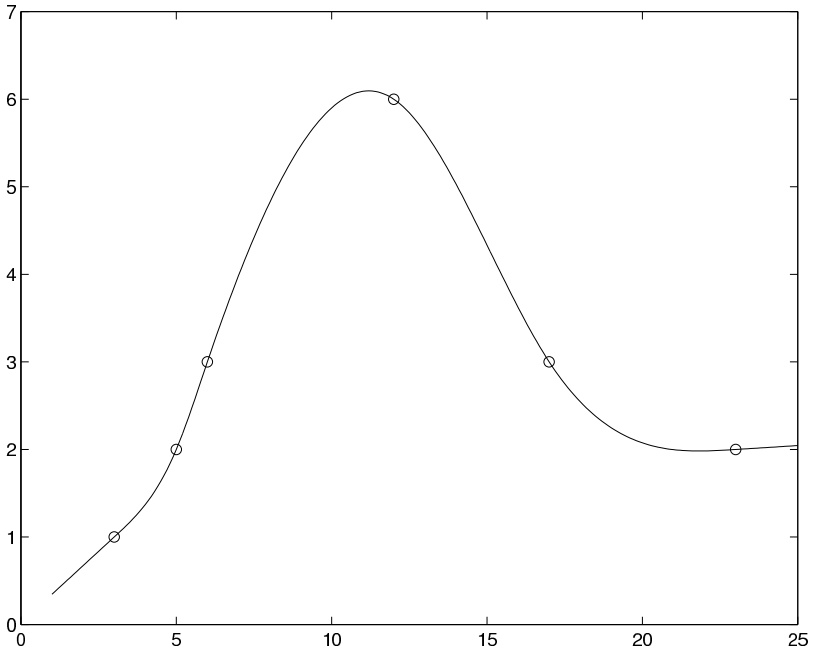


Figure 6.4: Interpolation from function values.

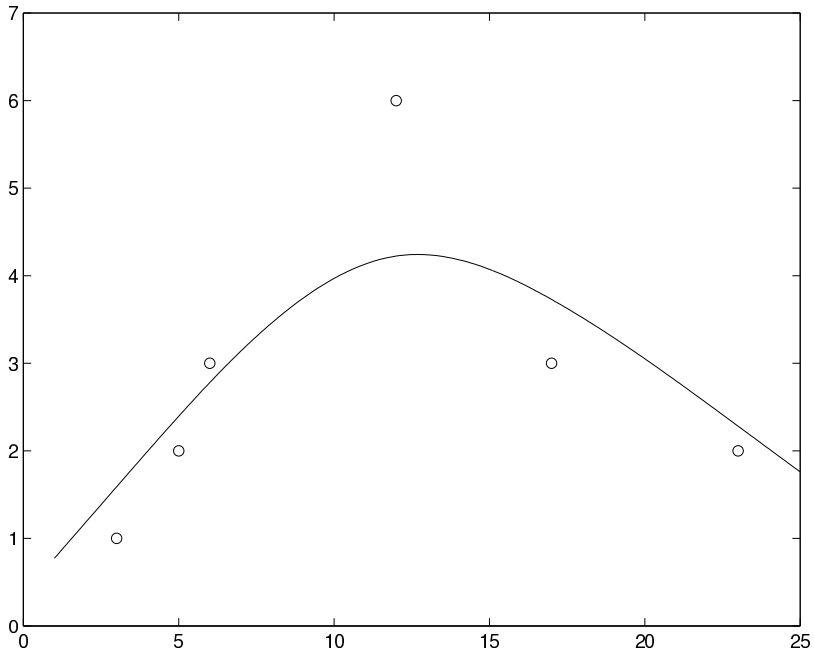


Figure 6.5: Approximation between given points for $\gamma = 10^{-3}$.

6.4.2 Smoothing splines

If the measured points from the preceding example are not exact, it is more appropriate to replace interpolation by approximation (see Section 6.3.5). The reconstruction formula (6.22) remains, while the equation set (6.18) used to determine the parameters a_i and λ_i is now replaced by the equation set (6.21). An example of a result for the same sampled points as before is shown in Figure 6.5. The smoothing spline method that we have just described is a non-parametric regression technique widely used in statistics [96].

6.4.3 Derivative sampling

Let us add derivative constraints $y'_i = f'(x_i)$ to the example from Section 6.4.1. The sampling filters will become $\mathsf{H} = \begin{bmatrix}\delta & \delta'\end{bmatrix}$. The first fundamental solution corresponding to $h_1 = \delta$ remains $\varphi_1 = c|x|^3$. The second one, corresponding to $h_2 = \delta'$, is obtained by

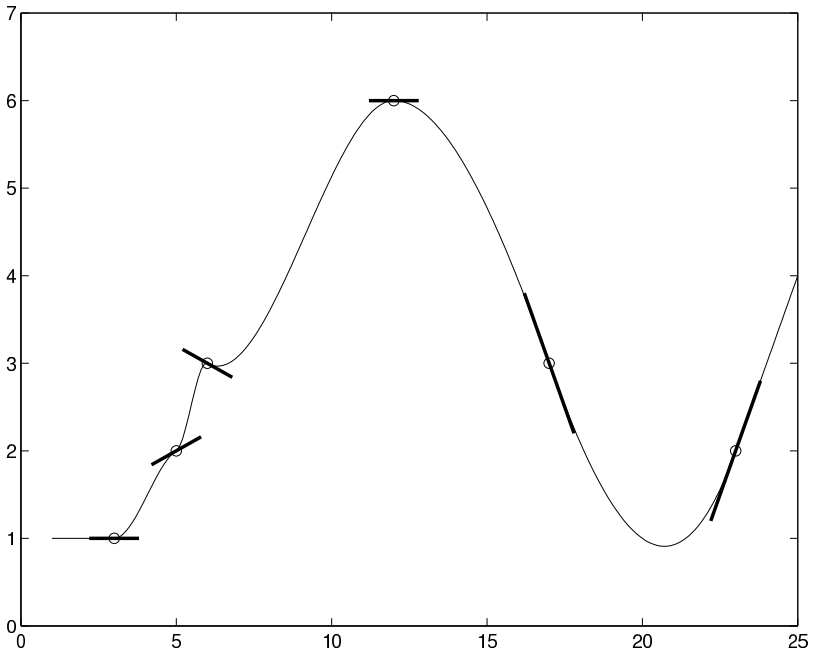


Figure 6.6: Interpolation from function values and derivatives.

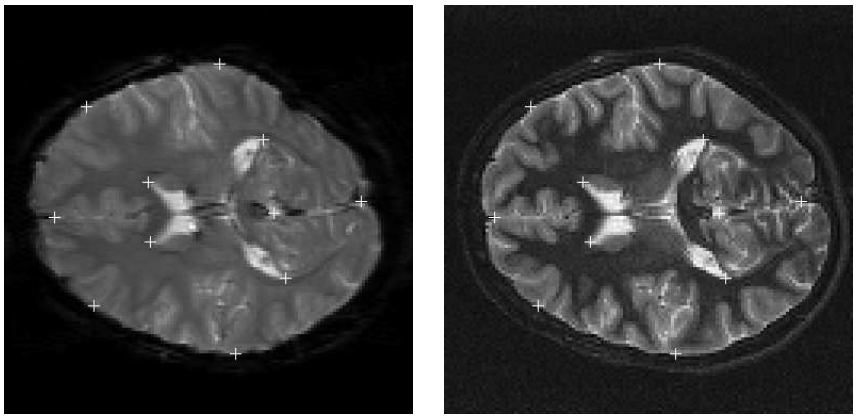
convolving φ_1 with h_2 which gives $\varphi_2 = 3c|x|x$. The reconstruction formula is thus

$$f_{\text{out}}(x) = a_0 + a_1 x + \sum_{i=1}^{N} \lambda_{i,1}|x - x_i|^3 + 3\lambda_{i,2}|x - x_i|(x - x_i) \tag{6.23}$$

The $2N + 2$ unknown parameters can be determined from $2N$ consistency equations $f(x_i) = y_i$ and $y_i' = f'(x_i)$ and two orthogonality conditions $\sum \lambda_{i,1} = 0$ and $\sum \lambda_{i,2} + \lambda_{i,1} x_i = 0$. These orthogonality conditions come from the requirement that $\sum_i \lambda_{i,1} p_k(x_i) + \lambda_{i,2} p_k'(x_i) = 0$ where $p_1 = 1$, $p_2 = x$ is the basis of the kernel. An example of reconstruction from derivative sampling is shown in Figure 6.6. A trivial extension is to sample the derivative values at different points than the function values.

6.4.4 Landmark based warping

The problem of image registration is encountered in many areas of image procesing. The task is to find correspondences between pixel coordinates in two distinct but similar images. In other words, we search for a function which gives us for each point in the first image the coordinates of the corresponding point in the second image.



(a) (b)

Figure 6.7: Corresponding slices of (a) functional MR and (b) anatomical (proton density) MR images with landmarks put manually at important points.

Here we choose an example of non-linear registration of pairs of MR images. Possible application include distortion compensation [78], inter-subject alignment and many others, see for example [86]. In some cases, it is necessary to use manual methods [15]. These mostly require the expert to specify a set of pairwise corresponding landmarks [14] (reference points) in both images. Then, an interpolation method is needed to find the deformation function also between the landmarks, which is exactly the problem studied in this paper. Supposing we want to find the 2D deformation function minimizing Duchon's semi-norm of order two, we see from Table 6.4 that we need to interpolate using the $r^2 \log r$ functions, also called thin-plate splines interpolation.

Figure 6.7 shows an example where landmark warping is used to compensate distortion in functional MRI (fMRI) images by registration with anatomically correct (proton density) MR images. More examples of landmark interpolation using different interpolating functions can be found on our web page `http://bigwww.epfl.ch/demo`.

6.4.5 Reconstruction consistent with Laplace equation

The problem treated in [134] by numerical integration— which we shall solve explicitly here—consists of reconstructing a function $\mathbb{R}^3 \rightarrow \mathbb{R}$ minimizing the norm of the 3D Laplacian operator $J_\Delta(f)^2 = \int_{\mathbb{R}^3} \|\Delta f\|^2 \mathrm{d}\mathbf{x}$. The problem is ill-posed without additional constraints, because the kernel $\mathcal{K}_\Delta$ is too big, including all functions that satisfy Laplace's equation $\Delta f = 0$, such as $x^2 - y^2$. It therefore permits an infinity of solutions with zero cost. To avoid this ambiguity, we will instead minimize a criterion $J(f) = \|f\|_{D_2}$, the

explicit expresssion of which can be found in Table 6.2. This makes sense because when $J < \infty$, the two criteria J and J_Δ are equivalent. The corresponding fundamental solution verifying $\Delta^2\varphi = c\,\delta$ in 3D is $\varphi(\mathbf{x}) = r$ (where $r = \|\mathbf{x}\|$). Since the kernel consists of linear polynomials, the solution takes the form

$$f_{\text{out}}(\mathbf{x}) = \begin{bmatrix} a_0 & a_1 & a_2 & a_3 \end{bmatrix} \begin{bmatrix} 1 \\ \mathbf{x} \end{bmatrix} + \sum_{i=1}^{N} \lambda_i \|\mathbf{x} - \mathbf{x}_i\| \tag{6.24}$$

with the auxiliary conditions $\sum_i \lambda_i = 0$, and $\sum_i \lambda_i x_i = \sum_i \lambda_i y_i = \sum_i \lambda_i z_i = 0$ for $j = 1, 2, 3$, where $\mathbf{x}_i = [x_i\ y_i\ z_i]^T$ are the coordinates of the i-th measurement point. As before, the coefficients a_i and λ_i must be determined in such a way that f_{out} passes by the desired points.

6.4.6 Derivative sampling in 2D

Adding another level of complexity, we are going to extend the derivative sampling from Section 6.4.3 to two dimensions. The task is to find a function $f : \mathbb{R}^2 \to \mathbb{R}$ given its values $f(\mathbf{x}_i)$ as well as the values of its first partial derivatives $\nabla_{\mathbf{x}} f(\mathbf{x}_i)$ at sampling points $\mathbf{x}_i = [x_i\ y_i]^T$. Our analysis filters are therefore:

$$\mathsf{H} = \begin{bmatrix} \delta & \dfrac{\partial \delta}{\partial x} & \dfrac{\partial \delta}{\partial y} \end{bmatrix}$$

For reasons given later, we choose f that minimizes $J(f) = \|f\|_{D_3}$. The kernel of this criterion consists of bivariate polynomials of degree less than or equal to 2 and its Green function (see Table 6.4) is $\psi(\mathbf{x}) = c\,r^4 \log r$. Consequently, the fundamental solutions φ_i corresponding to the three sampling filters are respectively ψ, and its partial derivatives with respect to both x and y.

This means that the solution f_{out}, besides the term from the kernel, consists of a linear combination of shifted fundamental solutions (see (6.15))

$$f_{\text{out}}(x, y) = a_0 + a_1 x + a_2 y + a_3 x^2 + a_4 y^2 + a_5 xy + \sum_{i=1}^{N} \boldsymbol{\lambda}_i^T \begin{bmatrix} \psi(\mathbf{x} - \mathbf{x}_i) \\ \frac{\partial \psi}{\partial x}(\mathbf{x} - \mathbf{x}_i) \\ \frac{\partial \psi}{\partial y}(\mathbf{x} - \mathbf{x}_i) \end{bmatrix} \tag{6.25}$$

where $\boldsymbol{\lambda}_i = [\lambda_{i,1}\ \lambda_{i,2}\ \lambda_{i,3}]^T$. The expression for f contains $6 + 3N$ unknowns. We will calculate them using the interpolation (or consistency) constraints and orthogonality constraints. To evaluate the interpolation constraints (6.16), we need to calculate partial derivatives such as

$$\frac{\partial f_{\text{out}}}{\partial x} = a_1 + 2a_3 + a_5 y + \sum_{i=1}^{N} \boldsymbol{\lambda}_i^T \begin{bmatrix} \frac{\partial \psi}{\partial x}(\mathbf{x} - \mathbf{x}_i) \\ \frac{\partial^2 \psi}{\partial x^2}(\mathbf{x} - \mathbf{x}_i) \\ \frac{\partial^2 \psi}{\partial x \partial y}(\mathbf{x} - \mathbf{x}_i) \end{bmatrix} \tag{6.26}$$

There are six orthogonality constraints (from (6.17)) corresponding to the six basis functions of the kernel:

$$1: \sum_i \lambda_{i,1} = 0 \qquad xy: \sum_i x_i y_i \lambda_{i,1} + y_i \lambda_{i,2} + x_i \lambda_{i,3} = 0$$
$$x: \sum_i x_i \lambda_{i,1} + \lambda_{i,2} = 0 \qquad y: \sum_i y_i \lambda_{i,1} + \lambda_{i,3} = 0$$
$$x^2: \sum_i x_i^2 \lambda_{i,1} + 2x_i \lambda_{i,2} = 0 \quad y^2: \sum_i y_i^2 \lambda_{i,1} + 2y_i \lambda_{i,3} = 0$$

Note that some care is needed in selecting the regularization criterion J. Had we chosen the classsical criterion $\|f\|_{D_2}$, we would have obtained $\psi(\mathbf{x}) = r^2 \log r$ as the fundamental solution, the second derivative of which is not bounded around zero, thus preventing the evaluation of (6.25) at grid points. In other words, no function f_{out} that we would have found by substituting into (6.25) some nontrivial $\boldsymbol{\lambda}$, would belong to F (see Theorem 1); this means that the interpolation problem does not admit a solution in F.

As an example, we have approximated a 2D Gaussian using thin-plate splines, $r^4 \log r$ functions without and with the derivative information. The results are shown in Figure 6.8. Not surprisingly, the method using the derivatives we have just described gives the best results.

The problem just described can be easily extended for finding vector functions $\mathbf{f}: \mathbb{R}^2 \to \mathbb{R}^2$ by taking $J(\mathbf{f}) = J(f_x) + J(f_y)$, where $(f_x, f_y) = \mathbf{f}$ are the components of $\mathbf{f}$. As the components are treated separately, the solution can be calculated independently for each of them. One possible application might be a semi-automatic landmark image warping with derivative constraints.

6.4.7 Tomographic reconstruction

A nice example of a classic inverse problem that also falls into our framework is tomographic reconstruction [135,136]. It consists of reconstructing a cross section of an object from its transaxial projections. We now show that tomographic reconstruction lends itself well to the variational formulation.

Let $f(x, y)$ be the unknown cross section of the object to be reconstructed. We measure the projections of f at q angles θ_i. For each angle, we measure an integral along a ray at N positions u_j, that is

$$s_{ij} = \int_{\mathbb{R}} f(t \cos\theta_i - u_j \sin\theta_i, t \sin\theta_i + u_j \cos\theta_i) \mathrm{d}t \tag{6.27}$$

This integral corresponds to our sampling operator. The variational formulation of the reconstruction problem is thus: Find a function f_{out} consistent with measurements $\mathbf{s}$ (yielded by (6.27)) and minimizing a plausibility criterion J. We choose J to be Duchon's semi-norm $J(f) = \|f\|_{D_2}$.

