

A GPU Parallelised Multimodal Registration Framework for Improved Anatomical Consistency



Frederik J Lange
St Catherine's College
University of Oxford

A thesis submitted for the degree of
Doctor of Philosophy
Trinity 2020

Acknowledgements

Personal

While it is a requirement to state that this thesis is my own work, I cannot pretend that the effort was all mine. There are many people without whose support and encouragement this work would never have been possible. In particular I wish to thank:

Firstly, my supervisors Jesper Andersson and Steve Smith. Your guidance in the technical aspects of my work has been exceptional, and I continue to learn more with each discussion we have. More than that though, I appreciate the time and energy you have put into supporting me both personally and professionally. Your willingness to fight for what was best for me has never gone unnoticed.

John Ashburner, for conversations and contributions to my first academic paper.

My friends and colleagues at the WIN, especially: Sophie Schauman, Amy Howard, Nicola Dinsdale, Carinne Piekma, Alex Smith, Daniel Papp and Christoph Arthofer. A wise person once said, “It is a wonderful thing to have research friends”.

All of my friends from Table 4: Matt and LaRae Cherukara, Tuti Ng’ang’a, Sam and Katie Aroney, David and Abigail Felce, Jordan and Kristin Cardenas, Peter Taylor and Sean Ravenhall. Thanks for all the prayers, cake, loud laughter and squash.

My ONBI cohort, coordinator Peter Jezzard, and administrator Lisa Bligh. I could not have asked for a better group to help me settle in at Oxford.

Many others who have provided support and friendship: Douglas and Patricia Dalrymple, Melissa Skorka, Alex von Klemperer, Edd Haigh and my housemates over the years; Kiyana Zarnani, Anna Booth, and especially Nick Hall.

My entire family, and in particular both sets of parents. Mar and Dad, I will forever be grateful for the way you raised me - for teaching me to value kindness above all, and embodying that ideal yourselves. And for the myriad sacrifices, both large and small, that you have made for your children’s well-being and education. Mom and Pops, you are the most thoughtful and generous people I have ever known. You have made my time in Oxford better in countless ways, without once expecting anything in return, and I hope to follow your example in the future.

Finally, and most importantly, my wife Sarah Fynes-Clinton. Every step on the path towards this point has only been possible thanks to you. Without your support - both emotional and practical - unending kindness, menace, and belief in me I would most likely still be in Ogies, eyeball deep in coal dust. Thank you.

Institutional

I wish to thank the Oppenheimer Memorial Trust, Rotary International, St Catherine's College Oxford and The Guarantors of Brain for funding I have received over the years.

The Wellcome Centre for Integrative Neuroimaging is supported by core funding from the Wellcome Trust [grant number 203139/Z/16/Z]

Data were provided [in part] by the Human Connectome Project, WU-Minn Consortium (Principal Investigators: David Van Essen and Kamil Ugurbil; 1U54MH091657) funded by the 16 NIH Institutes and Centers that support the NIH Blueprint for Neuroscience Research; and by the McDonnell Center for Systems Neuroscience at Washington University.

This work was supported by funding from the Engineering and Physical Sciences Research Council (EPSRC) and Medical Research Council (MRC) [grant number EP/L016052/1].

Abstract

Biomedical image registration is well established as a core component of most neuroimaging processing pipelines. It allows multiple subjects to be analysed using a common frame of reference, thereby facilitating both the localisation and quantification of similarities and differences between individuals or population groups.

An assumption inherent to such analyses is that registration produces a one-to-one mapping between common anatomical locations across subjects. Errors in image registration violate this assumption, typically reducing both the power to detect effects of interest and the ability to localise those effects. But image registration is an inherently ill-posed problem, and therefore the true, unique one-to-one mapping can generally never be found (if it even exists). Instead, registration methods can only attempt to produce the most likely deformation, given the subject data and models of what type of mappings are more probable than others.

In this thesis I present two approaches to improving the accuracy of nonlinear registration, particularly in the context of magnetic resonance imaging of the brain.

The first utilises a computationally complex regularisation model which penalises the stretching effect of a deformation in a highly nonlinear way. The method produces more anatomically plausible mappings, with less distortion of both volume and shape, when compared to current state-of-the-art methods. It achieves this while simultaneously equalling or bettering the alignment accuracy, as measured by the overlap of expert-segmented anatomical brain regions.

The second leverages the fact that it is increasingly common for both scalar and tensor imaging modalities, containing complementary spatial information, to be acquired for each subject in a dataset. The similarity between multiple modalities are then simultaneously compared, accounting for differences in both tissue location and orientation, to better inform the registration in a data-driven way. This leads to a measurable and significant improvement in anatomical consistency across subjects, evaluated using group-level task-based brain activation maps.

The cost associated with these improvements is an increase in algorithmic complexity. This is addressed by leveraging the massively parallelised computing capabilities of graphics processing units to offset the rise in complexity. To this end, I present my implementation of a new software tool, the MultiMOdal Registration Framework (MMORF), which forms the basis of all of the results herein.

My hope is that the benefits I present in this thesis will allow MMORF to have a real impact in unifying volumetric image registration in the field of neuroimaging.

Contents

List of Figures	x
List of Abbreviations	xii
1 Introduction	1
1.1 Introduction	2
1.1.1 History of Image Registration	3
1.1.2 Why We Still Need MMORF	6
1.2 MRI Modalities	8
1.2.1 T1-Weighted Imaging	10
1.2.2 T2-Weighted Imaging	10
1.2.3 Diffusion-Weighted Imaging	11
1.3 What is Image Registration?	12
1.3.1 Image Registration as Domain Mapping	13
1.3.2 Linear Image Registration	14
1.3.3 Nonlinear Image Registration	14
1.4 Nonlinear Deformation Models	15
1.4.1 Small Deformation Framework	15
1.4.2 Large Deformation Models	17
1.4.3 Diffeomorphism	18
1.5 Image Registration as Optimisation	19
1.5.1 Nonlinear Optimisation	20
1.5.2 Gradient-based Methods	21
1.5.3 Newton Methods	23
1.6 Image (Dis)Similarity	25
1.6.1 Global Intensity Measures	26
1.6.2 Local Intensity Measures	29
1.7 Regularisation	30
1.7.1 Mechanical	31
1.7.2 Smoothing	32
1.7.3 Hyperelastic	33
1.8 Exemplar Methods	34

1.8.1	FNIRT	34
1.8.2	Demons	35
1.8.3	ANTs	36
1.8.4	DARTEL	36
1.8.5	Geodesic Shooting	36
1.8.6	NiftyReg	37
1.8.7	Diffusion Methods	37
1.9	General Purpose Graphics Processing Units	37
1.9.1	Hardware Terms	38
1.9.2	Execution Coordination Terms	39
1.9.3	Performance Terms	41
2	The Multimodal Registration Framework	44
2.1	Introduction	45
2.2	Aims of MMORF	45
2.2.1	Assumptions	46
2.2.2	Constraints	46
2.3	Transformation Model	46
2.3.1	Small Deformation Framework	46
2.3.2	Parametrisation	48
2.4	Optimisation Strategy	52
2.5	Image Similarity	54
2.6	Multiresolution Pyramid and Smoothing	55
2.7	Modality Weighting	57
2.8	Software Design	62
2.8.1	UML Notation	62
2.8.2	Object Functionality	63
2.8.3	Object Relationships	67
2.8.4	Object Interactions	68
2.9	Code Optimisation	70
2.9.1	GPU Specific Considerations	70
2.9.2	Profiling Results	76
2.10	Conclusion	82
3	A Symmetric Prior for the Regularisation of Elastic Deformations	84
3.1	Introduction	85
3.2	Methods	89
3.2.1	Theory	89
3.2.2	Intuition	96
3.2.3	Implementation	97