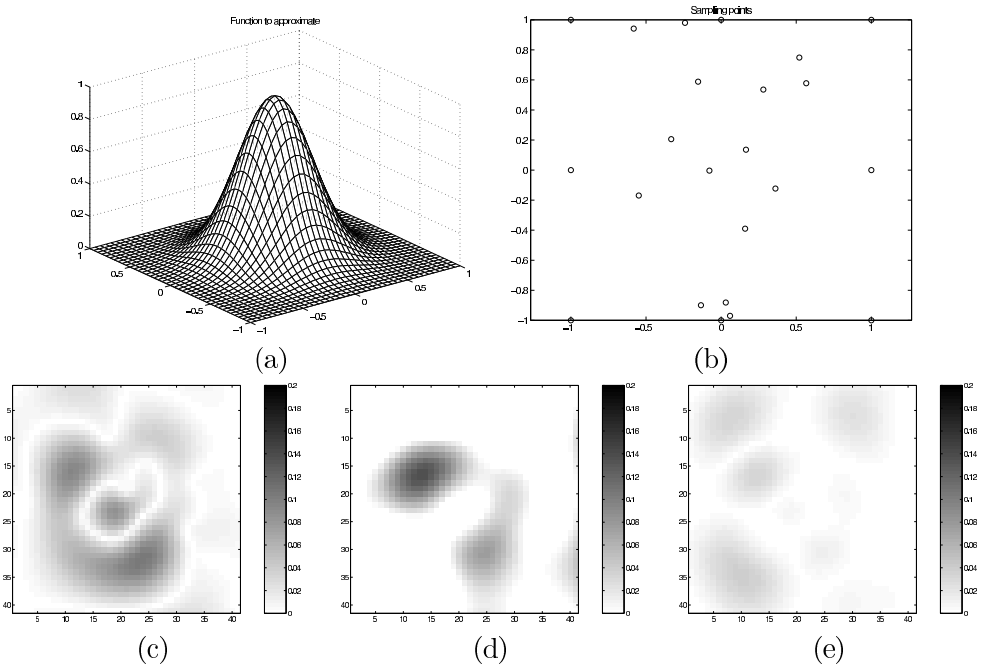


Figure 6.8: Approximating a 2D Gaussian (a) using sampling points (b). The difference between the true function and (c) thin-plate spline $r^2 \log r$ approximation, (d) approximation using $r^4 \log r$, and (e) approximation using $r^4 \log r$ with measurements of derivatives. The respective SNRs are: (c) 39.7 dB, (d) 25.7 dB, (e) 59.0 dB. The approximation using derivatives gives the best results. Bear in mind, though, that the derivative method uses an extra information (the derivatives) that the other methods cannot use.

The projection/sampling operator (6.27) can be written as a convolution

$$s_{ij} = \langle \delta(x \sin\theta_i - y\cos\theta_i + u_j), f \rangle$$
$$= (\underbrace{\delta(-x\sin\theta_i + y\cos\theta_i)}_{h_i} * f)(\underbrace{-\sin\theta_i u_j, \cos\theta_i u_j}_{\mathbf{x}_{ij}}) \quad (6.28)$$

The fundamental solution φ_i for the sampling operator (6.28) needs to satisfy the equation (6.12) which in this case has the form

$$\Delta^2 \varphi_i(x, y) = \delta(-x\sin\theta_i + y\cos\theta_i) \quad (6.29)$$

To find φ_i, we rotate our coordinate system by $-\theta_i$, which yields

$$\Delta^2 \varphi_i(x', y') = \delta(x') \quad (6.30)$$

because of the rotational invariance of the Δ^2 operator. Consequently, a 1D function $\varphi(x', y') = \varphi(x')$ that solves (6.30) also solves the corresponding 1D problem of Section 6.4.1, and *vice-versa*. The fundamental solutions of the 1D problem are $|x'|^3$. In our 2D case, after rotating the coordinate system back, we get

$$\varphi_i = |-\sin\theta_i x + \cos\theta_i y|^3 \quad (6.31)$$

Putting the pieces together, we find that our reconstruction takes the form

$$f_{\text{out}}(x, y) = [a_0\, a_1\, a_2] \begin{bmatrix} 1 \\ x \\ y \end{bmatrix} +$$
$$\sum_{j=1}^{q} \sum_{i=1}^{N} \lambda_{ij} |-\sin\theta_i x + \cos\theta_i y - u_j|^3 \quad (6.32)$$

The interesting thing is the structure of the generating functions φ_i that are back-projections (extensions) of the corresponding 1D fundamental solutions along the projection rays. This is the same form as the result of the standard back-projection algorithm [135].

For a more realistic application, we consider that the measurements $\mathbf{s}$ are noisy. We therefore use the approximation formulation from Section 6.3.5. Second, instead of integrating over the whole space in (6.27),(6.28), we only integrate over the part corresponding to the measurement device. If we also evaluate the regularization criterion in the same domain, the fundamental solutions φ remain the same.

The drawback of the method in its presented form is the necessity to solve a large linear equation set which is often ill-conditioned when the matrix gets large. The conditioning is improved by working in an equivalent basis of localized versions of the generating functions φ_i. To speed up the process, specialized iterative solvers can be used.

Figure 6.9 shows a comparison of the reconstruction using the variational algorithm and classical filtered back-projection [135] as implemented in Matlab. We can observe that for a small number of measurements, the variational reconstruction algorithm gives better result than the filtered back-projection. For a large number of measurements, the results of the variational reconstruction are comparable to that of the filtered back-projection. Thus, our method is especially useful in the case of few angles. More details can be found in [137].

6.5 Conclusions

We have presented an interpolation and approximation scheme capable of treating nonuniformly sampled multichannel output of a filter-bank. The reconstruction is optimal in the sense of a user-chosen criterion.

The main advantages of our theory include generality with respect to the sampling operators and sampling locations. It works for multivariate and multidimensional functions. It places only negligible additional constraints on the input signal. Unlike traditional approaches, we do not specify the reconstruction space *a priori*. Instead, our method yields basis functions (and thus the reconstruction space) that are optimal in a well-defined sense for the problem at hand. It transforms a continuously-defined variational problem into a generally well-posed algebraic problem; the solution thereof is then obtained from a system of linear equations. The method can also accommodate for non-exact fits including non-linear data terms. In the least-squares case, the system matrix is the regularized version of the matrix in the interpolation case. Otherwise, more general multidimensional nonlinear problem solver is needed.

Presently, the weak point of our method is that it requires the solution of a large, non-sparse system of equations. Additional research is required to develop fast numerical solvers. Related aspect is the ill-conditioning of the system matrix due to the non-local nature of the basis functions. We believe this can be improved using adequate preconditioners, e.g., by localizing the basis functions, similar to the construction of B-spline basis [6].

We believe that our reconstruction algorithm would be especially useful for applications where there are few measurements as it permits to use them in the best possible way. We have presented several examples to illustrate its possible uses.

Our theory includes and explains many previously used methods; e.g., the well known thin-plate splines.

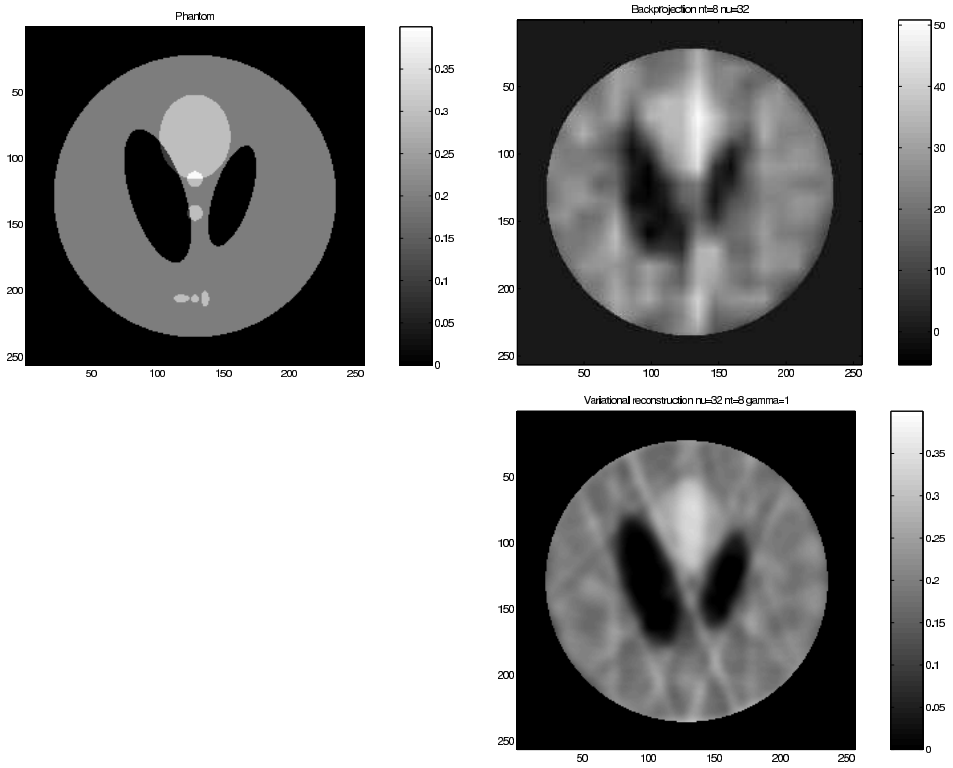


Figure 6.9: We are reconstructing the inner part of the Shepp and Logan phantom [135] (top left), using projections at 8 uniformly distributed angles, 32 measurements per angle. The result of the filtered back-projection (top right) has more artifacts and is less geometrically precise than the variational reconstruction (bottom right). Note also, that the unlike filtered backprojection, the variational reconstruction recovers well the absolute amplitudes in the image.

Chapter 7

Variational Reconstruction: Theory

This chapter is based on our article [138].

7.1 Abstract

The problem of generalized sampling described in Chapter 6 leads to the minimization of a variational functional under linear constraints. Imposing generally desirable properties on the solution limits largely the choice of the criterion. Linearity leads to a quadratic criterion based on bilinear forms. Specifically, we show that the requirements of translation, rotation, and scale invariance restricts the form of the criterion to essentially a one-parameter family. We show that the solution can be obtained as a linear combination of generating functions. We provide techniques to find these functions, and the solution itself. Throughout the paper, the affirmations from the companion paper are formulated more precisely, generalized, and also justified.

7.2 Introduction

In the preceding Chapter 6, we have presented a variational approach to generalized sampling. The basic problem is to reconstruct a signal from a series of measurements which are obtained by sampling one (or several) filtered version(s) of the original signal. Our formulation leads to a functional minimization problem under linear constraints. The original aspect is that the optimization is performed in the continuous domain rather than in the discrete one, as is usually the case with this type of application. Our method

is rather general and can handle all kinds of sampling configurations, including non-uniform and/or non-ideal sampling. It is also suitable for inverse problems where the measurement operator is shift-invariant; a typical example is tomographic reconstruction from projection data [137].

In this paper, we present the mathematical foundations of the method. We will adopt a slightly more general formulation of the problem and consider measurements that can be any linear functionals of the input signal. We will concentrate on minimizing quadratic energy functionals, as this yields a vector space characterization of the solution as a linear combination of basis functions. The key feature here is that the basis functions themselves are the result of a mathematical optimization. Consequently, they are optimally tailored to the problem at hand.

This article has four primary goals:

(1) To provide a precise mathematical formulation of generalized sampling in a variational setting. This is done in Sections 7.3.3 where we also state our assumptions and list some of the general properties of the solution.

(2) To understand and control the key properties of the solution through an appropriate selection of the regularization criterion. To this end, we investigate quadratic shift-invariant criteria and their corresponding bilinear forms. In Section 7.4, we provide the corresponding convolutional kernel representation in both time and frequency domains. Our strategy is to impose some desirable properties on the solution (enumerated in Section 7.3.4) and to infer the corresponding class of criteria. We find that a small set of perfectly justifiable requirements, such as rotation and scale invariance, essentially limits the degrees of freedom to a one-parameter family of criteria. This is formalized in Theorem 3 at the end of Section 7.5.

(3) To solve our generalized sampling problem under hard constraints (consistency requirement). The general solution is derived in Section 7.5 and described in Theorem 4. We show how to construct a basis for the solution space. The critical step involves finding the Green functions of the operator associated with the bilinear form of the criterion. The solution usually includes an additional polynomial term whose main effect is to make the reconstruction well-behaved far from the sampling points. These results lead to the specification of the linear system of equations that yields the optimal coefficients for the solution.

(4) To solve our generalized sampling problem under soft constraints. The idea here is to consider a cost function that is the sum of a non-linear data term and the same regularization criterion as before. In Section 7.6, we prove that the solution of this approximation problem—irrespective of the form of the data term— lies in the same subspace as in the previous case (hard constraints) (cf. Theorem 5). We also work out an explicit formula for the least-squares case. Interestingly, this solution can be obtained by an almost trivial modification of the hard-constrained equations

(addition of constant diagonal term to the system matrix). The advantage of this approach is that it stabilizes the reconstruction. It is also better suited for noisy situations where it is often counterproductive to impose hard constraints.

7.3 Formulation

7.3.1 Notation

We denote vectors by bold letters and consider them as columns, that is $\mathbf{x} = [x_1 \dots x_m]^T$. Matrices will be denoted by upright letters (X) with elements $(\mathsf{X})_{i,j} = x_{i,j}$

We define a scalar product of two multivariate vector functions as $\langle \mathbf{f}, \mathbf{g} \rangle = \int_{\mathbb{R}^m} \mathbf{f}(\mathbf{x})^T \mathbf{g}(\mathbf{x}) \mathrm{d}\mathbf{x}$. By extension, the notation $\langle \mathsf{X}, \mathbf{f} \rangle$ applied on a matrix X and a vector $\mathbf{f}$ is a vector of scalar products between columns of X and $\mathbf{f}$. Similarly, we define a convolution of vector and matrix functions following the usual rules for matrix multiplication, for example $\mathsf{X} * \mathbf{y} = \mathbf{z}$ means $\sum_j x_{i,j} * y_j = z_i$.

The most important symbols used are given by the following list:

$\mathcal{A}$	the aproximation problem
$B(\mathbf{f}, \mathbf{g})$	bilinear form associated to $J(\mathbf{f}) = B(\mathbf{f}, \mathbf{f})$
c	a constant
$\mathscr{D}$	space of test functions (indefinitely differentiable and compactly supported)
$\mathbf{e}_i$	unitary basis vector
$\hat{f}, \mathscr{F}f$	Fourier tranform of f, $\hat{f}(\boldsymbol{\omega}) = \int f(\mathbf{x}) \mathrm{e}^{-i\boldsymbol{\omega}^T \mathbf{x}} \mathrm{d}\mathbf{x}$
f_{out},$\mathbf{f}_{\text{out}}$	reconstructed function
F	reconstruction space
$\mathbf{h}_i$	sampling filter
H	sampling filterbank $\mathsf{H}_{N \times n} = [\mathbf{h}_1 \dots \mathbf{h}_q]$
I	identity matrix
$J(\mathbf{f})$	positive quadratic plausibility criterion
J_a	criterion for the approximation problem
J_*	extended criterion
$\mathcal{K}$	kernel of the criterion J
L	filterbank defining J
m,n	dimensionality of the function $\mathbf{f} : \mathbb{R}^m \to \mathbb{R}^n$
N	number of constraints
p_k,$\mathbf{p}_k$	basis vector of the kernel $\mathcal{K}$
P	dimensionality of $\mathcal{K}$
$\mathcal{P}$	the variational problem
p	number of columns of L
s_{ij}	measurements

U	convolutional kernel of the shift-invariant bilinear form, $\mathsf{U}(\mathbf{x}-\mathbf{y}) = \mathsf{V}(\mathbf{x},\mathbf{y})$
V	operator kernel of the bilinear form
λ_{ij},$\boldsymbol{\lambda}_i$	Lagrange multipliers, coefficients
φ,$\boldsymbol{\varphi}$	fundamental solution, generating function
ψ,$\boldsymbol{\psi}$	Green function
ω,$\boldsymbol{\omega}$	angular frequency
$\|\mathbf{f}\|_{M,s}$	Duchon's semi-norm
$\langle \mathbf{f}, \mathbf{g} \rangle$	L_2 scalar product
$\|\mathbf{f}\|_{L_2}$	L_2 norm, $\langle \mathbf{f}, \mathbf{f} \rangle^{1/2}$

7.3.2 Distributions

Many results in this article are obtained through calculations in the sense of *distributions*. The basic reference here is Schwartz [139] or a gentler introduction [140]. A distribution u is a function-like object defined indirectly through its scalar products $\langle u, v \rangle$ with arbitrary *test functions* v from the space $\mathscr{D}$ of compactly supported and infinitely differentiable functions. Distributions are generalizations of functions and can often be operated upon using the same rules, except, noteworthy, the multiplication. The majority of the practically-used functions are indeed distributions. The best known example of a distribution which is not a function is Dirac's δ, defined as $\langle \delta, v \rangle = v(0)$. Similarly, its derivative δ' gives $\langle \delta', v \rangle = -v'(0)$. Consequently, convolving $\delta * v$ yields v, while $\delta' * v = v'$.

7.3.3 Problem definition

The variational problem we consider consists of finding a vector function $\mathbf{f} : \mathbb{R}^m \to \mathbb{R}^n$ minimizing a non-negative functional criterion $J(\mathbf{f})$ under a finite number of constraints $\langle \mathbf{h}_i, \mathbf{f} \rangle = s_i$, where $i = 1 \dots N$, and $\mathbf{h}_i$ are sampling filters. The expression $\langle \mathbf{h}_i, \mathbf{f} \rangle$ linearly maps functions to real scalars. As $\mathbf{h}_i$ is a distribution, most linear forms can be written in this way. Alternatively, the constraints can be also written as convolutions:

$$\langle \mathbf{h}_i, \mathbf{f} \rangle = \mathbf{h}_i^{RT} * \mathbf{f}(\mathbf{0}) \tag{7.1}$$

where $\mathbf{h}_i^R(\mathbf{x}) = \mathbf{h}_i(-\mathbf{x})$ is a time-reversed version of $\mathbf{h}_i$.[1]

When $\mathbf{f}$ satisfies all the N constraints $\langle \mathbf{h}_i, \mathbf{f} \rangle = s_i$, we write

$$\langle \mathsf{H}_{N \times n}, \mathbf{f} \rangle = \mathsf{H}^{RT} * \mathbf{f}(\mathbf{0}) = \mathbf{s} \tag{7.2}$$

[1] For the sake of notational simplicity, we use a time-reversed version of $\mathbf{h}$ with respect to Chapter 6.

We only consider functions from a space F induced by the criterion J and measurable by the filters H

$$F = \left\{ \mathbf{f} : \mathbb{R}^m \to \mathbb{R}^n;\ J(\mathbf{f}) < \infty \quad \text{and} \quad \left\| \langle \mathsf{H}, \mathbf{f} \rangle \right\| < \infty \right\} \tag{7.3}$$

where $\| \cdot \|$ is the usual Euclidean norm of vectors in $\mathbb{R}^N$. We say that $\mathbf{f}_{\text{out}}$ solves the variational problem $\mathcal{P}(J, \mathsf{H}, \mathbf{s})$ iff $\mathbf{f}_{\text{out}}$ minimizes J in F under constraints (7.2). Note that there can be more than one of such functions $\mathbf{f}_{\text{out}}$ with the same value of $J(\mathbf{f}_{\text{out}})$.

The main application of the variational problem just described is the generalized reconstruction (in Chapter 6) where $\mathbf{h}_i$ represent sampling filters, s_i are samples and J is a regularization criterion. So as to be useful in this (and other) contexts, the solution should satisfy a certain number of properties, which will in turn impose constraints on the criterion J and the filters $\mathbf{h}_i$.

7.3.4 Properties of the solution

Ideally, we want the problem and its solution to have the properties detailed below. We will see that they help us to specify an essentially one-parameter family of criteria. We will be able to give a constructive theorem concerning the existence property, obtain unicity in the majority of useful cases, and guarantee the invariance and linearity of the solution in the sense detailed below.

For each property, we give an indication of how it can be verified or guaranteed. Note, however, that the conditions we give are only sufficient, not necessary, because searching for necessary conditions proved to be extremely difficult and of small practical interest. On the other hand, we will see in the forthcoming sections that our conditions yield a sufficiently general family of criteria.

Proposition 1 (Existence and uniqueness) *There is exactly one solution* $\mathbf{f}_{out}$.

The motivation of the existence requirement is clear: we want our method to give us at least one solution for any possible measurements s_i. There are various reasons why the problem $\mathcal{P}$ might not have a solution. For example, when the constraints are contradictory, or when the solution space is not complete with respect to J. That is to say, if for any sequence of functions $\mathbf{f}_i$ satisfying the constraints such that the criterion $J(\mathbf{f}_i)$ is decreasing, this sequence does not converge in F. A typical example might be a sequence of continuous functions converging towards a discontinuous one, under a derivative criterion.

We also want the solution to be unique. For the uniqueness, it is useful for J to be discriminative so that as few functions $\mathbf{f}$ as possible have the same criterion value J.

In practice, we verify existence and uniqueness *a posteriori*. We first construct a function and then verify that it solves the problem and that no other function does. In some cases, the work on the *a priori* analysis has been already done [72, 128].

Proposition 2 (Vector space of solutions) *If* $\mathbf{f}$ *solves* $\mathcal{P}(J, \mathsf{H}, \mathbf{r})$ *and* $\mathbf{g}$ *solves* $\mathcal{P}(J, \mathsf{H}, \mathbf{s})$ *then* $\alpha\mathbf{f} + \beta\mathbf{g}$ *solves* $\mathcal{P}(J, \mathsf{H}, \alpha\mathbf{r} + \beta\mathbf{s})$ *for* $\alpha, \beta \in \mathbb{R}$.

This scalar linearity property ensures that the solution space is a vector space and that consequently every solution can be expressed as a linear combination of N basis functions (where N is the number of constraints). We will see later (see Section 7.4.1) that this is ensured if the criterion J is a quadratic criterion.

Proposition 3 (Matrix linearity) *If* $\mathbf{f}$ *solves* $\mathcal{P}(J, A^T\mathsf{H}, \mathbf{s})$ *then* $\mathsf{A}\mathbf{f} + \mathbf{b}$ *solves* $\mathcal{P}(J, \mathsf{H}, \mathbf{s} + \langle \mathsf{H}, \mathbf{b} \rangle)$, *where* A *is an arbitrary matrix.*

This comprises the cases of rotating, scaling, shifting, permuting, inverting, and otherwise linearly deforming the 'output' coordinate system of the function $\mathbf{f}$. We want the solution to be invariant with respect to these changes. The matrix linearity property is guaranteed if J is *pseudo-invariant* with respect to these changes, which means that the criterion value for $\mathsf{A}\mathbf{f} + \mathbf{b}$ is proportional to the criterion value for $\mathbf{f}$, namely

$$J(\mathsf{A}\mathbf{f} + \mathbf{b}) = c(\mathsf{A}, \mathbf{b}) J(\mathbf{f}) \tag{7.4}$$

where c is a continuous function of A and $\mathbf{b}$, independent of $\mathbf{f}$.

For the remainder of this paper, we consider only matrices satisfying

$$\mathsf{A}\mathsf{A}^T = k\mathsf{I} \tag{7.5}$$

(where I is an identity matrix) which corresponds to orthogonal transformations and uniform scaling. See Section 7.5.4 for details.

Proposition 4 (Geometric invariance) *If* $\mathbf{g}$ *solves* $\mathcal{P}(J, \mathsf{H}, \mathbf{s})$ *then* $\mathbf{f}$ *solves* $\mathcal{P}(J, \mathsf{H}', \mathbf{s})$, *where* $\mathbf{g}(\mathbf{x}) = \mathbf{f}(\mathsf{A}\mathbf{x} + \mathbf{b})$, *provided that* $\langle \mathsf{H}', \mathbf{f} \rangle = \langle \mathsf{H}, \mathbf{g} \rangle$ *for all* $\mathbf{f}$.

This encompasses the cases of rotating, scaling, and shifting the coordinate system of $\mathbf{x}$. We want our solution to be invariant with respect to these changes. The new filter H' can be written in the functional form as $\mathsf{H}(\mathbf{x}) = (\det \mathsf{A})\mathsf{H}'(\mathsf{A}\mathbf{x} + \mathbf{b})$. Consequently, we want A to be an invertible matrix. Similarly to Property 3, the geometric invariance can be ensured by pseudo-invariance with respect to the geometric transformations; i.e.,

$$J\big(\mathbf{f}(\mathsf{A}\mathbf{x} + \mathbf{b})\big) = c(\mathsf{A}, \mathbf{b}) J\big(\mathbf{f}(\mathbf{x})\big) \tag{7.6}$$

where c is a continuous function of A and $\mathbf{b}$, independent of $\mathbf{f}$, and with no connection to c in (7.4).

We shall impose geometric invariance only with respect to orthogonal matrices A, satisfying $\mathsf{A}\mathsf{A}^T = \mathsf{I}$. See Section 7.5.1 for details.

Proposition 5 (Density) *The solution space F contains all test functions $\mathbf{u}$ from $\mathscr{D}$. For any function $\mathbf{f} \in F$, there is a sequence of test functions $\mathbf{u}_1, \mathbf{u}_2, \ldots$ such that $\lim_{i\to\infty} J(\mathbf{f} - \mathbf{u}_i) = 0$.*

This property is indeed somewhat technical but its significance can be readily grasped. The first part ensures that the solution space is large, that it contains as many 'good' functions as possible. It guarantees that at least all test functions can be measured using the criterion J. The second part concerns the behavior of J for functions on the closure of $\mathscr{D}$, that is to say, for functions that are not in $\mathscr{D}$ but which can be expressed as a limit of a sequence of test functions. It guarantees the continuity of J on F. Consequently, we can do most of our reasoning in the space of test functions and then extend the result to the whole of F using a limiting process.

In practice, Property 5 is always satisfied by the quadratic semi-norms we will be considering; in particular, by the semi-norms of Duchon. (This originates from the density of $\mathscr{D}$ in Sobolev spaces.)

7.4 Bilinear forms

From now on, we consider exclusively those criteria that can be expressed using a non-negative *bilinear form*. A bilinear form $B(\mathbf{f}, \mathbf{g})$ maps pairs of functions $\mathbf{f}$, $\mathbf{g}$ onto $\mathbb{R}$. It is *symmetric* ($B(\mathbf{f}, \mathbf{g}) = B(\mathbf{g}, \mathbf{f})$) and *linear* ($B(\alpha\mathbf{f} + \beta\mathbf{g}, \mathbf{h}) = \alpha B(\mathbf{f}, \mathbf{h}) + \beta B(\mathbf{f}, \mathbf{g})$) with respect to both its arguments. It is *non-negative* iff $B(\mathbf{f}, \mathbf{f}) \geq 0$ for all $\mathbf{f} \in F$. We associate B with a criterion

$$J(\mathbf{f}) = B(\mathbf{f}, \mathbf{f}) \tag{7.7}$$

which we call a *quadratic criterion*. Conversely, given a quadratic criterion J, the associated bilinear form B can be obtained as

$$B(\mathbf{f}, \mathbf{g}) = \frac{1}{4}\big(J(\mathbf{f} + \mathbf{g}) - J(\mathbf{f} - \mathbf{g})\big) \tag{7.8}$$

The square-root $\sqrt{J}$ of the criterion defined in this way is a *semi-norm*; i.e., it satisfies the triangular inequality and semi-linearity ($\sqrt{J(\lambda\mathbf{f})} = |\lambda|\sqrt{J(\mathbf{f})}$). Unlike for a norm, there might be more than one $\mathbf{f}$ satisfying $J(\mathbf{f}) = 0$. Such functions define a *kernel* $\mathcal{K}$.

The criterion J is convex. The important Cauchy-Schwartz inequality $|B(\mathbf{f}, \mathbf{g})| \leq \sqrt{J(\mathbf{f})J(\mathbf{g})}$ holds too. The equality is reached for $\mathbf{f}$,$\mathbf{g}$ such that $\exists \lambda, \mu \in \mathbb{R}, \lambda\mu \neq 0$; $\lambda\mathbf{f} + \mu\mathbf{g} \in \mathcal{K}$.

7.4.1 Quadratic criterion and linear constraints

The restriction to bilinear forms is justified, namely in view of satisfying Property 2 which yields a useful vector space structure for the solution space. The proof that a quadratic criterion J implies Property 2 can be found in Appendix A.

Because of the convexity of J, if there is a local minimum, it is also the global minimum. Moreover, if two functions $\mathbf{f}_1, \mathbf{f}_2$ solve the problem $\mathcal{P}$, then their difference $\mathbf{f}_1 - \mathbf{f}_2$ necessarily belongs to $\mathcal{K}$. (See Appendix B for a proof.) Therefore if the constraints (7.2) cannot be met by two distinct functions differing by an element from the kernel, the solution is unique. This is easy to check, because in most cases of interest the kernel $\mathcal{K}$ is fairly small. We will see later that it mostly consists of low-order polynomials.

7.4.2 Operator kernel of a bilinear form

Any bilinear form satisfying very mild conditions (see [139]) can be written in the form of a scalar product:

$$B(\mathbf{f}, \mathbf{g}) = \int_{\mathbb{R}^{2m}} \mathbf{f}^T(\mathbf{x}) \mathsf{V}(\mathbf{x}, \mathbf{y}) \mathbf{g}(\mathbf{y}) \mathrm{d}\mathbf{x} \mathrm{d}\mathbf{y} \tag{7.9}$$

where V is a $n \times n$ matrix of distributions called an *operator kernel* of the bilinear form. Technically, the existence of the integral is not guaranteed unless both $\mathbf{f}$ and $\mathbf{g}$ are from the class $\mathscr{D}$ of infinitely differentiable and compactly supported *test functions*.

Without any loss of generality, we can assume V to be symmetrical ($\mathsf{V}(\mathbf{x}, \mathbf{y}) = \mathsf{V}(\mathbf{y}, \mathbf{x})$), because the operator kernel can always be symmetrized as $\frac{1}{2}\big(\mathsf{V}(\mathbf{x}, \mathbf{y}) + \mathsf{V}(\mathbf{y}, \mathbf{x})\big)$ without affecting the associated bilinear form B. By exchanging $\mathbf{f}$ and $\mathbf{g}$ we also find that V must have a matrix symmetry $\mathsf{V} = \mathsf{V}^T$. The implications of (7.4) and (7.6) on the properties of V are studied in Section 7.5.

7.4.3 Convolutional kernel

If V is translation-invariant, it can be written using a single-parametric distribution matrix $\mathsf{U}(\mathbf{x} - \mathbf{y}) = \mathsf{V}(\mathbf{x}, \mathbf{y})$. This transforms (7.9) to

$$B(\mathbf{f}, \mathbf{g}) = \int_{\mathbb{R}^m} \mathbf{f}^T(\mathbf{x}) \mathsf{U}(\mathbf{x} - \mathbf{y}) \mathbf{g}(\mathbf{y}) \mathrm{d}\mathbf{x} \mathrm{d}\mathbf{y} \tag{7.10}$$

for test functions $\mathbf{f}$,$\mathbf{g}$. We recognize the convolution here:

$$B(\mathbf{f}, \mathbf{g}) = \langle \mathbf{f}, \mathsf{U} * \mathbf{g} \rangle \quad \text{for} \quad \mathbf{g} \in \mathscr{D} \tag{7.11}$$

where the restriction of $\mathbf{g}$ to test functions is useful to ensure that $\langle \mathbf{f}, \mathsf{U} * \mathbf{g} \rangle$ exists. We call U the *convolutional kernel* of the bilinear form. Because of the symmetries of V, we have the same symmetries on U; i.e., $\mathsf{U}(\mathbf{x}) = \mathsf{U}(-\mathbf{x}) = \mathsf{U}^T(\mathbf{x})$.

7.4.4 Fourier form

Both (7.9) and (7.10) can be also calculated in the Fourier domain. For this, we need the Fourier transforms $\hat{\mathbf{f}}$, $\hat{\mathbf{g}}$, and $\hat{\mathsf{U}}$ (see Section 7.3.1 for a definition). The Fourier

transform can be calculated for test functions and extended to *tempered distributions*, such as polynomials. For example, the expression (7.10) can be written using Parseval's theorem as

$$B(\mathbf{f},\mathbf{g}) = \frac{1}{(2\pi)^m}\int_{\mathbb{R}^m} \hat{\mathbf{f}}^T(\boldsymbol{\omega})\,\hat{\mathsf{U}}(\boldsymbol{\omega})\,\hat{\mathbf{g}}^*(\boldsymbol{\omega})\,\mathrm{d}\boldsymbol{\omega} \tag{7.12}$$

7.4.5 Extending the bilinear form

The original equations (7.9),(7.10) define $B(\mathbf{f},\mathbf{g})$ only for test functions $\mathscr{D}$. However, we will need later to evaluate B also for $\mathbf{f}$ from some larger class $\overline{F} \supset F$, conserving all the properties of the bilinear form. The parameter $\mathbf{g}$ is still in $\mathscr{D}$. Already, (7.9) retains a meaning if $\mathbf{f}$ belongs to the dual (distribution) space of the space $G = \left\{\int \mathsf{V}(\mathbf{x},\mathbf{y})\mathbf{g}(\mathbf{y})\mathrm{d}\mathbf{y};\ \mathbf{g}\in\mathscr{D}\right\}$. If we define B through (7.11), it allows us to consider $\mathbf{f}$ from the class $\overline{F}$ of distributions, provided that U is compactly supported. When the convolution $\mathsf{U}^{RT} * \mathbf{f}$ exists, we can alternatively calculate

$$B(\mathbf{f},\mathbf{g}) = \langle \mathsf{U}^{RT} * \mathbf{f}, \mathbf{g}\rangle \tag{7.13}$$

More extensions are indeed possible. They all coincide for test functions but might give different results when evaluated for other functions. To work with the extended version of B we need to define an *extended kernel* of functions $\mathbf{f}$ that yield zero $B(\mathbf{f},\mathbf{g})$ for all $\mathbf{g}\in\mathscr{D}$. Recall that the kernel $\mathcal{K}$ in the classical sense is defined as a set of functions for which $J(\mathbf{f}) = 0$. Generally, the extended kernel is a superset of a classical kernel. However, for the bilinear forms considered here, the two kernels are identical. Therefore we will not distinguish between them.

7.4.6 Factorizing the convolutional kernel

An alternative, symmetric definition of B is:

$$B(\mathbf{f},\mathbf{g}) = \left\langle \mathsf{L}^T * \mathbf{f}, \mathsf{L}^T * \mathbf{g}\right\rangle \quad \text{for} \quad \mathbf{f},\mathbf{g}\in\mathscr{D} \tag{7.14}$$

which leads to a very simple expression for J:

$$J(\mathbf{f}) = \|\mathsf{L}^T * \mathbf{f}\|^2 = \sum_{i=1}^{p} \|\mathbf{l}_i^T * \mathbf{f}\|^2 \tag{7.15}$$

The convolutional operator $\mathsf{L}_{N\times p}$ has an adjoint $\mathsf{L}^\dagger = \mathsf{L}^{RT}$ (that is, $\langle \mathsf{L} * \mathbf{f}, \mathbf{g}\rangle = \langle \mathbf{f}, \mathsf{L}^{RT} * \mathbf{g}\rangle$). We obtain an equivalence to (7.11) by setting $\mathsf{U} = \mathsf{L}^{RT} * \mathsf{L}$.