3.2.4	2D Example	100
3.2.5	Testing	101
3.3	Results	111
3.3.1	Overlap Scores	111
3.3.2	Jacobian Determinant Distributions	115
3.3.3	CVAR Distributions	118
3.3.4	Spatial Maps	119
3.4	Discussion	126
3.4.1	Relationship to Earlier Work	128
3.4.2	Beyond Enforcing Diffeomorphic Warps	129
3.4.3	Other Parallelised Algorithms	133
3.4.4	Registration Accuracy	134
3.4.5	Anatomical Plausibility	134
3.4.6	Performance Summary	136
3.5	Conclusion and Outlook	136
4	Diffusion Tensor Registration	138
4.1	Introduction	139
4.2	Motivation	139
4.3	Diffusion Imaging Modalities	139
4.4	Similarity Metric	140
4.5	Tensor Reorientation	142
4.6	Implementation Within MMORF	145
4.6.1	Cost and Transformation	145
4.6.2	Gradient	147
4.6.3	Hessian	149
4.7	Computational Complexity	150
4.7.1	Translation Complexity	151
4.7.2	Rotation Complexity	151
4.8	Existing Methods	152
4.8.1	DTI-TK	152
4.8.2	DT-REFinD	153
4.8.3	DR-TAMAS	154
4.9	Reorientation Testing	154
4.10	Single Subject Comparison	157
4.10.1	Methods	158
4.10.2	Results and Discussion	164
4.10.3	Conclusion	168

5	Multimodal Template Creation	171
5.1	Introduction	172
5.2	Motivation	172
5.3	An Unbiased Average	172
5.4	Existing Multimodal Templates	173
5.4.1	ICBM (Human)	174
5.4.2	SRI24 (Human)	175
5.4.3	VALiDATe29 (Squirrel Monkey)	176
5.4.4	TORTOISE HCP (Human)	177
5.5	Methodology	178
5.5.1	Initial Affine Template	179
5.5.2	Nonlinear Template Iterations	180
5.6	Lemur and Macaque Template	182
5.6.1	Ethical Approval	182
5.6.2	Methods	183
5.6.3	Results and Discussion	185
5.6.4	Conclusion	190
5.7	UK Biobank Template	190
5.7.1	Data Acquisition and Preprocessing	191
5.7.2	Methods	192
5.7.3	Results and Discussion	194
5.7.4	Conclusion	194
6	Multimodal vs Unimodal Accuracy	196
6.1	Introduction	196
6.2	Motivation	197
6.3	Methods	197
6.3.1	Template	198
6.3.2	Test Dataset	198
6.3.3	Affine Registration	199
6.3.4	Nonlinear Registration	200
6.3.5	Resampling	200
6.3.6	Group tfMRI Analysis	200
6.3.7	Warp Aggressiveness	203
6.4	Results	203
6.5	Discussion	210
6.6	Conclusion	214

7	Conclusion and Future Directions	215
7.1	Summary of Contributions	215
7.1.1	MMORF	216
7.1.2	SPRED	216
7.1.3	DTI Registration	217
7.1.4	Multimodal Template Creation	217
7.1.5	Unimodal versus Multimodal Accuracy	217
7.2	Future Directions	218
7.3	Final Conclusion	219
	References	220

List of Figures

1.1	T1-Weighted Image	10
1.2	T2-Weighted Image	11
1.3	Diffusion-Derived Image	12
1.4	Coalesced Memory Access	43
2.1	B-spline Overlap	51
2.2	Hessian Symmetry	52
2.3	UML Symbols	63
2.4	WarpField UML	64
2.5	CostFxn UML	65
2.6	Optimiser UML	66
2.7	LinearSolver UML	67
2.8	MMORF Object Relationships UML	68
2.9	MMORF Object Interactions UML	69
2.10	Sparse Diagonal Matrix	74
2.11	$\partial \mathbf{C} / \partial \mathbf{w}$ Speedup	79
2.12	$\mathbf{H}_{\text{symmetric}}$ Speedup	80
2.13	$\mathbf{H}_{\text{non-symmetric}}$ Speedup	81
3.1	Exact vs Approximate SPRED Penalty	93
3.2	2D B-spline Displacement Field and Spatial Derivatives	102
3.3	2D Jacobian Matrix Fields	103
3.4	2D Cost Field and Jacobian Matrix Gradient	103
3.5	2D Cost Field and Warp Parameter Gradient	104
3.6	Jaccard Coefficient Region-wise Comparison (Tools)	112
3.7	Jaccard Coefficient Combined Comparison (Tools)	113
3.8	MMORF vs ANTs-CC Jaccard Coefficients	114
3.9	Jaccard Coefficient Region-wise Comparison (Regularisers)	115
3.10	Jaccard Coefficient Combined Comparison (Regularisers)	116
3.11	Jacobian Determinant Distributions	117
3.12	Log-Jacobian Determinant Range Distributions (Tools)	118
3.13	MMORF vs ANTs-CC Log-Jacobian Determinant Range	119

3.14	Log-Jacobian Determinant Range Distributions (Regularisers)	120
3.15	Mean CVAR Distributions (Tools)	121
3.16	Mean CVAR Distributions (Regularisers)	122
3.17	Jacobian Determinant Spatial Maps (Tools)	123
3.18	CVAR Spatial Maps (Tools)	124
3.19	Jacobian Determinant Spatial Maps (Regularisers)	125
3.20	CVAR Spatial Maps (Regularisers)	127
3.21	Comparison of Deformed Grids	128
4.1	Changing Tensor Properties	141
4.2	NR vs FS vs PPD Reorientation	144
4.3	Tensors For Rotation Test	155
4.4	Reference and Moving Tensors for Rotation Test	156
4.5	Warp Fields for Rotation Test	157
4.6	Warped DEC Maps	164
4.7	DR-TAMAS: PPD vs PS Reorientation	166
4.8	Jacobian Determinant Maps	167
4.9	Log-Jacobian Determinant Distributions	168
5.1	ICBM templates	175
5.2	SRI24 template	176
5.3	VALiDATE29 template	177
5.4	TORTOISE template	179
5.5	Lemur Template	185
5.6	Macaque Template	186
5.7	Lemur and Macaque Template Differences	187
5.8	UKB-25 Scalp Signal	193
5.9	UKB-25 template	195
6.1	Group Level Activation Maps	205
6.2	Bland-Altman Masked t-statistic Comparison	206
6.3	5 th to 95 th log-Jacobian Determinant Percentile Range	208
6.4	Jacobian Determinant Maps	210
6.5	Bland-Altman Log-Jacobian Determinant Comparison	211

List of Abbreviations

1D, 2D, 3D	One, two or three Dimensional
CPU	Central Processing Unit
CUDA	Compute Unified Device Architecture
CVAR	Cube-Volume Aspect Ratio
DTI	Diffusion Tensor Imaging
FS	Finite Strain
GN	Gauss-Newton
GPU	Graphics Processing Unit
HCP	Human Connectome Project
L1, L2, L3	Primary, secondary or tertiary Eigenvalue
LM	Levenberg-Marquardt
LNCC	Local Normalised Cross Correlation
MM	Majorise-Minimisation or Minorise-Maximisation
MMORF	MultiMODal Registration Framework
MRI	Magnetic Resonance Imaging
MSQ	Mean Squared Error
NR	No Reorientation
PPD	Preservation of Principle Direction
RF	Radio Frequency
SPRED	Symmetric Prior for the Regularisation of Elastic Deformations
tfMRI	Task Functional Magnetic Resonance Imaging
UKB	UK Biobank
UML	Unified Modelling Language
V1, V2, V3	Primary, secondary or tertiary Eigenvector

1

Introduction

Contents

1.1	Introduction	2
1.1.1	History of Image Registration	3
1.1.2	Why We Still Need MMORF	6
1.2	MRI Modalities	8
1.2.1	T1-Weighted Imaging	10
1.2.2	T2-Weighted Imaging	10
1.2.3	Diffusion-Weighted Imaging	11
1.3	What is Image Registration?	12
1.3.1	Image Registration as Domain Mapping	13
1.3.2	Linear Image Registration	14
1.3.3	Nonlinear Image Registration	14
1.4	Nonlinear Deformation Models	15
1.4.1	Small Deformation Framework	15
1.4.2	Large Deformation Models	17
1.4.3	Diffeomorphism	18
1.5	Image Registration as Optimisation	19
1.5.1	Nonlinear Optimisation	20
1.5.2	Gradient-based Methods	21
1.5.3	Newton Methods	23
1.6	Image (Dis)Similarity	25
1.6.1	Global Intensity Measures	26
1.6.2	Local Intensity Measures	29
1.7	Regularisation	30
1.7.1	Mechanical	31
1.7.2	Smoothing	32
1.7.3	Hyperelastic	33
1.8	Exemplar Methods	34
1.8.1	FNIRT	34

1.8.2	Demons	35
1.8.3	ANTs	36
1.8.4	DARTEL	36
1.8.5	Geodesic Shooting	36
1.8.6	NiftyReg	37
1.8.7	Diffusion Methods	37
1.9	General Purpose Graphics Processing Units	37
1.9.1	Hardware Terms	38
1.9.2	Execution Coordination Terms	39
1.9.3	Performance Terms	41

1.1 Introduction

The central role image registration plays in neuroscience analysis pipelines is, by now, firmly established. At the single subject level it allows us to perform tasks such as propagating known anatomical labels from a template space to an individual subject, while at the group level it allows for robust and powerful statistical inference when considering differences in functional and structural features between populations. In all cases, the quality of the alignment achieved will greatly impact the accuracy and power of any analysis following on from the registration step.

At the most superficial level, you can think of image registration as a method of maximising the visual similarity between two images. However, upon thinking about it a bit more, it should quickly become apparent that visual similarity is simply a by-product of what it is you are truly trying to do. In fact, the easiest way to make two images *look* the same is to replace one of them with the other - but even someone with absolutely no understanding of neuroscience could tell you that this defeats the point of the exercise. So what is it then that registration really tries to achieve? It is, in fact, not a transformed image, but a one-to-one map which relates every point in one image to a unique point in the other, and vice versa.

When viewed in this light, the complexity of the problem may appear somewhat overwhelming. It is a task which even the most experienced neuroscientist would find difficult to perform manually, when you consider that an image of a brain

often contains millions of voxels, many of which appear to be nearly identical to their neighbours.

There exists then an inherent ambiguity in any such mapping. Consider the case where a few uniquely corresponding regions in two brains have been identified. You now have anchor points for this exact mapping, but what should be done about those areas in-between? For regions in close proximity to the known points of correspondence you may assume that they should remain nearby, but it becomes increasingly unclear what to do about points as you move further and further away. You may then counter this by increasing the number of corresponding anatomical locations, or by including some problem specific prior information about what is likely to occur between known points of correspondence. Every piece of information added in this way helps bring you closer to the true mapping.

In this thesis I seek to combine the rich and varied information provided by a range of MRI imaging modalities in order to improve the anatomical correctness of image registration results. Advances in computing power, particularly in the form of massively parallelised GPU processors, have allowed me to explore complex models of how brains deform that would otherwise have been computationally intractable. In particular, I focus on simultaneously finding the optimal mapping of the T1-weighted and DTI images from one subject to another as well as ensuring that areas with ambiguous mappings are transformed in a manner corresponding to a maximally plausible set of prior beliefs. In order to achieve this I have developed an entirely new software tool, the MultiMOdal Registration Framework (MMORF), which is instrumental in facilitating the research presented here.

1.1.1 History of Image Registration

Image registration is a well established technique, with over 50 years of research and development. Medical images have always been natural candidates for the application of image registration, and as such many of the advancements in image registration have coincided with advancements in medical imaging techniques.

Within individuals, image registration allows changes in shape, size, pose or contrast between time points to be determined. From this, subject movement can be corrected, tissue atrophy can be measured and contrast agents can be traced. Image registration can also be used to find the correspondence between multiple different imaging modalities in the same subject (so-called modality fusion), allowing for anatomically specific multivariate analyses which have the potential to be more informative than a simple mass univariate analysis might be.

Between individuals, image registration allows accurate anatomical correspondences to be determined, usually through the use of a template or atlas. This can be used to locate and characterise differences between sub-populations (such as patients and healthy controls), or automatically identify structures within an individual using known population average maps of that structure's anatomical location. Much of the development in this area has been driven by a desire to study the brain, which displays a large degree of inter-subject shape variation, particularly in the folding patterns of the cortex.

One of the earliest examples of image registration was carried out by Roberts (1965), not in the field of medical imaging, but rather of machine perception. A simple linear transform, consisting of translation, scaling and rotation was used to match a 2D image of a planar-surfaced polygon to a library of 3D models of known structures. This technique was limited to objects where any differences between the models and the objects being imaged could be accounted for by a linear operation.

But not all images are simple linear transformations of model objects. Fischler and Elschlager (1973) showed how this could be extended to objects where features within those objects can move nonlinearly with respect to each other. They applied their method to detecting faces from the relative position of features such as eyes, ears, nose and mouth.

Early applications of registration in the medical imaging field began appearing in the late 1970s and early 1980s. Singh et al. (1979), for example, used 2D affine registration of Thallium-201 nuclear images of the heart before and after activity to measure differences in perfusion of the myocardium. This was still a

semi-automated method, requiring manual identification of landmarks to initialise the registration, followed by an automated fine-tuning using an intensity based image similarity metric.