There are generally many possible factorizations, leading to many extensions as detailed in the previous section. To illustrate this point we consider the example of the

scalar distribution in two dimensions: $u = \frac{\partial^4}{\partial x^4}\delta + 2\frac{\partial^4}{\partial x^2 \partial y^2}\delta + \frac{\partial^4}{\partial y^4}\delta$. It can be factorized either simply as $\mathsf{L} = \left[\frac{\partial^2}{\partial x^2}\delta + \frac{\partial^2}{\partial y^2}\delta\right]$, or alternatively as $\mathsf{L} = \left[\frac{\partial^2}{\partial x^2}\delta \quad \sqrt{2}\frac{\partial^2}{\partial x \partial y}\delta \quad \frac{\partial^2}{\partial y^2}\delta\right]^T$.

The latter factorization leads to the Duchon's semi-norm (see Section 7.5.7)

$$\|f\|^2_{D_2} = \int_{\mathbb{R}^2} \left(\frac{\partial^2 f}{\partial x^2}\right)^2 + 2\left(\frac{\partial^2 f}{\partial x \partial y}\right)^2 + \left(\frac{\partial^2 f}{\partial y^2}\right)^2 \mathrm{d}x\mathrm{d}y \tag{7.16}$$

while the former gives a semi-norm based on the Laplacian:

$$\|f\|^2_{\Delta_2} = \int_{\mathbb{R}^2} \left(\frac{\partial^2 f}{\partial x^2}\right)^2 + 2\frac{\partial^2 f}{\partial x^2}\frac{\partial^2 f}{\partial y^2} + \left(\frac{\partial^2 f}{\partial y^2}\right)^2 \mathrm{d}x\mathrm{d}y \tag{7.17}$$

which is *not* strictly equivalent to (7.16). An example is $f = xy$ which gives $\|f\|^2_{\Delta_2} = 0$ but $\|f\|^2_{D_2} = \infty$.

An important case where the expressions (7.17) and (7.16) *are* equivalent is when f is a test function. Then, by integration by parts,

$$\int_{\mathbb{R}^2} \frac{\partial^2 f}{\partial x^2}\frac{\partial^2 f}{\partial y^2}\mathrm{d}x\mathrm{d}y = \int_{\mathbb{R}^2} \left(\frac{\partial^2 f}{\partial x \partial y}\right)^2 \mathrm{d}x\mathrm{d}y \tag{7.18}$$

These may sound like technicalities but they should not be overlooked; otherwise, one may easily formulate problems that are not well defined mathematically (as was for example the case in [134]).

Coming back to the general formulation, we write the Fourier domain equivalent of (7.14) and (7.15):

$$B(\mathbf{f}, \mathbf{g}) = \frac{1}{(2\pi)^m}\langle \hat{\mathsf{L}}\hat{\mathbf{f}}, \hat{\mathsf{L}}\hat{\mathbf{g}}\rangle \tag{7.19}$$

$$= \frac{1}{(2\pi)^m}\int_{\mathbb{R}^m} \left(\hat{\mathsf{L}}\hat{\mathbf{f}}\right)^T(\boldsymbol{\omega}) \left(\hat{\mathsf{L}}\hat{\mathbf{g}}\right)^*(\boldsymbol{\omega})\mathrm{d}\boldsymbol{\omega} \tag{7.20}$$

with an associated criterion

$$J(\mathbf{f}) = \frac{1}{(2\pi)^m}\int_{\mathbb{R}^m} \|\hat{\mathsf{L}}\hat{f}\|^2 \mathrm{d}\boldsymbol{\omega} \tag{7.21}$$

where $\hat{\mathsf{U}} = \hat{\mathsf{L}}^T\hat{\mathsf{L}}^*$. Note that the phase of $\hat{\mathsf{L}}$ can be freely chosen, in addition to the freedom demonstrated in the time-domain factorization. The phase of $\hat{\mathsf{L}}$ may represent the shift of L in the time-domain; more generally, it corresponds to applying an all-pass (unitary-gain) filter to L.

7.5 Imposing invariance properties

The intent of this section is to apply the first principles from Section 7.3.4 to come up with a constrained form of the variational criterion that is consistent with our invariance requirements. We will end up with what is essentially a one-parameter family of criteria (cf. Theorem 3).

As we have seen, sufficient conditions to ensure Properties 3 and 4 are given by (7.4) and (7.6), respectively. We now show how (7.4) and (7.6) constrain our choice of the kernel V of the bilinear form B. It is useful to realize that if

$$\int \mathbf{f}^T(\mathbf{x})\mathsf{V}(\mathbf{x},\mathbf{y})\mathbf{f}(\mathbf{y})\mathrm{d}\mathbf{x}\mathrm{d}\mathbf{y} = \int \mathbf{f}^T(\mathbf{x})\mathsf{W}(\mathbf{x},\mathbf{y})\mathbf{f}(\mathbf{y})\mathrm{d}\mathbf{x}\mathrm{d}\mathbf{y} \quad \text{for all} \quad \mathbf{f} \in \mathscr{D} \tag{7.22}$$

then by considering $\mathbf{f}+\mathbf{g}$ and $\mathbf{f}-\mathbf{g}$ instead of $\mathbf{f}$, we get

$$\int \mathbf{f}^T(\mathbf{x})\mathsf{V}(\mathbf{x},\mathbf{y})\mathbf{g}(\mathbf{y})\mathrm{d}\mathbf{x}\mathrm{d}\mathbf{y} = \int \mathbf{f}^T(\mathbf{x})\mathsf{W}(\mathbf{x},\mathbf{y})\mathbf{g}(\mathbf{y})\mathrm{d}\mathbf{x}\mathrm{d}\mathbf{y} \quad \text{for all} \quad \mathbf{f},\mathbf{g} \in \mathscr{D} \tag{7.23}$$

which is equivalent to saying that $\mathsf{V} = \mathsf{W}$ in the distributional sense. The converse holds too, by substituting $\mathbf{f} = \mathbf{g}$. Therefore, equations (7.4) and (7.6) on the criterion translate into equations for the distributional kernel V as follows

Matrix linearity:
$$\mathsf{A}^T\mathsf{V}(\mathbf{x},\mathbf{y})\mathsf{A} = c(\mathsf{A},\mathbf{b})\mathsf{V}(\mathbf{x},\mathbf{y}) \tag{7.24}$$

and
$$\int \mathsf{V}(\mathbf{x},\mathbf{y})\mathrm{d}\mathbf{y} = 0 \tag{7.25}$$

Geometric invariance:
$$\mathsf{V}(\mathbf{x},\mathbf{y}) = c(\mathsf{A},\mathbf{b})\mathsf{V}(\mathsf{A}\mathbf{x}+\mathbf{b},\mathsf{A}\mathbf{y}+\mathbf{b})(\det\mathsf{A})^2 \tag{7.26}$$

7.5.1 Geometric translation invariance

From (7.25), we directly see that

$$B(\mathbf{f},\mathbf{b}) = 0 \quad \text{for any constant } \mathbf{b} \text{ and } \mathbf{f} \in \mathscr{D} \tag{7.27}$$

and thus there exists an extension of J to functions outside of $\mathscr{D}$ such that $J(\mathbf{b}) = 0$. In other words, the criterion J must give zero for constant functions.

We can now consider geometric translation invariance (in the domain of $\mathbf{x}$), by setting $\mathsf{A} = -\mathsf{I}$ (reflection about the origin) and $\mathbf{b} = \mathbf{x}+\mathbf{y}$ in (7.26), which simplifies to

$$\mathsf{V}(\mathbf{x},\mathbf{y}) = c(-\mathsf{I},\mathbf{x}+\mathbf{y})\mathsf{V}(\mathbf{y},\mathbf{x}) \tag{7.28}$$

Using the symmetry of V, this implies $c(-\mathsf{I}, \mathbf{x}+\mathbf{y}) = 1$ for all $\mathbf{x}$,$\mathbf{y}$. Letting $\mathbf{b} = \mathbf{y}$ in (7.26) leads to

$$\mathsf{V}(\mathbf{x}, \mathbf{y}) = \mathsf{U}(\mathbf{y} - \mathbf{x}) \tag{7.29}$$

where we have substituted $\mathsf{V}(\mathbf{y} - \mathbf{x}, \mathbf{0}) = \mathsf{U}(\mathbf{y} - \mathbf{x})$. Because of the symmetry of V, we get

$$\mathsf{U}(\mathbf{y} - \mathbf{x}) = \mathsf{V}(\mathbf{x}, \mathbf{y}) = \mathsf{V}(\mathbf{y}, \mathbf{x}) = \mathsf{U}(\mathbf{x} - \mathbf{y}) \tag{7.30}$$

which means we can use the simpler expression (7.10) instead of (7.9). Equation (7.26) then becomes

$$\mathsf{U}(\mathbf{x}) = c(\mathsf{A})\mathsf{U}(\mathsf{A}\mathbf{x}) \tag{7.31}$$

for any matrix A.

7.5.2 Rotational invariance

Another special case of geometrical transformations are rotations and symmetries; i.e., matrices which satisfy the orthogonality condition $\mathsf{A}\mathsf{A}^T = \mathsf{I}$. Applying (7.31) twice yields

$$c(\mathsf{A}\mathsf{B}) = c(\mathsf{A})c(\mathsf{B}) \tag{7.32}$$

Consider this equation for a Householder matrix $\mathsf{A} = \mathsf{I} - 2\mathbf{v}\mathbf{v}^T$, where $\mathbf{v}^T\mathbf{v} = 1$. Since $\mathsf{A}^2 = \mathsf{I}$, from (7.32) we have $|c(\mathsf{A})| = 1$. Furthermore, as $J \geq 0$, we necessarily have $c(\mathsf{A}) \geq 0$ and thus $c(\mathsf{A}) = 1$.

The equation (7.31) becomes $\mathsf{U}(\mathbf{x}) = \mathsf{U}(\mathsf{A}\mathbf{x})$. It is always possible to choose A such that $\mathsf{A}\mathbf{x} = \|\mathbf{x}\|\mathbf{e}_1$, where $\mathbf{e}_1$ is the first basis vector, see [141]. Consequently, the distribution U must be radial:

$$\mathsf{U}(\mathbf{x}) = \mathsf{U}_0(r) \quad \text{where} \quad r = \|\mathbf{x}\| \tag{7.33}$$

It is easy to verify that thanks to the orthonormality of A, rotating $\mathbf{x} \rightarrow \mathsf{A}\mathbf{x}$ does not change r.

7.5.3 Scale invariance

The last remaining class of geometrical transformations we consider is uniform scaling. Using expression (7.31) as before yields: $\mathsf{U}_0(\mathbf{x}) = c(\lambda)\mathsf{U}_0(\lambda\mathbf{x})$, where λ is a real scaling factor, and where we have accommodated the Jacobian λ^{2m} into $c(\lambda)$. We use the rotation invariant form (7.33) which gives $\mathsf{U}_0(r) = c(\lambda)\mathsf{U}_0(\lambda r)$ for $\lambda > 0$. Note that $c(\lambda\lambda') = c(\lambda)c(\lambda')$. Repetitive scaling by λ yields $c(\lambda)^k = c(\lambda^k)$. This implies $c(\lambda)^{1/q} =$

$c(\lambda^{1/q})$ and $c(\lambda)^{p/q} = c(\lambda^{p/q})$. By continuity also $c(\lambda)^x = c(\lambda^x)$, for real x. Consequently we have $c(\lambda) = \lambda^\gamma$ and

$$\mathsf{U}_0(r) = \lambda^\gamma \mathsf{U}_0(\lambda r) \tag{7.34}$$

In the case where the radial form of the convolutional kernel $\mathsf{U}_0(r)$ is a function, the preceding equation implies $\mathsf{U}_0(r) = cr^{-\gamma}$. Note that when $\gamma \leq 2m$, then $\mathsf{U}(\mathbf{x})$ is not locally integrable over $\mathbb{R}^m \times \mathbb{R}^m$. Therefore, we need to consider the integral in the sense of distributions.

The corresponding expression in the Fourier domain is

$$\hat{\mathsf{U}}(\boldsymbol{\omega}) = c\|\boldsymbol{\omega}\|^{2\alpha} \tag{7.35}$$

where $2\alpha = \gamma - m$ and the factor 2 is for future convenience and notational consistency with [45].

7.5.4 Matrix linearity

We have already studied the effect of $\mathbf{b}$ in (7.24). Let us now concentrate on the implications of A. Substituting (7.30) yields: $\mathsf{A}^T\mathsf{U}(\mathbf{x})\mathsf{A} = c(\mathsf{A})\mathsf{U}(\mathbf{x})$, thus $c(\mathsf{AB}) = c(\mathsf{A})c(\mathsf{B})$. We show that $c(\mathsf{A}) = 1$ by the same proof as in Section 7.5.2. Thus $\mathsf{U}(\mathbf{x})$ commutes with an arbitrary orthogonal matrix A:

$$\mathsf{UA} = \mathsf{AU} \tag{7.36}$$

Similarly to Section 7.5.2, we will consider only Householder matrices $\mathsf{A} = \mathsf{I} - 2\mathbf{v}\mathbf{v}^T$. Substituting into (7.36) yields $\mathbf{v}\mathbf{v}^T\mathsf{U} = \mathsf{U}\mathbf{v}\mathbf{v}^T$. Right-multiplying by $\mathbf{v}$ shows that $\mathbf{v}(\mathbf{v}^T\mathsf{U}\mathbf{v}) = \mathsf{U}\mathbf{v}$, which means that any vector $\mathbf{v}$ is an eigenvector of U. This implies that U is a multiply of an identity matrix.

$$\mathsf{U}(\mathbf{x}) = \mathbf{u}(\mathbf{x}) \cdot \mathsf{I} \tag{7.37}$$

7.5.5 Form of the criterion

A direct consequence of the results from the preceding sections is the following theorem:

Theorem 3 (Form of the criterion) *Let $J(f)$ be a quadratic criterion. Then any associated variational problem $\mathcal{P}$ satisfies Property 2. Furthermore, $\mathcal{P}$ satisfies Properties 3 and 4 if and only if J can be expressed in the following form:*

$$J(f) = c \int_{\mathbb{R}^m} \sum_{i=1}^{n} \underbrace{\|\boldsymbol{\omega}\|^{2\alpha}}_{(\hat{\mathsf{U}})_{ii}} |\hat{f}_i(\boldsymbol{\omega})|^2 \, \mathrm{d}\boldsymbol{\omega} \tag{7.38}$$

for any tempered distribution f.

The bilinear form associated to (7.38) is

$$B(f,g) = c \int_{\mathbb{R}^m} \sum_{i=1}^{n} \|\boldsymbol{\omega}\|^{2\alpha} \hat{f}_i(\boldsymbol{\omega})\, \hat{g}_i^*(\boldsymbol{\omega})\, \mathrm{d}\boldsymbol{\omega} \tag{7.39}$$

7.5.6 Laplacian semi-norm

The criterion defined in the Fourier domain by (7.38) is easily associated to an equivalent semi-norm in the time domain using an iterated m-dimensional Laplacian for even α

$$J_{\Delta_\alpha}(f) = \int_{\mathbb{R}^m} \left| \Delta^{\alpha/2} f(\mathbf{x}) \right|^2 \mathrm{d}\mathbf{x} \tag{7.40}$$

7.5.7 Duchon's semi-norms

The principal disadvantage of (7.40) is that its kernel $\mathcal{K}_\Delta$ is too large. For example for $\alpha = 2$, it contains every function that satisfies Laplace equation, such as the real part of an analytical function, for example $(x+iy)^k + (x-iy)^k$. Therefore, the variational problem with this criterion will typically have an infinite number of solutions.

Fortunately, it turns out that there are other time-domain forms which correspond to (7.38) and which do not have this problem. Namely, we now present the family of semi-norms introduced by Duchon [45]. He first defines a differential operator D^M as a vector of all possible partial derivatives of f of order M:

$$D^M f = \left[\frac{\partial^M f}{\partial x_1^M}, \ldots, \frac{\partial^M f}{\partial x_{k_1} \ldots \partial x_{k_M}}, \ldots, \frac{\partial^M f}{\partial x_m^M} \right] \tag{7.41}$$

with $k_1, \ldots, k_M \in \{1, \ldots, m\}^M$. For example, for $m = 2$, $M = 2$ we get

$$D^2 f(x,y) = \left[\frac{\partial^2 f}{\partial x^2}, \frac{\partial^2 f}{\partial x \partial y}, \frac{\partial^2 f}{\partial y \partial x}, \frac{\partial^2 f}{\partial y^2} \right] \tag{7.42}$$

Then he defines a semi-norm by taking the sum of the squares of all the elements and integrate it over the space $\mathbb{R}^m$

$$\|f\|_{D_M} = \left(\int_{\mathbb{R}^m} \|D^M f\|^2 \mathrm{d}\mathbf{x} \right)^{1/2} \tag{7.43}$$

where $\|\cdot\|$ is an euclidean norm in $\mathbb{R}^{m^M}$. More explicitely (using the commutativity of the partial derivatives)

$$\|f\|_{D_M}^2 = \sum_{|\mathbf{l}|=M} \frac{M!}{\mathbf{l}!} \int_{\mathbb{R}^m} \left(\frac{\partial^M f}{\partial \mathbf{x}^{\mathbf{l}}} \right)^2 \mathrm{d}\mathbf{x} \tag{7.44}$$

where $l_1, \ldots, l_m \in \{0, \ldots, M\}$, $\mathbf{l}! = \prod_{s=1}^{m} l_s!$, $|\mathbf{l}| = \sum_{s=1}^{m} l_s$ and $\partial \mathbf{x}^{\mathbf{l}} = \partial x_1^{l_1} \ldots \partial x_m^{l_m}$. Following our example for $m = 2$ and $M = 2$, we get the most often used Duchon's semi-norm (7.16). This semi-norm leads to the well known thin-plate splines [96].