Bohm et al. (1983), described how an atlas of labelled brain regions, derived from a digitised photograph of a post-mortem brain, could be used to identify structures not visible in a computed tomography (CT) or positron emission tomography (PET) image in order to guide stereotaxic surgery or define regions of interest for metabolic activity analysis, by linearly deforming the atlas. Bajcsy et al. (1983) went on to show a similar but more complex method, where an individual subject's CT image was matched to a CT atlas using nonlinear registration.

Up until this point, much of the development of automated registration techniques was fairly ad hoc. Then, in the late 1980s, Witkin et al. (1987) formally described the general image registration problem in its modern recognisable format - as an iterative energy minimisation problem using a combination of image similarity and smoothness (regularisation) terms.

Much of the work in the 1990s was focussed on new information theoretic approaches to determining image similarity, based on Shannon entropy (Shannon 1948). Methods such as normalised mutual information (NMI) by Studholme et al. (1999) greatly improved the quality of inter-modal, intra-subject registration - opening up the field of multimodal data fusion techniques.

With rising computing power, and decreasing computing cost, the 2000s and early 2010s saw a resurgence in computationally intensive intra-modal, nonlinear registration methods. Very high dimensional deformations could now be estimated on reasonable timescales, and more complex image types (such as vector and tensor valued images) could be used to drive the registration, rather than simple scalar images only. A number of example methods from this period are presented in Section 1.8, and some of these still constitute the current state-of-the-art for registration accuracy.

In the last 5 years or so, the increasing popularity of deep learning, and in particular the rise of convolutional neural networks, has led to the development

of deep learning based registration methods. These range from fully supervised methods, such as Cao et al. (2017) who use the outputs of classical registration methods to train the networks, to unsupervised methods, such as Balakrishnan et al. (2018) who learn to optimise a similarity metric between images directly from the images themselves in a single step. These methods are appealing, as much of the computational burden is shifted from running a registration to training the network, resulting in orders of magnitude speed up versus classical optimisation based registration methods. However, current implementations are still mostly limited to unimodal registration, and while some of these methods when trained on a specific dataset can outperform classical registration tools, they still suffer from poor generalisability across datasets (Haskins et al. 2020). It is likely that in the future, though, we may see deep learning based image registration methods overtake their classical counterparts and become more widely adopted by medical imaging end-users such as neuroimaging researchers.

1.1.2 Why We Still Need MMORF

Given the long history of image registration, and the current trend towards deep learning based methods, it is fair to ask whether there is a need for a method such as MMORF.

The first reason that MMORF was proposed is one of practicality rather than technicality. MMORF has been designed to work within the FMRIB Software Library (FSL) (Jenkinson et al. 2012), which is a suite of tools for the end-to-end analysis of neuroimaging data from basic preprocessing through to statistical inference. An important component of that pipeline is a nonlinear registration tool. The current method used in FSL is FNIRT (Andersson et al. 2007; Andersson et al. 2019), which is only able to register unimodal scalar image pairs. The increasing use of FSL as the primary software for analysing large, multimodal neuroimaging studies such as the Human Connectome Project (HCP) (Van Essen et al. 2012; Van Essen et al. 2013; Glasser et al. 2013), and UK Biobank (UKB) (K. L. Miller et al. 2016) meant that a need was identified for a tool which would

both fit into the FSL ecosystem and was capable of dealing with both scalar and tensor valued images in a unified manner.

The second reason for MMORF’s development was to try and leverage potential advantages that *simultaneously* aligning scalar and tensor valued images might offer. Multiple different software tools are usually required to align the various modalities acquired in modern neuroimaging studies. This means that either only a single modality is used to perform alignment between subjects and it is assumed that this modality correctly maps all other modalities with it, or that a separate registration must be run for each modality independently. The result of this is that there is a potential reduction in accuracy of the anatomical correspondence between subjects, and most analyses using multimodal data are constrained to analyse each modality in isolation. In the UKB, one of the major goals is to derive biomarkers capable of predicting disease before the onset of symptoms. One source of potential biomarkers are imaging derived phenotypes (IDPs). IDPs are summary measures normally defined within some region of interest (such as the total volume of grey matter within the brain) which can be correlated with non-imaging biomarkers (such as genetic sequences) or patient outcomes (such as the development of dementia), and each IDP tends to be calculated from a single imaging modality. By having all modalities for a subject accurately aligned to the same template space, more complex multivariate IDPs can be developed and tested, with the hopes that they might have greater predictive power than their univariate counterparts.

The third reason for developing MMORF from scratch is because of the performance benefits of developing code to be run on graphics processing units (GPUs). GPUs can perform thousands of calculations simultaneously, and are ideally suited to high-dimensional optimisation problems such as image registration. Simultaneously aligning multiple modalities is computationally far more burdensome than unimodal registration - particularly when aligning tensor-valued images which are affected by image rotations rather than just translations. Therefore the benefits of improved runtimes are worth the extra development effort required when coding for GPUs. This will allow MMORF to be usable on large population-level datasets,

like UKB, with upwards of 100,000 individuals being scanned. This is even more important as increases in the magnetic field strength and imaging gradient strengths push the resolution of MRI scans ever higher. An added advantage of using GPUs is that MMORF is able to use more complex (and therefore more computationally taxing) models to constrain (or regularise) deformations - which has led to MMORF bettering state-of-the-art methods even for unimodal registration (F. J. Lange et al. 2020a).

Lastly, when deep-learning methods begin to implement multimodal registration, MMORF can provide an important classical registration benchmark against which to compare.

With all this in mind, I believe now is the perfect time to introduce a new, state-of-the-art, classical registration tool. MMORF is able to take advantage of the rich history of methods development that has come before it, and will hopefully stand as the benchmark against which to compare future advances in deep learning based multimodal image registration.

1.2 MRI Modalities

The description of the MRI signal and associated image contrasts are adapted from Brown et al. (2014).

Magnetic resonance imaging (MRI) is a non-invasive medical imaging method which combines a strong, constant magnetic field (B_0), weaker variable magnetic fields (gradients) and radio frequency (RF) transmission and reception fields/coils ($B_1^{transmit}$ and $B_1^{receive}$) to generate images without the need for ionising radiation.

For neuroimaging purposes, MRI scanners are typically based around a large toroidal magnet with an approximately homogeneous magnetic field of between 1.5 and 7 Tesla (T) at the centre of the bore. This is termed the B_0 field. When placed in the scanner, the magnetic spin of Hydrogen nuclei (protons) align with the B_0 field and precess at an angular frequency of:

$$\omega_0 = \gamma B_0, \tag{1.1}$$

where ω_0 is known as the the Larmor frequency, and $\gamma = 42.6 \text{ MHz T}^{-1}$. At equilibrium, the magnitude of the average magnetic moment in the direction of B_0 (z -direction) is M_0 and is determined by the proton density and the magnitude B_0 .

If a $B_1^{transmit}$ RF pulse is applied at the Larmor frequency, then the proton spins can be tipped out of alignment with B_0 (excited) and into the plane perpendicular to B_0 . The precession of the perpendicular proton spins creates an oscillating magnetic field which then induces an electrical signal in a $B_1^{receive}$ receive RF coil.

Gradient coils are used to create a small, linearly spatially varying magnetic field causing the precession frequency of the spins to similarly spatially vary. The application of such a field following the RF pulse will cause the spins to linearly de-phase along the direction of varying field. When applied correctly, the subsequent received signal represents the Fourier transform of a 1D projection of the object in the scanner in the direction of the applied gradient. By combining multiple gradient directions, and multiple echo acquisitions, a full 3D image of the object in the scanner can be reconstructed using the inverse Fourier transform.

If an RF pulse tips the spins by 90° , then the magnetisation in the z -direction will recover according to:

$$M_z(t) = M_0(1 - e^{-t/T_1}), \quad (1.2)$$

with T_1 a tissue-specific time constant. If the magnetization is not allowed to recover back to equilibrium before another RF pulse is applied, then the amount of magnetization tipped into the transverse plane is similarly reduced. The time between RF excitations is called the repetition time (T_R). For a given repetition time, the instantaneous magnetisation in the transverse plane at time T_E (echo time) following the RF pulse is given by:

$$M_\perp(T_E) = M_0(1 - e^{-T_R/T_1})e^{-T_E/T_2}. \quad (1.3)$$

From this it can be seen that the transverse magnetisation decay is controlled by another tissue specific time constant T_2 . Note that this equation assumes a spin-echo imaging sequence where a second 180° RF pulse has been used to refocus

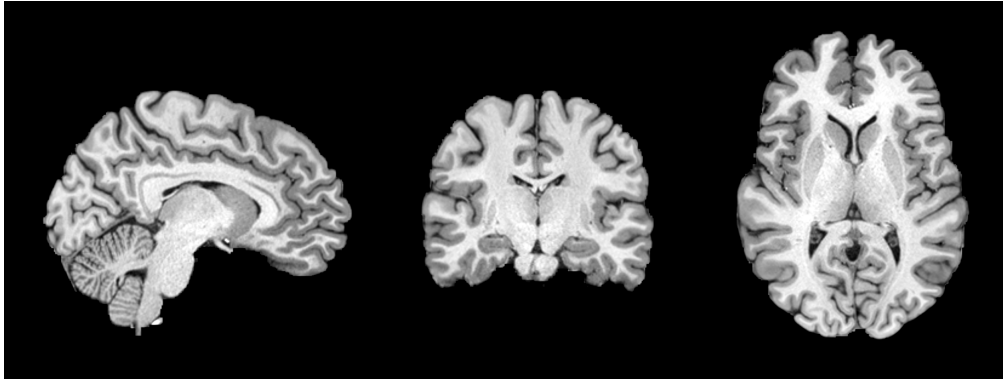


Figure 1.1: T1-weighted image - Note that white matter appears brighter than grey matter, which appears brighter than CSF.

spins which have dephased due to non-random magnetic field inhomogeneities. If that were not the case, then T_2 would need to be replaced with $T_2^* < T_2$. For the remainder of this section I will assume that M_0 is approximately equal for all tissue types. T1-weighted, T2-weighted and Diffusion-weighted images each provide different information about tissue type, and can therefore be used to generate images with different signal contrasts, which can later provide unique information for image registration.

1.2.1 T1-Weighted Imaging

From Equation 1.2 it can be seen that choosing $T_E \ll T_2$ and $T_R < T_1$ will cause the transverse magnetisation, and therefore the received signal, to be largely determined (or weighted) by T_1 . In this case the image contrast will be based on each tissue type's T_1 value, with tissues that have a shorter T_1 appearing brighter. In the brain, white matter has the shortest T_1 , followed by grey matter and then cerebrospinal fluid (CSF). Therefore, white matter appears brighter than grey matter, which appears brighter than CSF - as shown in Figure 1.1.

1.2.2 T2-Weighted Imaging

Again, from Equation 1.2 it can be seen that choosing $T_E \approx T_2$ and $T_R \gg T_1$ will cause the received signal to be weighted by T_2 . Therefore the image contrast will be based on each tissue type's T_2 value, with tissues that have a longer T_2

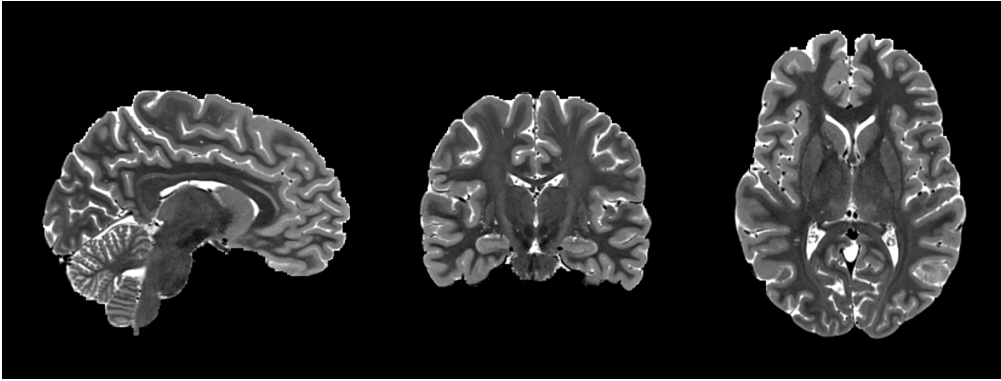


Figure 1.2: T2-weighted image - Note that CSF appears brighter than grey matter, which appears brighter than white matter.

appearing brighter. In the brain, CSF has the longest T_2 , followed by grey matter and then white matter. Therefore, CSF appears brighter than grey matter, which appears brighter than white matter - as shown in Figure 1.2.

1.2.3 Diffusion-Weighted Imaging

If, prior to applying the magnetic field gradients during image readout, two identical diffusion encoding gradients separated by a 180° RF pulse are applied, this will cause the spins to first de-phase (in a manner which spatially varies in the direction of the diffusion encoding gradient) and then re-phase. In the ideal case, the net change in phase for each proton will be zero, and this will then have no effect on the signal at all. However, if the spins are randomly diffusing in the same direction as the gradient orientation, then some of the spins will be at different positions for each diffusion encoding gradient, and the phase loss and accrual will not cancel resulting in a net loss of phase coherence and therefore signal strength. This is known as diffusion weighting. For Gaussian diffusion, the signal is changed from the non-diffusion weighted signal S_0 by:

$$S_D = S_0 e^{-bD}, \quad (1.4)$$

where D is the diffusivity, and b is proportional to the square of the diffusion gradient strength and duration, and the time between the gradients. By applying diffusion encoding gradients in many different directions, a model of how diffusion