Interestingly, the kernel of $\|\cdot\|_{D_2}$ contains only functions whose second partial derivatives are zero; i.e., linear polynomials $a_0 + a_1 x + a_2 y$. Generally, the kernel $\mathcal{K}_{D_M}$ of Duchon's semi-norm of order M contains polynomials of degree $M - 1$.

7.5.8 Semi-norms for fractional derivatives

Duchon goes one step further in combining the time and Fourier domain definitions to obtain also semi-norms corresponding to fractional derivatives.

$$J(f)_{D_{M,s}} = \int_{\mathbb{R}^m} \|\boldsymbol{\omega}\|^{2s} \left\| \mathscr{F} D^M f \right\|_{l_2}^2 \mathrm{d}\boldsymbol{\omega} \tag{7.45}$$

where $\mathscr{F}$ is the Fourier transform operator as defined in Section 7.4.4. When $s = 0$, this definition is completely equivalent to (7.43), that is, $\|f\|_{D_M}^2 = J(f)_{D_{M,0}}$. When, on the other hand, $M = 0$, this definition is equivalent to (7.38); i.e., $J(f)_{\Delta_\alpha} = \|f\|_{D_{0,\alpha}}$. For $s < 1$, the kernel of $J(f)_{D_{M,s}}$ is the kernel of $\|f\|_{D_M}$.

7.5.9 Corresponding bilinear forms

All the Duchon's semi-norms presented above can be associated with a bilinear form so that $\|f\|^2 = J(f) = B(f, f)$. The basic form (7.43) gives

$$B_M(f, g) = \int_{\mathbb{R}^m} (D^M f) \cdot (D^M g) \mathrm{d}\mathbf{x} \tag{7.46}$$

or equivalently

$$B_M(f, g) = \int_{\mathbb{R}^m} \sum_{\sum k_l = M} \frac{M!}{\mathbf{k}!} \frac{\partial^M f}{\partial \mathbf{x}^{\mathbf{k}}} \frac{\partial^M g}{\partial \mathbf{x}^{\mathbf{k}}} \mathrm{d}\mathbf{x} \tag{7.47}$$

The general form (7.45) leads to

$$B_{M,s}(f, g) = \int_{\mathbb{R}^m} \|\boldsymbol{\omega}\|^{2s} \left\| \mathscr{F} D^M f(\mathbf{x}) \right\| \left\| \mathscr{F} D^M g(\mathbf{x}) \right\| \mathrm{d}\boldsymbol{\omega} \tag{7.48}$$

7.6 Solution of the variational problem

In this section, we reconsider our variational problem $\mathcal{P}(J, \mathsf{H}, \mathbf{s})$ defined in Section 7.3.3, derive some properties of its solution $\mathbf{f}_{\text{out}}$ and use them to obtain the explicit form of the solution.

7.6.1 Lagrange multipliers

First, we construct an augmented criterion according to the Lagrange multipliers' method.

$$J_*(\mathbf{f}, \boldsymbol{\lambda}) = J(\mathbf{f}) - 2\boldsymbol{\lambda}^T \left(\langle \mathsf{H}, \mathbf{f} \rangle - \mathbf{s}\right) \tag{7.49}$$

where $\boldsymbol{\lambda} \in \mathbb{R}^N$ is the vector of Lagrange multipliers. If $\mathbf{f}_\lambda$ minimizes $J_*(\mathbf{f}, \boldsymbol{\lambda})$ then choosing $\boldsymbol{\lambda}$ such that $\langle \mathsf{H}, \mathbf{f} \rangle = \mathbf{s}$ implies that $\mathbf{f}_{\text{out}} = \mathbf{f}_{\lambda_{\text{opt}}}$ minimizes $J(\mathbf{f})$ under constraints (7.2).

We carry on using a standard variational argument. We take a small perturbation $\alpha\mathbf{g}$, where $\mathbf{g} \in F$ and $\alpha \in \mathbb{R}$, add it to $\mathbf{f}_{\text{out}}$ and study the change

$$\xi(\alpha) = J_*(\mathbf{f}_{\text{out}} + \alpha\mathbf{g}) - J_*(\mathbf{f}_{\text{out}}) \tag{7.50}$$

it induces in the criterion. We consider its derivative

$$\frac{\partial}{\partial \alpha}\xi = 2B(\mathbf{f}_{\text{out}}, \mathbf{g}) - 2\boldsymbol{\lambda}^T \langle \mathsf{H}, \mathbf{g} \rangle \tag{7.51}$$

Iff $(\mathbf{f}_{\text{out}}, \boldsymbol{\lambda})$ is a saddle point, then $\frac{\partial}{\partial \alpha}\xi = 0$, and $\langle \mathsf{H}, \mathbf{f}_{\text{out}} \rangle = \mathbf{s}$. (See Appendix C for a proof.) This directly leads to the following lemma:

Lemma 1 *A function $\mathbf{f}_{out}$ from F satisfying $\langle \mathsf{H}, \mathbf{f}_{out} \rangle = \mathbf{s}$ solves the variational problem $\mathcal{P}$, if and only if there is a real vector $\boldsymbol{\lambda}$ such that for all $\mathbf{g} \in F$*

$$B(\mathbf{f}_{out}, \mathbf{g}) = \boldsymbol{\lambda}^T \langle \mathsf{H}, \mathbf{g} \rangle \tag{7.52}$$

Note that because of the Property 5 on density, we can initially consider only $\mathbf{g}$ from $\mathscr{D}$ and then extend to F, while the Lemma remains valid.

For $\mathbf{g}$ from the kernel $\mathcal{K}$, we have $B(\mathbf{f}_{\text{out}}, \mathbf{g}) = 0$ (because $B(\mathbf{g}, \mathbf{g}) = 0$) and thus

$$\boldsymbol{\lambda}^T \langle \mathsf{H}, \mathbf{g} \rangle = 0 \quad \text{for each} \quad \mathbf{g} \in \mathcal{K} \tag{7.53}$$

7.6.2 Introducing fundamental solutions

We now suppose that we have found a set of functions $\boldsymbol{\varphi}_i$ such that

$$B(\boldsymbol{\varphi}_i, \mathbf{g}) = \langle \mathbf{h}_i, \mathbf{g} \rangle \quad \text{for all} \quad \mathbf{g} \in \mathscr{D} \tag{7.54}$$

We call $\boldsymbol{\varphi}_i$ a *fundamental solution* corresponding to a filter $\mathbf{h}_i$. (See also Section 7.6.5.) Often there is no fundamental solution $\boldsymbol{\varphi}_i$ in F. Then we search $\boldsymbol{\varphi}_i$ in $\overline{F}$, which is why we had to restrict $\mathbf{g}$ to $\mathscr{D}$ (Section 7.4.5).

We want $\mathsf{H}^{RT} * \boldsymbol{\varphi}_i$ to be finite for all $i = 1, \ldots N$. If this is not the case, we can suspect that our minimization problem does not have a solution in F, which can hopefully be proven using another method.

7.6.3 Explicit solution of the variational problem

In order to obtain a more useful result than Lemma 1, we will use the linearity of B. Take a function

$$\mathbf{f}(\mathbf{x}) = \sum_{i=1}^{N} \lambda_i \boldsymbol{\varphi}_i \tag{7.55}$$

Because of (7.54), the function (7.55) clearly satisfies (7.52). We might be tempted to conclude that, if $\mathbf{f}$ is consistent with constraints (7.2), then it solves $\mathcal{P}$. However, this will not necesarrily work because $\boldsymbol{\varphi}_i$, and therefore $\mathbf{f}$ in (7.55), does not in general belong to the admissible solution space F.

With (7.55), we have exactly as many λ_i's as there are consistency constraints (7.2). This means that there are not enough degrees of freedom in (7.55) to ensure the condition $\mathbf{f} \in F$.

Note that if $\mathbf{p}$ belongs to $\mathcal{K}$ then $B(\mathbf{f}+\mathbf{p}, \mathbf{g}) = B(\mathbf{f}, \mathbf{g})$. We can therefore add to $\mathbf{f}$ a function $\mathbf{p}$ from $\mathcal{K}$, obtaining $\mathbf{f}_{\text{out}} = \mathbf{f} + \mathbf{p}$, which gives us the possibility to make $\mathbf{f}_{\text{out}} \in F$, while conserving the validity of (7.52).

Equation (7.53) will allow us to find the $\mathbf{p}$. If $\mathcal{K}$ has a finite basis, we can express $\mathbf{p}(\mathbf{x})$ as

$$\mathbf{p}(\mathbf{x}) = \sum_{k=0}^{P-1} a_k \mathbf{p}_k(\mathbf{x}) \tag{7.56}$$

Through linearity, (7.53) is equivalent to the orthogonality constraint

$$\boldsymbol{\lambda}^T \langle \mathsf{H}, \mathbf{p}_k \rangle = 0 \quad \text{for each} \quad k \tag{7.57}$$

which, together with the consistency constraints (7.2), gives the same number of constraints as there are additional unknowns in (7.56). Combining (7.57) and (7.2) gives us a set of linear equations for exactly as many unknowns λ_i and a_k, which is a necessary condition for the unicity of the solution. (More on unicity in Section 7.6.7.) Adding the kernel term is in general sufficient to ensure that $\mathbf{f}_{\text{out}} \in F$. We summarize our findings in the form of a theorem. See Appendix C for a proof.

Theorem 4 (Variational problem solution) *Let λ_i and a_k be real numbers and $\{\mathbf{p}_k\}$ a basis of the kernel $\mathcal{K}$ of J. Let further $\{\boldsymbol{\varphi}_i\}$ be a set of fundamental solutions corresponding to filters H, in the sense of* (7.54)*. Then the function*

$$\mathbf{f}_{out}(\mathbf{x}) = \sum_{k=0}^{P-1} a_k \mathbf{p}_k + \sum_{i=1}^{N} \lambda_i \boldsymbol{\varphi}_i \tag{7.58}$$

solves the interpolation problem $\mathcal{P}(J, \mathsf{H}, \mathbf{s})$ (where $J(\mathbf{f}) = B(\mathbf{f}, \mathbf{f})$) if and only if the following three conditions are satisfied:

(i) *The solution* $\mathbf{f}_{out}$ *belongs to* F *as defined by* (7.3)*, i.e.,* $J(\mathbf{f}_{out}) < \infty$.

(ii) *The solution* $\mathbf{f}_{out}$ *is consistent with the constraints* (7.2)*,i.e.,* $\langle \mathsf{H}, \mathbf{f}_{out} \rangle = \mathbf{s}$.

(iii) *The coefficients* λ_i *are orthogonal in the sense of* (7.57)*, i.e.,* $\boldsymbol{\lambda}^T \langle \mathsf{H}, \mathbf{p}_k \rangle = 0, \quad \forall\, k$.

Symbolically, we can combine (7.52) and (7.2) by substituting $\mathbf{g} = \mathbf{f}_{\text{out}}$, yielding a very simple expression for the optimal value of the criterion $J(\mathbf{f}_{\text{out}})$

$$J(\mathbf{f}_{\text{out}}) = \boldsymbol{\lambda}^T \mathbf{s} \tag{7.59}$$

where $\mathbf{s}$ is the measurement vector.

7.6.4 Linear equation set

The solution $\mathbf{f}_{\text{out}}(\mathbf{x})$ is given by equation (7.58) which contains $P+N$ unknown parameters. These parameters are determined from a linear system of equations

$$\underbrace{\begin{bmatrix} \mathsf{A} & \mathsf{Q}_1 \\ \mathsf{Q}_2 & \mathbf{0} \end{bmatrix}}_{\mathsf{B}} \begin{bmatrix} \boldsymbol{\lambda} \\ \mathbf{a} \end{bmatrix} = \begin{bmatrix} \mathbf{s} \\ \mathbf{0} \end{bmatrix} \tag{7.60}$$

The sub-matrix $[\mathsf{A}\ \mathsf{Q}_1]$ corresponds to the constraints (7.2). The sub-matrix A of size $N \times N$ represents the contribution of the fundamental solutions and its elements are $(\mathsf{A})_{i,j} = \langle \mathbf{h}_i, \varphi_j \rangle$. In many cases $\langle \mathbf{h}_i, \varphi_j \rangle = \langle \mathbf{h}_j, \varphi_i \rangle$ and the matrix A is symmetrical.

The sub-matrix Q_1 of size $N \times P$ represents the contribution of the kernel element $\mathbf{p}$ in (7.58). Its elements are thus $(\mathsf{Q}_1)_{i,k} = \langle \mathbf{h}_i, \mathbf{p}_k \rangle$. The third sub-matrix Q_2 is of size $P \times N$; it represents the orthogonality conditions (7.57). Its elements are $(\mathsf{Q}_1)_{k,i} = \langle \mathbf{h}_i, \mathbf{p}_k \rangle$. We can see that $\mathsf{Q}_1 = \mathsf{Q}_2^T \stackrel{\text{def}}{=} \mathsf{Q}$. Consequently, if A is symmetrical, then B is symmetrical as well.

7.6.5 Finding the fundamental solutions

To find the fundamental solutions $\boldsymbol{\varphi}$ as defined by (7.54), it is useful to start from the convolutional formulation of the bilinear form (7.13). Similarly, we use the convolutional form (7.2) to describe the sampling. Thus, the (7.54) becomes

$$\langle \mathsf{U} * \boldsymbol{\varphi}_i, \mathbf{g} \rangle = \langle \mathbf{h}_i, \mathbf{g} \rangle \tag{7.61}$$

The fundamental solutions are then defined through the distributional equations

$$\mathsf{U} * \underbrace{[\boldsymbol{\varphi}_1 \dots \boldsymbol{\varphi}_q]}_{\boldsymbol{\Phi}} = \mathsf{H} \tag{7.62}$$

The task can be broken in two parts. We first solve for the Green functions $\boldsymbol{\psi}_i$

$$\mathsf{U} * \underbrace{[\boldsymbol{\psi}_1 \dots \boldsymbol{\psi}_n]}_{\boldsymbol{\Psi}} = \delta(\mathbf{x})\, \mathbf{I}_{n\times n}, \quad \mathbf{x} \in \mathbb{R}^m \tag{7.63}$$

Once we have the Green functions $\boldsymbol{\psi}_i$, we get the fundamental solutions $\boldsymbol{\varphi}_i$ by convolution with the measurement operators H:

$$[\boldsymbol{\varphi}_1 \dots \boldsymbol{\varphi}_q] = \boldsymbol{\Phi} = \boldsymbol{\Psi} * \mathsf{H} \tag{7.64}$$

We see from (7.63) that when U is symmetrical, then $\boldsymbol{\Psi}$ is symmetrical too. When further H is symmetrical, then the same holds true for $\boldsymbol{\Phi}$ (from (7.64)). This is often the case, as U and H are mostly diagonal. Consequently, this causes the matrices A and B to be symmetrical as well. (See Section 7.6.4.)

7.6.6 Green functions

As an example, let us first study a simple scalar case ($n = 1$, $m = 1$). As criterion, we choose Duchon's semi-norm $\|f\|_{D_2} = \|f''\|_{L_2}$ which corresponds to $l = \delta''$ in (7.14) and thus $u = l^R * l = \frac{\mathrm{d}^4}{\mathrm{d}x^4}\delta$. The corresponding Green function must satisfy

$$u * \psi = \frac{\mathrm{d}^4\delta}{\mathrm{d}x^4} * \psi = \frac{\mathrm{d}^4\psi}{\mathrm{d}x^4} = \delta \tag{7.65}$$

Integrating four times, we get a family of possible Green functions $\psi(x) = x_+^3/12 + a_3x^3 + a_2x^2 + a_1x + a_0$, where x_+^3 is the one-sided power function. For convenience, we choose the symmetrical one $\psi(x) = |x|^3/12$.

The Green functions corresponding to general Duchon's semi-norms (7.45) are best analyzed in the Fourier domain using (7.38) with $\alpha = M + s$. Then the following must hold:

$$\hat{u}\,\hat{\psi} = \|\boldsymbol{\omega}\|^{2\alpha}\,\hat{\psi} = 1 \tag{7.66}$$

also in the distributional sense. Because both u and δ are radial[2] distributions, and a convolution of two radial distributions is also radial, ψ is radial as well. That is why the resulting functions ψ (and φ, if h preserves radiality) are called *radial basis functions*. We express $\psi(\mathbf{x})$ as $\psi(r)$, with $r = \|\mathbf{x}\|$. Consequently $\hat{\psi}(\boldsymbol{\omega})$ is radial too and can be expressed as $\hat{\psi}(\|\boldsymbol{\omega}\|)$.