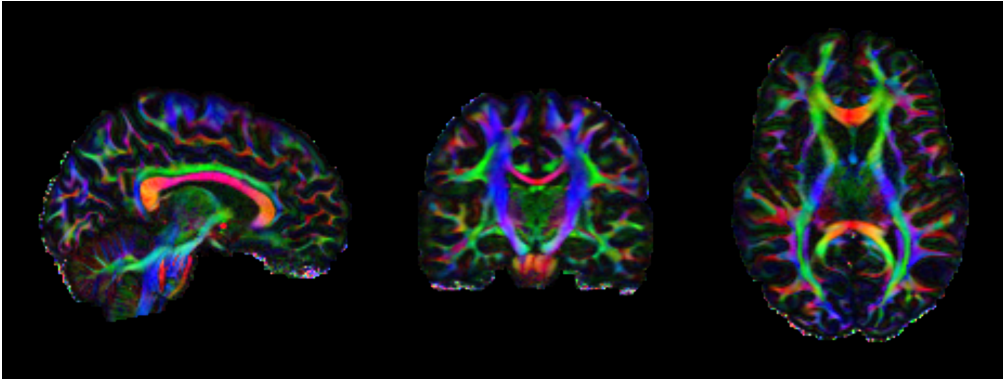


Figure 1.3: Diffusion-derived image - The image intensity indicates how strongly water has a preferred direction of diffusion at each position. The colour indicates the primary direction of preferred diffusion with red representing left-right, green representing anterior-posterior, and blue representing superior-inferior.

varies with orientation in each voxel can be determined (see Section 4.3 for one such model, the diffusion tensor). In areas where the tissue microstructure causes spins to have a preferred direction of diffusion (e.g., parallel to axons in white matter) then this provides orientation dependent information about diffusivity in the tissue. In areas where there is no preferred direction of diffusion, this still provides information about how freely spins are able to diffuse on average (e.g., more in CSF than in grey matter).

Figure 1.3 shows an example of where multiple diffusion-weighted images have been used to model the preferred direction of diffusion within the brain. Note how there is clearly more of a preferred direction in the white matter than in grey matter or CSF, and that the preferred direction tends to be parallel to white matter tracts (e.g., the red corpus callosum indicating a left-right preference).

1.3 What is Image Registration?

This section presents the overarching concept of image registration, and sets up the foundational knowledge required to understand the remainder of the thesis. It explores the concept of image registration as a domain mapping problem, before describing the two largest classes of image registration problems, namely *Linear* and *Nonlinear* registration.

But first, let us consider what an image actually is. I will define an image f as a continuous function measuring some property Ψ of an object defined on a domain Ω . That is:

$$f : \Omega \rightarrow \Psi, \quad (1.5)$$

where

$$\Omega \subseteq \mathbb{R}^d \quad (1.6)$$

$$\Psi \subseteq \mathbb{R}^p \quad (1.7)$$

When $p = 1$, f is referred to as a scalar image. For $p > 1$, f most often represents a multi-channel (e.g. an RGB colour), vector (e.g. velocity field) or tensor (e.g. material strain field) image.

1.3.1 Image Registration as Domain Mapping

Assume there exist two images f and g . If they are images of the same object, then for every point $\mathbf{x} \in \Omega$, there exists a point $\mathbf{x}' \in \Omega$ such that $f(\mathbf{x}) = g(\mathbf{x}')$. One can therefore define the problem of image registration as finding the mapping, or transformation, ϕ for which:

$$\mathbf{x}' = \phi(\mathbf{x}) \quad (1.8)$$

where

$$\phi : \Omega \rightarrow \Omega \quad (1.9)$$

In general though, f and g do not have to represent the same object - and in medical imaging they regularly do not. In this case, the image registration problem becomes one of finding the transformation ϕ which maps the images such that they are as similar as possible. The measure of this similarity is something which is discussed in later sections.

The following subsections cover two important families of transformations for linear and nonlinear registration respectively.

1.3.2 Linear Image Registration

Linear image registration restricts the transformation ϕ to the family of affine matrices $\mathbf{A}$, which include translation, scaling, rotation, shear or any combination of the above. That is:

$$\phi_A(\mathbf{x}) = \mathbf{A}\mathbf{x} \quad (1.10)$$

You may note that for $\mathbf{x} \in \mathbb{R}^d$, the above formulation cannot perform a translation. An “augmented” form of $\mathbf{x}$ is therefore commonly used, with $\mathbf{x}_{aug} = [\mathbf{x}^\top, 1]^\top$, where $\mathbf{x}$ is a column vector of length d .

When images f and g represent the same object (e.g., two scans of the same person’s brain) then an affine transformation is often the most appropriate way to model the transformation. In fact, in the absence of any deformation of the object it is sufficient to include only translations and rotations in $\mathbf{A}$. This is termed a *rigid body* deformation.

However, when f and g represent different objects (e.g., scans of two different people’s brains) then the additional degrees of freedom offered by scaling and shearing can lead to better alignment.

1.3.3 Nonlinear Image Registration

Nonlinear image registration takes a more flexible approach to the transformation ϕ , in that the mapping is no longer required to be the same globally. A convenient way of achieving this is to represent ϕ as the sum of the original position $\mathbf{x}$ and a displacement vector $\mathbf{d}$ which varies as a function of $\mathbf{x}$. Additionally, how $\mathbf{d}$ varies may be controlled using a set of parameters $\mathbf{w}$. That is:

$$\phi_{nonlinear}(\mathbf{x}, \mathbf{w}) = \mathbf{x} + \mathbf{d}(\mathbf{x}, \mathbf{w}) \quad (1.11)$$

1.4 Nonlinear Deformation Models

One of the key features which distinguishes the different methods of estimating ϕ is how the deformation is modelled and parametrised. This section presents an overview of the most common models.

1.4.1 Small Deformation Framework

The small deformation framework's distinguishing characteristic is that it estimates the displacement field $\mathbf{d}$ directly (Bajcsy and Kovačič 1989; M. Miller et al. 1997). The simplest version of this is a non-parametric, completely free-form deformation. In this case $\mathbf{d}$ is defined independently at each point $\mathbf{x}$. Whilst this representation is very flexible, it is also poorly constrained. Furthermore, a completely free-form deformation $\mathbf{d}$ is discrete, and is therefore not defined between sample points in Ω .

Parametric representations of $\mathbf{d}$ overcome some of the limitations of non-parametric small deformation methods by explicitly defining the allowable modes of variation of displacements (Sederberg and Parry 1986; Rueckert et al. 1999; Ashburner and Friston 1999). In general these methods use some set of weighted basis functions. That is:

$$\mathbf{d}(\mathbf{x}, \mathbf{w}) = \mathbf{w}^\top \mathbf{b}(\mathbf{x}) \quad (1.12)$$

where $\mathbf{w}$ is a vector of weights, and $\mathbf{b}(\mathbf{x})$ is a vector of basis functions evaluated at $\mathbf{x}$.

A distinguishing characteristic of parametric displacement fields is whether the basis functions have global or local support - that is whether the basis functions take on non-zero values over an infinite or a finite region of Ω . Discrete cosine transforms are an example of basis functions with global support due to their infinite oscillatory behaviour (Ashburner and Friston 1999). B-splines, on the other hand, are an example of basis functions with finite support as each spline is only non-zero within a finite radius of the centre of its support (or “knot”) (Rueckert et al. 1999; Andersson et al. 2007).