The problem of finding $\hat{\psi}$ from (7.66) is well studied (cf. [139], page 258). In the Fourier domain, we get, for $2\alpha - m$ not an even integer:

$$\mathscr{F}^{-1}\mathrm{Pf}\,\|\omega\|^{-2\alpha} = c\,r^{2\alpha-m} \tag{7.67}$$

[2]We call a distribution u radial, if $\langle u, v(\mathbf{x})\rangle = \langle u, v(\Omega\mathbf{x})\rangle$ holds for any test function v and rotation Ω about the origin.

where c is a constant which can be calculated but which is irrelevant for our purposes. The 'Pf' ('partie finie' [139]—finite part) symbol means that we are considering a distribution that coincides with the function $\|\omega\|^{-2\alpha}$ for $\omega \neq \mathbf{0}$, which does not hinder the validity of the equation (7.66) in the distributional sense.

If $2\alpha - m$ is an even positive integer, the above formula has to be modified as

$$\mathscr{F}^{-1}\mathrm{Pf}\|\omega\|^{-2\alpha} = c_0 r^{2\alpha-m}\log r + c_1 r^{2\alpha-m} \tag{7.68}$$

For our task, we do not have to consider the $c_1 r^{2\alpha-m}$ part of (7.68) because it belongs to $\mathcal{K}$.

For $2\alpha - m$ even, it is actually easier to work directly in the time domain. If we have a radial function $g(r)$ that satisfies $g' = r^{1-m}$, then $\Delta g = S_m \delta$. (See Appendix D for a proof.) The constant S_m is the surface of the m-dimensional unit hypersphere.[3] For example, for $m = 2$ we get $\Delta \log r = 2\pi\delta$. Iteratively applying the formula for the Laplacian of a radial function

$$\Delta\psi(\mathbf{x}) = \psi''(r) + \frac{m-1}{r}\psi'(r) \tag{7.69}$$

yields $\Delta^2 r^2 \log r = \Delta(4\log r + 4) = 8\pi\delta$, $\Delta^3 r^4 \log r = 64\pi\delta$, and $\Delta^4 r^6 \log r = 2304\pi\delta$. For $m = 3$, we have $\Delta r^{-1} = 4\pi\delta$, $\Delta^2 r = 8\pi\delta$, $\Delta^3 r^3 = 96\pi\delta$, etc.

Generally, Duchon's semi-norm $\|f\|_{D_{M,s}}$ leads to a fundamental solution $\varphi(r) = cr^{2(M+s)-m}$, if the exponent is not even, or $\varphi(r) = cr^{2(M+s)-m}\log r$ otherwise. This permits us to choose from the continuum of Duchons' semi-norms the one that suits us best.

In the multidimensional ($n > 1$) case, where $\mathbf{U} = u\mathbf{I}$, we get simply $\mathbf{\Psi} = \psi\mathbf{I}$.

7.6.7 Unicity of the solution

Let us suppose that the set of fundamental solutions $\{\boldsymbol{\varphi}_i\}$ and a finite basis $\{\mathbf{p}_k\}$ exist. Then there is a set of linear equations (7.60) to determine the unknowns λ_i and a_k. If this set has a unique solution, the interpolation problem will also have a unique solution, provided, of course, that (7.57) implies $\mathbf{f} \in F$.

In the single channel, ideal sampling ($\langle h_i, f\rangle = f(\mathbf{x}_i)$), unidimensional case, the set has the form (7.60) with $(\mathsf{A})_{ij} = \varphi(\mathbf{x}_i - \mathbf{x}_j)$ and $(\mathsf{Q}_1)_{ik} = (\mathsf{Q}_2^T)_{ik} = \mathbf{p}_k(\mathbf{x}_i)$. If φ is radial, it can be written as $\varphi(\mathbf{x}) = \varphi(r) = \phi(r^2)$. Micchelli [128] proved that A is nonsingular provided that ϕ' is completely monotonic but not constant on $(0,\infty)$, ϕ is continuous on $[0,\infty)$ and positive on $(0,\infty)$, and $\mathbf{x}_j$ are distinct. (A function ϕ is completely monotonic provided it is in C^∞ and $(-1)^l\phi^{(l)} \geq 0$ for $l = 0, 1, \ldots$.)

Powell [72] has additionally shown that if $\mathcal{K}$ is the space of polynomials of order $M-1$ and if either $\phi^{(M)}$ or $-\phi^{(M)}$ is strictly completely monotonic on $(0,\infty)$, then B is

[3] $S_m = 2\pi^{m/2}/\Gamma(m/2)$ which for $m = 2, m = 3$ yields the familiar values 2π and 4π [127].

nonsingular provided that the $\mathbf{x}_j$ are distinct and that there is no non-zero polynomial Q of order $M-1$ such that $Q(\mathbf{x}_j)=0$ for all j. This is closely related to our observation in Section 7.4.1.

The radial functions $\varphi(r)=r^\beta$ give completely monotonic $\phi^{(M)}$ which guarantees the regularity of A, as do the most often used $r^\beta \log r$, stemming from Duchon semi-norms with $\beta=2(M+s)-m$. (The semi-norm $||f||_{D_{M,s}}$ leads to $\phi(r)=cr^{M+s-m/2}$, eventually $\phi(r)=cr^{M+s-m/2}\log r$. The kernel $\mathcal{K}$ contains polynomials up to order $M-1$, so we need to calculate $\phi^{(M)}$.)

7.7 Approximation problem

In some applications it might be interesting to replace the 'hard' constraints (7.2) by 'soft' ones, by adding a data term penalizing solution far from the constraints. To define a variational *approximation problem* we introduce a combined criterion J_a. We consider the following general form:

$$J_a(\mathbf{f}) = J(\mathbf{f}) + \mathscr{L}\big(\langle \mathsf{H}, \mathbf{f}\rangle, \mathbf{s}\big) \tag{7.70}$$

where $\mathscr{L} : \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}$ is an arbitrary distance function. We use it to measure the distance between the measurements and the sampled solution $\langle \mathsf{H}, \mathbf{f}\rangle$. We then say that $\mathbf{f}_{\text{out}} \in F$ is a solution to an approximation problem $\mathcal{A}(J_a, \mathsf{H}, \mathbf{s})$, iff for all functions $\mathbf{f} \in F$ we have $J_a(\mathbf{f}) \geq J_a(\mathbf{f}_{\text{out}})$.

The problem of solving the approximation problem $\mathcal{A}$ is closely related to the constrained problem $\mathcal{P}$, as demonstrated by the following Theorem.

Theorem 5 (AP solution) *Let us denote $J_{min}(\mathbf{z})$ the criterion value $J(\mathbf{f})$ of a function $\mathbf{f}$ solving a constrained problem $\mathcal{P}(J, \mathsf{H}, \mathbf{z})$. Let us further define $\mathbf{z}_*$ as*

$$\mathbf{z}_* = \arg\min_{\mathbf{z}} \big(J_{min}(\mathbf{z}) + \mathscr{L}(\mathbf{z}, \mathbf{s})\big) \tag{7.71}$$

Then $\mathbf{f}_{out}$ solves the problem $\mathcal{A}(J_a, \mathsf{H}, \mathbf{s})$, iff it solves the interpolation problem $\mathcal{P}(J, \mathsf{H}, \mathbf{z}_*)$

The proof follows from the observation that the data term in J_a depends only on the measurements $z_i = \langle \mathbf{h}_i, \mathbf{f}\rangle$ of the solution $\mathbf{f}$. Thus, the minimization

$$\mathbf{f}_{\text{out}} = \arg\min_{\mathbf{f}\in F} J_a(\mathbf{f}) \tag{7.72}$$

required to solve $\mathcal{A}$ can be broken into two parts: (a) external minimization with respect to the $\mathbf{z}$, and (b) internal minimization trying to find the proper $\mathbf{f}_{\text{out}}$ minimizing J given $\mathbf{z}$. We see that the internal optimization is exactly the constrained variational problem described previously. Once it is solved, the external minimization becomes a standard multidimensional optimization problem which can be solved by existing numerical methods [50], or in some special cases analytically (see next section for an example).

7.7.1 Least-squares approximation

Often, the general criterion (7.70) can be replaced by a simple least-squares form

$$J_a(\mathbf{f}) = J(\mathbf{f}) + \gamma \sum_i \left(\langle \mathbf{h}_i, \mathbf{f} \rangle - s_i\right)^2 \tag{7.73}$$

We first realize that according to Theorem 5, the solution has the form (7.58). We then use the method of small perturbations by evaluating the change (similarly to (7.50))

$$\xi(\alpha) = J_a(\mathbf{f}_{\text{out}} + \alpha \mathbf{g}) - J_a(\mathbf{f}_{\text{out}}) \tag{7.74}$$

Its derivative needs to be zero for all $\mathbf{g} \in F$, in order for $\mathbf{f}_{\text{out}}$ to be a minimum,i.e., $\frac{\partial \xi}{\partial \alpha} = 0$. This implies

$$\frac{\partial}{\partial \alpha}\xi = 2B(\mathbf{f}_{\text{out}}, \mathbf{g}) + 2\gamma \sum_i \langle \mathbf{h}_i, \mathbf{g} \rangle \left(\langle \mathbf{h}_i, \mathbf{f}_{\text{out}} \rangle - s_i\right) = 0 \tag{7.75}$$

Substituting (7.52) into $B(\mathbf{f}_{\text{out}}, \mathbf{g})$ in (7.75) yields

$$-\sum_i \lambda_i \langle \mathbf{h}_i, \mathbf{g} \rangle = \gamma \sum_{i=1}^{N} \langle \mathbf{h}_i, \mathbf{g} \rangle \left(\langle \mathbf{h}_i, \mathbf{f}_{\text{out}} \rangle - s_i\right) \tag{7.76}$$

As this equality needs to hold for all $\mathbf{g}$, we can factor out the term dependent on $\mathbf{g}$. Substituting in the solution (7.58) yields

$$\gamma^{-1}\lambda_i = s_i - \sum_{j=1}^{N} \lambda_j \langle \mathbf{h}_i, \boldsymbol{\varphi}_j \rangle - \sum_{k=0}^{P-1} a_k \langle \mathbf{h}_i, \mathbf{p}_k \rangle \tag{7.77}$$

By taking $\mathbf{g} \in \mathcal{K}$ in (7.75) we get the same orthogonality constraints (7.57) as in the interpolation case. Combining (7.57) with the preceding equation leads to a linear equation set

$$\underbrace{\begin{bmatrix} \mathbf{A} + \gamma^{-1}\mathbf{I} & \mathrm{Q} \\ \mathrm{Q}^T & \mathbf{0} \end{bmatrix}}_{\mathrm{B}} \begin{bmatrix} \boldsymbol{\lambda} \\ \mathbf{a} \end{bmatrix} = \begin{bmatrix} \mathbf{s} \\ \mathbf{0} \end{bmatrix} \tag{7.78}$$

Note the fundamental similarity to (7.60) and the simple form of the regularization by adding a diagonal matrix.

7.8 Conclusions

We have presented a systematic way of solving variational problems minimizing quadratic regularization criteria under general linear constraints. We have also considered replacing

the constraints by a corresponding penalty function and we show that it leads to a solution with the same form. The solution of such problems lies in a vector space uniquely corresponding to the problem at hand, generated by a system of fundamental solutions, related to Green functions. We have shown how the requirements we impose on the variational problem solution determine the choice of the criterion, leading to the family of semi-norms introduced by Duchon.

7.9 Appendix

A Linearity with respect to measurements

Let us have a function $\mathbf{v}$ from F_0 where $F_0 = \{\mathbf{v} \in F; \forall\, i;\ \langle \mathbf{h}_i, \mathbf{v}\rangle = 0\}$. By linearity, $\gamma\mathbf{v}$ also belongs to F_0 for $\gamma \in \mathbb{R}$. We then have $J(\mathbf{f}+\gamma\mathbf{v}) \geq J(\mathbf{f})$ because $\mathbf{f}$ solves the GIP. Consequently $2\gamma B(\mathbf{f},\mathbf{v}) + \gamma^2 B(\mathbf{v},\mathbf{v}) \geq 0$ and thus $\gamma B(\mathbf{f},\mathbf{v}) \geq 0$ for sufficiently small (positive or negative) γ which implies

$$B(\mathbf{f},\mathbf{v}) = 0 \quad \text{for any } \mathbf{v} \text{ from } F_0 \tag{7.79}$$

This leads to $J(\alpha\mathbf{f}+\beta\mathbf{g}+\mathbf{v}) - J(\alpha\mathbf{f}+\beta\mathbf{g}) = 2B(\alpha\mathbf{f}+\beta\mathbf{g},\mathbf{v}) + J(\mathbf{v}) = J(\mathbf{v}) \geq 0$, which proves that $\alpha\mathbf{f}+\beta\mathbf{g}$ solves the problem with measurements $\alpha\mathbf{r}+\beta\mathbf{s}$, when $\mathbf{f}$ and $\mathbf{g}$ solve problems with measurements $\mathbf{r}$ and $\mathbf{s}$, respectively.

B Difference between two solutions

We prove that if two functions $\mathbf{f}_1$ and $\mathbf{f}_2$ both minimize $J(\mathbf{f})$ under some constraints (7.2), then $J(\mathbf{f}_1 - \mathbf{f}_2) = 0$. Using (7.79) we deduce $B(\mathbf{f}_1, \mathbf{f}_1 - \mathbf{f}_2) = B(\mathbf{f}_2, \mathbf{f}_1 - \mathbf{f}_2) = 0$. This directly yields $J(\mathbf{f}_1 - \mathbf{f}_2) = B(\mathbf{f}_1, \mathbf{f}_1 - \mathbf{f}_2) + B(\mathbf{f}_2, \mathbf{f}_2 - \mathbf{f}_1) = 0$.

C GIP solution

First suppose that $\mathbf{f}_{\text{out}}$ solves the GIP. Then by definition $\mathbf{f}_{\text{out}} \in F$ and $\langle \mathsf{H}, \mathbf{f}_{\text{out}}\rangle = \mathbf{s}$. The equations (7.53) and (7.2) are valid by construction. As $\mathbf{p}_k \in \mathcal{K}$, we have (7.57). Conversely, suppose that $\langle \mathsf{H}, \mathbf{f}_{\text{out}}\rangle = \mathbf{s}$ and (7.57) holds. The formula (7.58) for $\mathbf{f}_{\text{out}}$ gives $B(\mathbf{f}_{\text{out}}, \mathbf{g}) = \sum_i \lambda_i B(\boldsymbol{\varphi}_i, \mathbf{g})$. Substituting (7.54) leads to $B(\mathbf{f},\mathbf{g}) = \sum_i \lambda_i \langle \mathbf{h}_i, \mathbf{g}\rangle$ for all $\mathbf{g} \in \mathscr{D}$. As $\mathbf{f} \in F$ and $\mathscr{D}$ is dense in F, the preceding formula holds also for all $\mathbf{f} \in F$ which permits us to apply Lemma 1.

D Dirac Laplacian

Consider $\langle \Delta g, v\rangle$, where v is a test function. This scalar product equals $-\int \nabla g \nabla v \, \mathrm{d}\mathbf{x}$. We change to spherical coordinates $\mathbf{x} \to (r, \phi_1, \ldots, \phi_{m-1})$. The integral becomes $-\int g' \frac{\partial v}{\partial r} r^{m-1} \mathrm{d}r\mathrm{d}\boldsymbol{\Omega}$ where $\mathrm{d}\boldsymbol{\Omega} = \mathrm{d}\phi_1 \ldots \mathrm{d}\phi_{m-1}$ and $\int \mathrm{d}\boldsymbol{\Omega} = S_m$. We use the fact that

$r^{m-1}g' = 1$. Then by integration over r we get $-\int [v(r)]_{r=0}^{r=\infty} \mathrm{d}\mathbf{\Omega}$. As v is a test function, $v(\infty) = 0$, and the integral simplifies to $-S_m[v]_0^\infty = S_m v(0)$. Consequently, $\Delta g = S_m \delta$.

Bibliography

[1] Lisa Brown, "A survey of image registration techniques," *ACM Computing Surveys*, vol. 24, no. 4, pp. 326–376, Dec. 1992.

[2] Jan Kybic, Philippe Thévenaz, and Michael Unser, "Multiresolution spline warping for EPI registration," in *Proceedings of SPIE*, Denver, Colorado, July 1999, vol. 3813, pp. 571–579, SPIE.

[3] Ruzena Bajcsy and Stane Kovačič, "Multiresolution elastic matching," *Computer Vision, Graphics, and Image Processing*, vol. 46, pp. 1–21, 1989.

[4] M. Unser, A. Aldroubi, and M. Eden, "B-spline signal processing: Part I—Theory," *IEEE Transactions on Signal Processing*, vol. 41, no. 2, pp. 821–832, Feb. 1993.

[5] M. Unser, A. Aldroubi, and M. Eden, "B-spline signal processing: Part II—Efficient design and applications," *IEEE Transactions on Signal Processing*, vol. 41, no. 2, pp. 834–848, Feb. 1993.

[6] C. de Boor, *A practical guide to splines*, Springer-Verlag, New York, 1978.

[7] E. H. W. Meijering, W. J. Niessen, and M. A. Viergever, "Quantitative evaluation of convolution-based methods for medical image interpolation," *Medical Image Analysis*, vol. 5, no. 2, pp. 111–126, 2001.

[8] P. Thévenaz, T. Blu, and M. Unser, "Interpolation revisited," *IEEE Transactions on Medical Imaging*, vol. 19, no. 7, pp. 739–758, July 2000.

[9] M. Satter and A. Goshtasby, "Registration of deformed images," in *Image Registration Workshop*, NASA Goddard Space Flight Center, Greenbelt, MD, 1997, pp. 221–229.

[10] Olivier Faugeras and Renaud Keriven, "Variational principles, surface evolution, PDE's, level set methods, and the stereo problem," *IEEE Transactions on Image Processing*, vol. 7, no. 3, pp. 336–344, Mar. 1998.

[11] A. Spinei, D. Pellerin, and J. Hérault, "Spatiotemporal energy-based method for velocity estimation," *Signal Processing*, vol. 65, pp. 347–362, 1998.

[12] C. Bernard, "Fast optical flow computation with discrete wavelets," Tech. Rep., Ecole Polytechnique, Paris, 1997.

[13] Y. Wu, T. Kanade, J. Cohn, and C. Li, "Image registration using wavelet-based motion model," in *Proceedings of Image Registration Workshop*, Jacqueline Le Moigne, Ed., 1997.

[14] F. Bookstein, *Morphometric Tools for Landmark Data: Geometry and Biology*, Cambridge University Press, 1997.

[15] K. Rohr, H. S. Stiehl, R. Sprengel, W. Beil, T. M. Buzug, J. Weese, and M. H. Kuhn, "Point-based elastic registration of medical image data using approximating thin-plate splines," in *Visualization in Biomedical Computing*, Karl Heinz Höhne and Ron Kikinis, Eds., pp. 297–306. Springer-Verlag, 1996.

[16] B. McGregor, "Automatic registration of images of pigmented skin lesions," *Pattern Recognition*, vol. 31, no. 6, pp. 805–817, 1998.

[17] Fred L. Bookstein, "Principal warps: Thin-plate splines and the decomposition of deformations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 6, no. 6, pp. 567–585, June 1989.

[18] C. Davatzikos, J. Prince, and R. Bryan, "Image registration based on boundary mapping," *IEEE Transactions on Medical Imaging*, vol. 15, no. 1, Feb. 1996.

[19] C. Huang, W. Chang, L. Wu, and J. Wang, "Three-dimensional PET emission scan registration and transmission scan synthesis," *IEEE Transactions on Medical Imaging*, vol. 16, no. 5, Oct. 1997.

[20] P. Rösch, J. Weese, T. Netsch, M. Quist, G.P. Penney, and D.L.G. Hill, "Robust 3D deformation field estimation by template propagation," in *Proceedings of MICCAI*, 2000.

[21] Fai Yeung, F. Levinson, Dongshan Fu, and Kevin J. Parker, "Feature-adaptive motion tracking of ultrasound image sequences using a deformable mesh," *IEEE Transactions on Medical Imaging*, vol. 17, no. 6, pp. 945–956, Dec. 1998.

[22] B. Horn and B. Schunck, "Determining optical flow," *Artificial Inteligence*, vol. 17, pp. 185–203, 1981.

[23] Gary Christensen, *Deformable Shape Models for Anatomy*, Ph.D. thesis, Washington University, Saint Louis, Mississippi, 1994.

[24] Gary Christensen, S. Joshi, and M. Miller, "Volumetric transformation of brain anatomy," *IEEE Transactions on Medical Imaging*, vol. 16, Dec. 1997.

[25] M. Bro-Nielsen and C. Gramkow, "Fast fluid registration of medical images," in *Visualization in Biomedical Computing*, Karl Heinz Höhne and Ron Kikinis, Eds., pp. 267–276. Springer-Verlag, 1996.

[26] Dan V. Iosifescu, Martha E. Shenton, Simon K. Warfield, Ron Kikinis, Joachim Dengler, Ferenc A. Jolesz, and Robert W. McCarley, "An automated registration algorithm for measuring MRI subcortical brain structures," *Neuroimage*, , no. 6, pp. 14–25, 1997.

[27] Y. Tai, K. Lin, C. Hoh, S. Huang, and E. Hoffman, "Utilization of 3D elastic transformations in the registration of chest X-ray CT and whole body PET," *IEEE Transactions on Nuclear Science*, vol. 44, no. 4, Aug. 1997.

[28] T. Schormann, S. Henn, and K. Zilles, "A new approach to fast elastic alignment with applications to human brains," in *Visualization in Biomedical Computing*, Karl Heinz Höhne and Ron Kikinis, Eds. Springer-Verlag, 1996.

[29] Ulrik Kjems, Stephen C. Strother, Jon Anderson, Ian Law, and Lars Kai Hansen, "Enhancing the multivariate signal of ^{15}O water PET studies with a new nonlinear neuroanatomical registration algorithm," *IEEE Transactions on Medical Imaging*, vol. 18, no. 4, pp. 306–319, Apr. 1999.

[30] J. Bergen, P. Anandan, K. Hanna, and R. Hingorani, "Hierarchical model-based motion estimation," in *Second European Conference on Computer Vision (ECCV'92)*. 1992, pp. 237–252, Springer-Verlag.

[31] Philippe Thévenaz, Urs E. Ruttimann, and Michael Unser, "A pyramid approach to subpixel registration based on intensity," *IEEE Transactions on Image Processing*, vol. 7, no. 1, pp. 1–15, Jan. 1998.

[32] N. Sicotte, R. Woods, and J. Mazziotta, "Automated image registration using a 105 parameter non-linear model," *Neuroimage*, vol. 3, no. 3, June 1996, Second International Conference on Functional Mapping of the Human Brain.

[33] S. Kiebel, J. Ashburner, J. Poline, and K. Friston, "MRI and PET coregistration—a cross validation of statistical parametric mapping and automated image registration," *Neuroimage*, , no. 5, 1997.

[34] John Ashburner and Karl J. Friston, "Nonlinear spatial normalization using basis functions," in *Brain Warping*, Arthur W. Toga, Ed., pp. 254–266. Academic Press, San Diego, 1999.

[35] Richard Szeliski and James Coughlan, "Spline-based image registration," *International Journal of Computer Vision*, vol. 22, pp. 199–218, 1997.

[36] Colin Studholme, R. Todd Constable, and James S. Duncan, "Incorporating an image distortion model in non-rigid alignment of EPI with conventional MRI," in *International Conference on Information Processing in Medical Imaging, Lecture Notes in Computer Science*, Visegrád, Hungary, June 1999, pp. 454–459, Springer-Verlag.

[37] Richard Szeliski and Heung-Yeung Shum, "Motion estimation with quadtree splines," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 12, pp. 1199–1207, Dec. 1996.

[38] H. Lester and S.R. Arridge, "Summarising fluid registration by thin-plate spline warps with many landmarks," in *Medical Image Understanding and Analysis*, July 1997, http://www.robots.ox.ac.uk/~mvl/miua97/.

[39] Z. Zhang, "Iterative point matching for registration of free-form curves and surfaces," *International Journal of Computer Vision*, vol. 13, no. 2, pp. 119–152, Dec. 1994.

[40] J. Weese, P. Rösch, T. Netsch, T. Blaffert, and Quist M., "Gray-value based registration of CT and MR images by maximization of local correlation," in *Proceedings of MICCAI*, 1999.

[41] Philippe Thévenaz and Michael Unser, "Optimization of mutual information for multiresolution image registration," *IEEE Transactions on Image Processing*, vol. 9, no. 12, pp. 2083–2099, Dec. 2000.

[42] B. Kim, J. Boes, K. Frey, and C. Meyer, "Mutual information for automated multimodal image warping," in *Visualization in Biomedical Computing*, Karl Heinz Höhne and Ron Kikinis, Eds. Springer-Verlag, 1996.

[43] F. Maes, A. Collignong, D. Vandermeulen, G. Marchal, and P. Suetens, "Multimodality image registration by maximization of mutual information," *IEEE Transactions on Medical Imaging*, vol. 16, no. 2, Apr. 1997.

[44] C. Studholme, D. Hill, and D. Hawkes, "Automated three-dimensional registration of magnetic resonance and positron emission tomography brain images by multiresolution optimization of voxel similarity measures," *Medical Physics*, vol. 24, no. 1, Jan. 1997.

[45] J. Duchon, "Splines minimizing rotation-invariant semi-norms in Sobolev spaces," in *Constructive Theory of Functions of Several Variables*, W. Schempp and K. Zeller, Eds., Berlin, 1977, pp. 85–100, Springer-Verlag.

[46] David Suter and Fang Chen, "Left ventricular motion reconstruction based on elastic vector splines," *IEEE Transactions on Medical Imaging*, vol. 19, no. 4, pp. 295–305, Apr. 2000.

[47] C. Quentin Davis and Dennis M. Freeman, "Statistics of subpixel registration algorithms based on spatiotemporal gradients or block matching," *Optical Engineering*, vol. 4, no. 37, pp. 1290–1298, Apr. 1998.

[48] C. Tomasi and R. Manduchi, "Stereo matching as a nearest-neighbor problem," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 3, Mar. 1998.

[49] M. Unser, G. Pelle, P. Brun, and M. Eden, "Computer analysis of M-mode echocardiograms: estimation of spatial deformation with time," in *Cardiovascular Dynamics and Models*, pp. 304–310. Institut National de la Santé et de la Recherche Médicale, Paris, 1988.

[50] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery, *Numerical Recipes in C*, Cambridge University Press, second edition, 1992.

[51] Gilbert Strang and G. Fix, *An Analysis of the Finite Element Method*, Wellesley-Cambridge Press, 1988.

[52] J. Martin, A. Pentland, S. Sclaroff, and R. Kikinis, "Characterization of neuropathological shape deformations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 2, no. 2, Feb. 1998.

[53] P. Moulin, R. Krishnamurthy, and J. Woods, "Multiscale modeling and estimation of motion fields for video coding," *IEEE Transactions on Image Processing*, vol. 6, no. 12, Dec. 1997.

[54] A. Brandt, "Multi-level adaptive techniques for boundary-value problems," *Math. Comput.*, vol. 31, no. 138, pp. 339–390, 1977.

[55] H. Yoshida, "Removal of normal anatomic structures in radiographs using wavelet-based non-linear variational method for image matching," in preparation.

[56] Y. Amit, "A non-linear variational problem for image matching," *SIAM Journal on Scientific Computing*, vol. 15, no. 1, pp. 207–224, 1994.

[57] J. Dennis, Jr. and R. Schnabel, *Numerical Methods for Unconstrained Optimisation and Nonlinear Equations*, SIAM J. Math. Anal., 1997.

[58] J. Thirion, "Fast non-rigid matching of 3D medical images," Tech. Rep. 2547, INRIA, Nice, May 1995.

[59] Simon Warfield, Andre Robatino, Joachim Dengler, Ferenc Jolesz, and Ron Kikinis, "Nonlinear registration and template-driven segmentation," in *Brain Warping*, Arthur W. Toga, Ed., pp. 67–84. Academic Press, San Diego, 1999.

[60] J.C. Gee, "On matching brain volumes," *Pattern Recognition*, , no. 32, pp. 99–111, 1999.

[61] Petra A. van den Elsen, Evert-Jan D. Pol, and Max A. Viergever, "Medical image matching—A review with classification," *IEEE Engineering in Medicine and Biology*, pp. 26–39, Mar. 1993.

[62] Hava Lester and Simon R. Arridge, "A survey of hierarchical non-linear medical image registration," *Pattern Recognition*, vol. 32, no. 1, pp. 129, Jan. 1999.

[63] Michael Unser and Thierry Blu, "Wavelets and radial basis functions: A unifying perspective," in *Proceedings of the SPIE Conference on Mathematical Imaging: Wavelet Applications in Signal and Image Processing VIII*, San Diego, CA, 2000, vol. 4119, pp. 487–493.

[64] L. Verard, P. Allain, J.M. Travere, J. C. Baron, and D. Bloyet, "Fully automatic identification of AC and PC landmarks on brain MRI using scene analysis," *IEEE Transactions on Medical Imaging*, vol. 16, no. 5, pp. 6100–616, Oct. 1997.

[65] N. Lam, "Spatial interpolation method: A review," *The American Cartographer*, vol. 10, no. 2, pp. 129–149, 1983.

[66] R. Franke, "Scattered data interpolation: test of some methods," *Mathematics of Computation*, vol. 38, no. 157, pp. 181–200, 1982.

[67] Michael Unser and Thierry Blu, "Wavelets and radial basis functions: A unifying perspective," in *Proceedings of the SPIE Conference on Mathematical Imaging: Wavelet Applications in Signal and Image Processing VIII*, San Diego, CA, 2000, vol. 4119.

[68] Heinz W. Werntges, "Partitions of unity improve neural function approximators," in *Proceedings of IEEE International Conference on Neural Networks*, San Francisco, CA, 1993, vol. 2, pp. 914–918.

[69] S. Mallat, *A Wavelet Tour of Signal Processing*, Academic Press, San Diego, CA, 1998.

[70] Maria Gabrani and Oleh J. Tretiak, "Surface-based matching using elastic transformations," *Pattern Recognition*, , no. 32, pp. 87–97, 1999.

[71] Jean Duchon, "Interpolation des fonctions de deux variables suivant le principe de la flexion des plaques minces," *Revue Française d'Automatique, Informatique et Recherche Operationelle*, vol. 10, no. 12, pp. 5–12, Dec. 1976.

[72] M. J. D. Powell, "The theory of radial basis function approximation in 1990," in *Advances in Numerical Analysis II: Wavelets, Subdivision Algorithms and Radial Functions*, W. A. Light, Ed., pp. 105–210. Oxford University Press, Oxford, UK, 1992.

[73] J. Liouville, "Sur une formule pour les différentielles à indices quelconques à l'occasion d'un mémoire de M. Tortolini," *J. Math. Pures Appl.*, vol. 20, pp. unnumbered, 1855, in French.

[74] Michael Unser and Thierry Blu, "Fractional splines and wavelets," *SIAM Review*, vol. 42, no. 1, pp. 43–67, 2000.

[75] Patrick Flandrin, "Wavelet analysis and synthesis of fractional Brownian motion," *IEEE Transactions Inform. Th.*, vol. 38, no. 2, pp. 910–917, Mar. 1992.

[76] Thomas Poggio and Federico Girosi, "Networks for approximation and learning," *Proceedings of IEEE*, vol. 78, no. 0, pp. 1481–1497, Sept. 1990.

[77] J. H. Ahlberg, E. N. Nilson, and J. L. Walsh, *The theory of splines and their applications*, Academic Press, New York, 1967.

[78] Jan Kybic, Philippe Thévenaz, Arto Nirkko, and Michael Unser, "Unwarping of unidirectionally distorted EPI images," *IEEE Transactions on Medical Imaging*, vol. 19, no. 2, pp. 80–93, Feb. 2000.

[79] Zang-Hee Cho, Joie P. Jones, and Manbir Singh, *Foundations of Medical Imaging*, John Wiley & Sons, 1993.

[80] Xin Wan, Grant T. Gullberg, Dennis L. Parker, and Gengsheng L. Zeng, "Reduction of geometric and intensity distortions in echo-planar imaging using a multireference scan," *Magnetic Resonance in Medicine*, vol. 37, no. 6, pp. 932–942, 1997.

[81] Hsuan Chang and J. Michael Fitzpatrick, "A technique for accurate magnetic resonance imaging in the presence of field inhomogeneities," *IEEE Transactions on Medical Imaging*, vol. 11, pp. 319–329, 1992.

[82] Bharat B. Biswal and James S. Hyde, "Contour-based registration technique to differentiate between task-activated and head motion-induced signal variations in fMRI," *Magnetic Resonance in Medicine*, vol. 38, no. 3, pp. 470–476, 1997.

[83] Andrew L. Alexander, Jay S. Tsuruda, and Dennis L. Parker, "Elimination of Eddy current artifacts in diffusion-weighted echo-planar images: The use of bipolar gradients," *Magnetic Resonance in Medicine*, vol. 38, no. 6, pp. 1016–1021, 1997.

[84] P. Jezzard and R. S. Balaban, "Correction for geometric distortion in echo planar images from B_0 field variations," *Magnetic Resonance in Medicine*, , no. 34, pp. 65–73, 1995.

[85] R. Woods, S. Grafton, N. Sicotte, and J. Mazziotta, "Automated image registration: II. Intersubject validation of linear and nonlinear models," *Journal of Computer Assisted Tomography*, vol. 22, no. 1, pp. 153–165, 1998.

[86] Arthur W. Toga, Ed., *Brain Warping*, Academic Press, San Diego, 1999.

[87] P.Thompson and Arthur W. Toga, "A surface-based technique for warping 3-dimensional images of the brain," *IEEE Transactions on Medical Imaging*, vol. 4, no. 15, pp. 1–16, Aug. 1996.

[88] C. Davatzikos, "Spatial normalization of 3D brain images using deformable models," *Journal of Computer Assisted Tomography*, vol. 4, no. 20, pp. 656–665, 1996.

[89] J. Declerck, G. Subsol, J. Ph. Thirion, and N. Ayache, "Automatic retireval of anatomical structures in 3D medical images," in *CVRMed'95, Lecture Notes in Computer Science*, N. Ayache, Ed., pp. 153–162. Springer-Verlag, Nice, France, Apr. 1995.

[90] J.H. Ahlberg, E.N. Nilson, and J.L. Walsh, *The Theory of Splines and Their Applications*, Academic Press, New York, 1967.

[91] I.J. Schoenberg, "Spline functions and the problem of graduation," *Proc. Nat. Acad. Sci.*, vol. 52, pp. 947–950, 1964.

[92] Thierry Blu and Michael Unser, "Quantitative Fourier analysis of approximation techniques: Part I—interpolators and projectors," *IEEE Transactions on Signal Processing*, 1999.

[93] Michael Unser, Akram Aldroubi, and Murray Eden, "Fast B-spline transforms for continuous image representation and interpolation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 3, Mar. 1991.