1.4.1.1 B-splines

It is worth expanding slightly on the topic of B-splines (Unser et al. 1993a; Unser et al. 1993b) as they form an important part of this work. The “B” in B-splines refers to “Basis”. A piecewise polynomial can be described using a weighted set of B-splines as a basis. In all cases here I will assume that the basis functions are positioned on a regular grid with a constant knot spacing. An element of such a basis is defined as $\beta_m^n(t)$, where $m \in \{0, 1, \dots, M\}$ is the element number, $n \in \{0, 1, \dots\}$ is the polynomial order, and t is the position measured in knot spacings from the centre of spline β_0^n .

The 0th order B-spline in 1D is defined as follows:

$$\beta_m^0(t) = \begin{cases} 0, & \text{for } |t - m| \geq \frac{1}{2} \\ 1, & \text{for } |t - m| < \frac{1}{2} \end{cases}, \quad (1.13)$$

which, for $m = 0$ is a rectangular function with height and length 1, centred at the origin of t . All higher orders of B-spline are then defined by recursive convolution, such that:

$$\beta_m^n(t) = \beta_m^{n-1} * \beta_m^0(t) \quad (1.14)$$

$$= \underbrace{\beta_m^0 * \beta_m^0 * \dots * \beta_m^0}_{n+1 \text{ times}}(t). \quad (1.15)$$

From this it can be deduced that each spline has a non-zero support of exactly $n + 1$ knots, and integrates to 1. 3rd order (cubic) B-splines are of particular interest as they provide a good balance between the complexity of the deformations they can describe, and the extent of their support, and are described by (Ruijters and Thevenaz 2012):

$$\beta_m^3(t) = \begin{cases} 0, & \text{for } 2 \leq |t - m| \\ \frac{1}{6}(2 - |t - m|)^3, & \text{for } 1 \leq |t - m| < 2 \\ \frac{2}{3} - \frac{1}{2}|t - m|^2(2 - |t - m|), & \text{for } |t - m| < 1 \end{cases}. \quad (1.16)$$

A B-spline parametrised function f is then defined as:

$$f(t) = \sum_{m=0}^M c_m \beta_m^n(t), \quad (1.17)$$

where c_m is a weighting coefficient.

B-splines can be easily extended to higher dimensions, as an ND spline is simply the tensor product of N 1D splines. A 3D function with $K \times L \times M$ B-splines is defined as:

$$f(t_k, t_l, t_m) = \sum_{k=0}^K \sum_{l=0}^L \sum_{m=0}^M c_{k,l,m} \beta_k^n(t_k) \beta_l^n(t_l) \beta_m^n(t_m) \quad (1.18)$$

1.4.2 Large Deformation Models

The large deformation framework is broadly characterised by estimating a velocity (or flow) field which governs how the reference image is deformed to match the moving image (Bro-Nielsen and Gramkow 1996; Christensen et al. 1996; M. Miller et al. 1997). The displacement field $\mathbf{d}$ is then calculated through integration of this velocity field over time.

1.4.2.1 Lagrangian and Eulerian Frames of Reference

Flow fields can be described using two different but related definitions depending on the observer's frame of reference. The Lagrangian specification describes the flow field at time t as the location of an observer who began at position $\mathbf{x}_0$ when $t = 0$, and is being transported along by the field. The Eulerian specification describes the flow field at time t as the velocity measured by an observer who remains at position $\mathbf{x}$ at all times.

Formally the Lagrangian field $\mathbf{L}_f$ is defined as:

$$\mathbf{L}_f(\mathbf{x}_0, t) = \mathbf{x}_0 + \mathbf{d}(\mathbf{x}_0, t) \quad (1.19)$$

and the Eulerian field $\mathbf{E}_f$ is defined as:

$$\mathbf{E}_f(\mathbf{x}, t) = \mathbf{v}(\mathbf{x}, t) \quad (1.20)$$

The two definitions are related as follows:

$$\mathbf{v}(\mathbf{x}_0 + \mathbf{d}(\mathbf{x}_0, t), t) = \frac{\partial \mathbf{d}(\mathbf{x}_0, t)}{\partial t} \quad (1.21)$$

The Lagrangian definition is perhaps the more intuitive, and if the time dependence is constrained to be linear then it is equivalent to the small deformation framework. However, for non-linear time dependent flow fields, it can become challenging to work with. This is due to the fact that at any time beyond $t = 0$ the flow is no longer defined on a uniform grid.

In contrast, the Eulerian definition can easily be defined on a regular grid for all time. Practically, most methods employ the Eulerian definition, and solve the partial differential equation in Equation 1.21 to convert to the Lagrangian definition when, for example, resampling the moving image.

1.4.2.2 Stationary vs Time-varying Velocity Fields

Consider the Eulerian velocity field definition in Equation 1.19. If the field is constant with respect to time, then $\mathbf{E}_f$ describes a stationary velocity field - that is, a field which varies across space but not time. Such fields can greatly simplify both finding and applying velocity fields to images, but they do reduce the possible space of all warps (Ashburner 2007). In contrast, time-varying velocity fields can represent a much richer variety of deformations, but at increased computational complexity (Beg et al. 2005).

1.4.3 Diffeomorphism

One of the most often discussed characteristics of warps is whether or not they are diffeomorphisms. Consider two images $f(\mathbf{x}_f)$ and $g(\mathbf{x}_g)$, each defined over a subset of three dimensional euclidean space, such that:

$$\mathbf{x}_f \in A \subset \mathbb{R}^3 \quad (1.22)$$

$$\mathbf{x}_g \in B \subset \mathbb{R}^3 \quad (1.23)$$

A diffeomorphic transformation $\phi_{diff} : A \rightarrow B$ is defined as a smooth, bijective mapping between the manifolds A and B . Bijection implies a one-to-one mapping, such that each point in A maps to a unique point in B (injection), and that each point in A is reachable from some point in B (surjection). Smoothness implies that both ϕ_{diff} and its inverse $\phi_{diff}^{-1} : B \rightarrow A$ are differentiable.

In practical terms, diffeomorphism defines a set of criteria which, when fulfilled, guarantee that the transformation will not result in any folds or tears in the image. In other words, they preserve the topology of the original images. It seems like a reasonable assumption that there exists a topology preserving mapping between the anatomy and function of brains from different subjects. It also ensures invertibility, which is a very useful property for many applications.

It is much simpler to enforce diffeomorphism in the large deformation framework. As integration of a velocity field is essentially the composition of many small displacement fields, then, provided that each small displacement is diffeomorphic, the composition will be too. In this sense, the large deformation framework can be seen as diffeomorphic *by design*.

This does not mean that small deformation framework transformations *cannot* be diffeomorphic, merely that they *allow* non-diffeomorphic transformations and must therefore be constrained to be diffeomorphic by other means (Rohlfing et al. 2003).