[94] Gérard Subsol, *Construction automatique d'atlas anatomiques morphométriques à partir d'images médicales tridimensionnelles*, Ph.D. thesis, Ecole Centrale, Paris, France, 1995.

[95] Frank Müller, Patrick Brigger, Klaus Illgner, and Michael Unser, "Multiresolution approximation using shifted splines," *IEEE Transactions on Signal Processing*, vol. 46, no. 9, pp. 2555–2558, Sept. 1998.

[96] Grace Wahba, *Spline Models for Observational Data*, SIAM, Philadelphia, PA, 1990.

[97] Josien P. W. Pluim, J. B. Antoine Maintz, and Max A. Viergever, "Image registration by maximization of combined mutual information and gradient information," *IEEE Transactions Med. Imag.*, vol. 19, no. 8, Aug. 2000.

[98] Yu-Te Wu, *Image Registration using wavelet-based motion model and its applications*, Ph.D. thesis, University of Pittsburgh, Saint Louis, Mississippi, 1997.

[99] M. Unser, A. Aldroubi, and M. Eden, "A family of polynomial spline wavelet transforms," *Signal Processing*, vol. 30, no. 2, Jan. 1993.

[100] Lanitis A., C.J. Taylor, and T.F. Cootes, "Automatic interpretation and coding of face images using flexible models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 743–756, July 1997.

[101] Michael Unser, Akram Aldroubi, and Murray Eden, "The l_2 polynomial spline pyramid," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 4, Apr. 1993.

[102] I. Kanno and N.A. Lassen, "Two methods for calculating regional cerebral blood flow from emission computed tomography of inert gas concentrations," *Journal of Computer Assisted Tomography*, vol. 1, no. 3, pp. 71–76, 1979.

[103] Jan Kybic and Michael Unser, "Multidimensional elastic registration of images using splines," in *Proceedings of ICIP*, Vancouver, Canada, 2000, vol. II, pp. 455–458.

[104] María J. Ledesmay-Carbayo, Jan Kybic, Manuel Desco, Andrés Santos, and Michael Unser, "Cardiac motion analysis from ultrasound sequences using non-rigid registration," in *Proceedings of MICCAI*, Utrecht, The Netherlands, Oct. 2001, in print.

[105] Jan Kybic, Thierry Blu, and Michael Unser, "Generalized sampling: A variational approach. Part I — Applications," *IEEE Transactions on Signal Processing*, 2001, submitted.

[106] C.E. Shannon, "Communication in the presence of noise," in *Proc. IRE*, Jan. 1949, vol. 37, pp. 10–21.

[107] A. Papoulis, "Generalized sampling expansion," *IEEE Transactions Circ. Syst.*, vol. 24, pp. 652–654, 1977.

[108] J. L. Brown Jr. and K. Sangsari, "Sampling reconstruction of n-dimensional bandlimited images after multi-linear filtering," *IEEE Transactions Circ. Syst.*, pp. 1035–1038, July 1989.

[109] D. Seidner, M. Feder, D. Cubanski, and S. Blackstock, "Introducing to vector sampling expansions," *IEEE Signal Process. Lett.*, vol. 5, pp. 115–117, May 1998.

[110] I. Djokovic and P.P. Vaidyanathan, "Generalized sampling theorems in multiresolution subspaces," *IEEE Transactions on Signal Processing*, vol. 45, no. 3, pp. 583–599, 1997.

[111] G. Thomas, "A comparison of motion-compensated interlace-to-progressive conversion methods," *Signal Processing: Image Communications*, vol. 12, no. 3, pp. 209–29, 1998.

[112] E. Bellers and G. de Haan, "New algorithm for motion estimation on interlaced video," in *Proceedings of SPIE—Visual Communication and Image Processing*, 1998, vol. 3309, pp. 111–121.

[113] H. Ur and D. Gross, "Improved resolution from subpixel shifted pictures," *Computer Vision, Graphics, and Image Processing*, vol. 54, no. 2, pp. 181–186, 1991.

[114] H. Shekarforoush, M. Berthod, and Josianne Zerubia, "3D super-resolution using generalized sampling expansion," in *Proceedings of Int. Conf. Image Processing*, Washington, DC, 1995, vol. II, pp. 300–303.

[115] Michael Unser and Akram Aldroubi, "A general sampling theory for nonideal acquisition devices," *IEEE Transactions on Signal Processing*, vol. 42, no. 11, pp. 2915–2925, Nov. 1994.

[116] Michael Unser and Josiane Zerubia, "A generalized sampling theory without bandlimiting constraints," *IEEE Transactions on Circuits and Systems—II: Analog and digital signal processing*, vol. 45, pp. 959–969, Aug. 1998.

[117] Jonathan C. Carr, W. Richard Fright, and Richard K. Beatson, "Surface interpolation with radial basis functions for medical imaging," *IEEE Transactions Med. Imag.*, vol. 16, no. 1, Feb. 1997.

[118] H. J. Landau, "Necessary density conditions for sampling and interpolation of certain entire functions," *Acta Mathematica*, vol. 117, pp. 37–52, 1967.

[119] Ahmed I. Zayed and Antonio G. García, "Nonuniform sampling of bandlimited signals with polynomial growth on the real axis," *IEEE Transactions on Information Theory*, vol. 43, no. 5, pp. 1717–1721, Sept. 1997.

[120] Akram Aldroubi and H. Feichtinger, "Exact iterative reconstruction algorithm for multivariate irregularly sampled functions in spline-like spaces: the l_p theory," *Proceedings of American Mathematical Society*, vol. 126, no. 9, pp. 2677–2686, 1998.

[121] A. J. Jerri, "The Shannon sampling theorem—its various extensions and applications: A tutorial review," *Proceedings of IEEE*, vol. 65, no. 11, pp. 1565–1596, 1977.

[122] Michael Unser, "Sampling–50 years after Shannon," *Proceedings of the IEEE*, vol. 88, no. 4, April 2000.

[123] Jelena Kovačević and Martin Vetterli, "Nonseparable two- and three-dimensional wavelets," *IEEE Transactions on Signal Processing*, vol. 43, no. 5, pp. 1269–1273, May 1995.

[124] X.G. Xia and B.W. Suter, "Vector-valued wavelets and vector filter banks," *IEEE Transactions on Signal Processing*, vol. 44, no. 3, pp. 508–518, Mar. 1996.

[125] E. Arge, Daehlen. M., and A. Tveito, "Box spline interpolation - a computational study," *Journal of computational and applied mathematics*, vol. 44, no. 3, pp. 303–329, Dec. 1992.

[126] G. G. Walter, "A sampling theorem for wavelet subspaces," *IEEE Transactions Inform. Th.*, vol. 38, pp. 881–884, 1992.

[127] Eric W. Weisstein, *CRC Concise Encyclopedia of Mathematics*, Chapman & Hall/CRC, Boca Raton, Florida, 1998.

[128] Charles A. Micchelli, "Interpolation of scattered data: Distance matrices and conditionally positive definite functions," *Constr. Approx.*, , no. 2, pp. 11–22, 1986.

[129] D.M. Allen, "The relationship between variable selection and data augmentation and a method for prediction," *Technometrics*, vol. 1, no. 16, pp. 125–127, 1974.

[130] I.J. Schoenberg, *Cardinal spline interpolation*, SIAM, Philadelphia, PA, 1973.

[131] L.L. Schumaker, *Spline Functions Basic Theory*, Wiley&Sons, New York, NY, 1981.

[132] Yu. Lyubarskii and Madych W. R., "The recovery of irregularly sampled band limited functions via tempered splines," *Journal of Functional Analysis*, vol. 125, no. 1, pp. 201–222, Oct. 1994.

[133] Akram Aldroubi, Michael Unser, and Murray Eden, "Cardinal spline filters: stability and convergence to the ideal sinc interpolator," *Signal Processing*, vol. 28, no. 2, pp. 127–138, 1992.

[134] Jonathan Maltz, Robert De Mello Koch, and Andrew Willis, "Reproducing kernel Hilbert space method for optimal interpolation of potential field data," *IEEE Transactions on Image Processing*, vol. 7, no. 12, Dec. 1998.

[135] Anil K. Jain, *Fundamentals of Digital Image Processing*, Prentice Hall, 1989.

[136] Frank Natterer, *The mathematics of computerized tomography*, John Wiley & Sons, 1986.

[137] Jan Kybic, Thierry Blu, and Michael Unser, "Variational approach to tomographic reconstruction," in *Proceedings of SPIE*, San Diego, CA, Feb. 2001.

[138] Jan Kybic, Thierry Blu, and Michael Unser, "Generalized sampling: A variational approach. Part II — Theory," *IEEE Transactions on Signal Processing*, 2001, submitted.

[139] L. Schwartz, *Théorie des Distributions*, Hermann, Paris, France, 1966, in French.

[140] Ian Richards and Heekyung Youn, *Theory of Distributions: A non-technical introduction*, Cambridge University Press, 1990.

[141] Gene H. Golub and Charles F. Van Loan, *Matrix Computations*, The Johns Hopkins University Press, third edition, 1996.

Conclusions

Elastic registration is an unavoidable part of any comparison or evaluation of biomedical images involving different subjects, techniques, or acquired at different times. It permits to eliminate the differences due to different acquisition conditions, while keeping the relevant changes.

The intensity-based approach is preferable for automatic registration because it alleviates the need for a difficult feature detection step. The sum of square differences is a reasonable measure of the quality of the fit, and we have found it to perform well in our applications. B-spline representation of both the image and the deformation is computationally efficient, has good approximation properties and lends itself well to a multiresolution approach. The described automatic algorithm takes advantage of all these properties; it is reasonably fast, robust, and has been used in a number of real applications, including MRI, SPECT and ultrasound imaging.

A semi-automatic method combining the automatic method with landmark information yields good results even in situation where the automatic method by itself fails, and its accuracy is much better than that of purely landmark based registration.

Classical landmark interpolation can be formulated as a special case of a more general variational interpolation or approximation problem. Any object can be reconstructed given a quadratic plausibility criterion and a set of linear measurements. Moreover, the reconstruction lies in a vector space and is determined by a simple linear equation set.

Availability of the software

I wrote countless lines of code while working on my Ph.D. thesis. A major part of what I wrote is available to your perusal under the GNU Public License (GPL),[4] provided you give me the appropriate credit. The GPL basically means you can use my programs for free but there is no waranty and you may not incorporate it into a proprietary program. The source code can be found either on a CD coming with some copies of this thesis or from my web page: `http://bigwww.epfl.ch/kybic`. If you publish your work and you have used my code, you should reference either this thesis or one of our relevant papers.

The first generation of the registration algorithm for unidirectional deformations was written in *Matlab*, in summer 1998. The time critical routines were reimplemented in *C* and the resulting algorithm is described in Chapter 4. The second generation added support for multidimensional deformations, still in *Matlab* and *C*, was used to register the SPECT images (Chapter 5) and in part of the ultrasound experiments. The third generation of the registration algorithm, developed in January 2001, sports *Python* as its glue language, replacing *Matlab*. *Python* is free, portable, and results in much better structured code. Functionally, it is equivalent to the second generation, only faster. Finally the current, fourth reincarnation is even faster, the code has been restructured, the deformation model modified, and new optimizer introduced. It also adds the semi-automatic virtual spring mode. All experiments in Chapter 5 (except SPECT registration) were performed using the fourth generation code.

The variational reconstruction algorithms (Chapters 6 and 7) and landmark warping (Chapter 3) were mostly implemented in *Matlab* with some time critical routines reimplemented in *C*.

I also wrote web server software *MOW* (Matlab on the Web) permitting to make interactive demonstrations written in *Matlab* available on the Web. The server part is written in *Python*, the client part is a *Java* applet. A variant called *MOS* (Matlab on the server) eliminates the need for *Java* by generating the complete pages on the server.

All software was developed on Unix. However, it should not be difficult to port it to other platforms. In fact, the third generation of the registration algorithm was

[4] Available from `http://www.gnu.org/copyleft/gpl.html` or by mail from *Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA*

successfully run under the *Windows 95* operating system.

Curriculum vitæ

July 6, 2001

Jan KYBIC
BIG IOA DMT
Ecole Polytechnique Fédérale de Lausanne
CH-1015 Lausanne
Switzerland
+41-(21)-693 5142 (work)
Jan.Kybic@epfl.ch, kybic@ieee.org
http://bigwww.epfl.ch/kybic

Born on 18^{th} January 1974 in Prague, Czech Republic. Male.

Education

1998–2001 PhD in biomedical image processing at *Ecole Polytechnique Fédérale de Lausanne (EPFL)*, Switzerland, with thesis *Biomedical Image Registration by Elastic Warping*.

1996–1998 Master (Engineer) in control engineering from *Czech Technical University (CTU)*, with honors, with thesis *Kalman Filtering and Speech Enhancement*

1994–1996 Bachelor in control engineering from *CTU*, with thesis *Programming Contest Scoring System*

Educational travel

Autumn 1997 Diploma work at *EPFL*, Switzerland, in the *CIRC* laboratory (now *LANOS*), with Prof. Martin Hasler.

May 1997 Work on fuzzy interpolation in *Budapest Technical University*, sponsored by *CEEPUS*, with Prof. Laszlo Koczy.

Summer 1996 *Global Development Program*, organized by *Milwaukee School of Engineering* and *Allen-Bradley*, in Milwaukee, Wisconsin.

Autumn 1995 Technology and financing course by *Institut français du petrol*, sponsored by *EST*.

Summer 1993 A *CTU* delegate at *Youth International Science Forum* in London, invited speaker in 1995.

Work experience

1998–2001 Assistant at EPFL.

1997 System administrator for *Gide-Loyrette-Noel*, Prague.

1997 Windows programmer for *SIDAT*, Prague, on a contract for *Siemens*, Erlangen.

Summer 1995 Serial communication protocol development and system administration for *Digitron, AB*, in Aarau, Switzerland, organized by *IAESTE*.

Summer 1994 Developing videophone applications at *Helsinki University of Technology*, Finland, organized by *IAESTE*.

Summer 1992 Evaluation and plotting software development for *Continental Microwave Technology, Ltd.*, England.

Research interests

Signal and image processing, image registration, splines and wavelets, speech processing and enhancement, computer vision, numerical methods, algorithm theory, parallel algorithms, control theory.

Language skills

Mother tongue Czech, fluent English, French, and Spanish, conversational knowledge of German, and Italian, intermediate Russian.

Computer skills

Program development using C, Python, Matlab, IDL, Perl, Java, Tcl/Tk, PVM. Web design. Typesetting using TEX. Linux internals. Linux/UNIX administration.

Professional activities

Article reviews for *IEEE Transactions on Medical Imaging*. Student member of *IEEE*, *Czech TEX Users Group*, *Czech Linux Users Group*. Organized *Local ACM Programming Contest* at *CTU* in 1996–1999.

Competitions

Silver medal at the *International Olympiad in Informatics* in Bonn in 1992. Member of the *CTU* team in *ACM Programming Contest* in Zürich in 1995 and 1996 (second place), world finals in San Jose, California, 1997.

Hobbies

Music (piano), photography, travelling, literature, cinema, hiking, volleyball, skiing, sailing.

Publications

- Jan Kybic, Philippe Thévenaz, Arto Nirkko, and Michael Unser, "Unwarping of unidirectionally distorted EPI images," *IEEE Transactions on Medical Imaging*, vol. 19, no. 2, pp. 80–93, Feb. 2000.
- Jan Kybic, Thierry Blu, and Michael Unser, "Generalized sampling: A variational approach. Part I — Applications," *IEEE Transactions on Signal Processing*, 2001, submitted.
- Jan Kybic, Thierry Blu, and Michael Unser, "Generalized sampling: A variational approach. Part II — Theory," *IEEE Transactions on Signal Processing*, 2001, submitted.
- Jan Kybic and Michael Unser, "Fast multidimensional elastic image registration," *IEEE Transactions on Image Processing*, 2001, in preparation.
- María J. Ledesmay-Carbayo, Jan Kybic, Manuel Desco, Andrés Santos, and Michael Unser, "Cardiac motion analysis from ultrasound sequences using non-rigid registration," in *Proceedings of MICCAI*, Utrecht, The Netherlands, Oct. 2001, in print.
- Jan Kybic, Thierry Blu, and Michael Unser, "Generalized sampling: A variational approach," in *Proceedings of SampTA*, Orlando, Florida, May 2001, in print.
- Jan Kybic, Thierry Blu, and Michael Unser, "Variational approach to tomographic reconstruction," in *Proceedings of SPIE*, San Diego, CA, Feb. 2001.
- Jan Kybic and Michael Unser, "Multidimensional elastic registration of images using splines," in *Proceedings of ICIP*, Vancouver, Canada, 2000, vol. II, pp. 455–458.
- Jan Kybic, Philippe Thévenaz, and Michael Unser, "Compensation of unidirectional geometric distortion in EPI using spline warping," in *Proceedings of the IEEE Internation Conference on Image Processing*, Kobe, Japan, 1999.
- Jan Kybic, Philippe Thévenaz, and Michael Unser, "Multiresolution spline warping for EPI registration," in *Proceedings of SPIE*, Denver, Colorado, July 1999, vol. 3813, pp. 571–579, SPIE.
- Petr Sovka, Petr Pollak, and Jan Kybic, "Extended spectral subtraction," in *Proceedings of EUSIPCO'96*, 1996, vol. 2, p. 963.
- Peter Baranyi, Jan Kybic, Yeung Yam, and Laszlo Koczy, "Extension of singular value-based rule base reduction to the general fuzzy rule interpolation," Tech. Rep., Budapest Technical University, Budapest, Hungary, 1997.