1.5 Image Registration as Optimisation

Once a model of image transformation $\phi(\mathbf{x}, \mathbf{w})$ has been settled on, the process of performing image registration can be seen as an optimisation problem over the parameters $\mathbf{w}$ of the transformation (where $\mathbf{w}$ is a column vector of length M). Performing the optimisation requires some objective function $\mathcal{O}$ to optimise. One component of the objective function is invariably a measure of image similarity (which one seeks to maximise) or dissimilarity (which one seeks to minimise). That is, one seeks either:

$$\arg \max_{\mathbf{w}} \mathcal{O}(\mathbf{w}) \quad (1.24)$$

or:

$$\arg \min_{\mathbf{w}} \mathcal{O}(\mathbf{w}) \quad (1.25)$$

A maximisation problem can be easily converted into a minimisation problem through negation. I will therefore only consider minimisation from this point on, and replace the objective function $\mathcal{O}$ with a cost function $\mathcal{C}$.

The majority of cost functions associated with image registration are neither *linear* nor *convex*. Nonlinearity means that there is rarely a closed form solution for finding the minimum, and non-convex means that there are parts of the domain from which the global minimum is non-trivial to reach.

1.5.1 Nonlinear Optimisation

Nonlinear optimisation is concerned with cost functions which are nonlinear with respect to their parameters. A consequence of this is that the minimum may occur either at or within a constraint boundary - and therefore constraints may not be necessary to find a meaningful minimum. Furthermore, local optima are not guaranteed to be global optima.

A special case of nonlinear optimisation is when the cost function is convex. That is:

$$\mathcal{C}(\alpha \mathbf{w}_1 + (1 - \alpha) \mathbf{w}_2) \leq \alpha \mathcal{C}(\mathbf{w}_1) + (1 - \alpha) \mathcal{C}(\mathbf{w}_2) \quad \forall \alpha \in [0, 1] \quad (1.26)$$

In such cases any local optimum is also a global optimum.

The majority of this work focuses on nonlinear problems which are neither linear, nor convex. Most methods of minimising these functions find only local minima, and they are therefore sensitive to the region in which the search for these minima is initialised.

Registration cost functions do not generally have closed form solutions to their minima. The process of minimising them is therefore to iteratively evaluate them for different parameters $\mathbf{w}$ until a minimum is found. In principle this can be done

via an exhaustive “brute force” evaluation of all possible parameter combinations, however for all but the most trivial of cases this is computationally prohibitive. More principled approaches to choosing which parameter combinations to evaluate represents the core distinguishing feature of the optimisation algorithms which I describe next. Normally an initial guess $\mathbf{w}_0$ of the parameters is made, and a number of adjustments or “steps” $\Delta\mathbf{w}$ are taken such that $\mathbf{w}_{k+1} = \mathbf{w}_k + \Delta\mathbf{w}$. In general, $\Delta\mathbf{w} = \alpha\mathbf{p}$, where $\mathbf{p}$ represents a search *direction* and α represents a *step size*.

1.5.2 Gradient-based Methods

The simplest, at least conceptually, family of optimisation algorithms are those based on the gradient of the cost function $\nabla\mathcal{C}$, where:

$$\nabla\mathcal{C}(\mathbf{w}) = \frac{\partial\mathcal{C}}{\partial\mathbf{w}} \tag{1.27}$$

$$= \left[\frac{\partial\mathcal{C}}{\partial w_1}, \frac{\partial\mathcal{C}}{\partial w_1}, \dots, \frac{\partial\mathcal{C}}{\partial w_M} \right]^\top. \tag{1.28}$$

Except when $\nabla\mathcal{C} = \mathbf{0}$, the negative gradient always provides a direction in which the cost function decreases, therefore:

$$\mathcal{C}(\mathbf{w} + \Delta\mathbf{w}) < \mathcal{C}(\mathbf{w}), \tag{1.29}$$

where:

$$\Delta\mathbf{w} = -\alpha\nabla\mathcal{C}(\mathbf{w}), \tag{1.30}$$

for α positive and sufficiently close to 0.

1.5.2.1 Steepest Descent

The steepest descent algorithm (often simply referred to as gradient descent) chooses the search direction $\mathbf{p}$ to be $-\nabla\mathcal{C}(\mathbf{w})$ (Nocedal and Wright 2006, Ch.3, pp 30). There are two common approaches for choosing α . The simplest is to simply use a fixed value for α at each iteration. Setting α too large can lead to oscillatory behaviour preventing convergence, whereas setting α too small can lead to many

steps needing to be taken before convergence. An optimal α can be chosen through a line search in the direction $-\nabla\mathcal{C}(\mathbf{w})$ such that:

$$\alpha_{optimal} = \arg \min_{\alpha} \mathcal{C}(\mathbf{w} - \alpha \nabla\mathcal{C}(\mathbf{w})), \quad (1.31)$$

however this can be slow, particularly if evaluating $\mathcal{C}(\mathbf{w})$ is computationally costly.

1.5.2.2 Conjugate Gradient

A drawback of the line search steepest descent method is that sequential search directions are always perpendicular to each other. Taking a step in a new direction therefore tends to “undo” some of the optimisation achieved in the previous step. When $\mathbf{w}$ is 2 dimensional this behaviour is obvious and manifests as zigzagging through parameter space, with $\nabla\mathcal{C}(\mathbf{w}_k) = \lambda \nabla\mathcal{C}(\mathbf{w}_{k+2})$ where λ is a scalar. A way to overcome this is then to rather define the search direction $\mathbf{p}$ as:

$$\mathbf{p}_k = -(\nabla\mathcal{C}(\mathbf{w}_k) + \beta_k \mathbf{p}_{k-1}) \quad (1.32)$$

, where β can be thought of as a momentum. Therefore:

$$\Delta\mathbf{w} = -\alpha(\nabla\mathcal{C}(\mathbf{w}_k) + \beta_k \mathbf{p}_{k-1}) \quad (1.33)$$

A common way of choosing β comes from the nonlinear extension of the conjugate gradient method for linear systems of equations. The Fletcher-Reeves algorithm is of this type and defines (Nocedal and Wright 2006, Ch.5, pp 121):

$$\beta_k = \frac{\nabla\mathcal{C}(\mathbf{w}_k)^\top \nabla\mathcal{C}(\mathbf{w}_k)}{\nabla\mathcal{C}(\mathbf{w}_{k-1})^\top \nabla\mathcal{C}(\mathbf{w}_{k-1})}. \quad (1.34)$$

The conjugate gradient algorithm is often a more effective choice of gradient optimisation scheme than steepest descent, requiring fewer optimisation steps without adding too much computational overhead.

1.5.3 Newton Methods

Newton methods work by approximating the true cost function by a quadratic approximation around the current best guess of $\mathbf{w}$ (Nocedal and Wright 2006, Ch.3 pp 44). It can be derived using the second order Taylor expansion as follows:

$$\mathcal{C}(\mathbf{w}_{k+1}) = \mathcal{C}(\mathbf{w}_k + \Delta\mathbf{w}) \quad (1.35)$$

$$\approx \mathcal{C}(\mathbf{w}_k) + \nabla\mathcal{C}(\mathbf{w}_k)^\top \Delta\mathbf{w} + \frac{1}{2} \Delta\mathbf{w}^\top \mathbf{H}(\mathbf{w}_k) \Delta\mathbf{w} \quad (1.36)$$

Note that:

$$\arg \min_{\Delta\mathbf{w}} \mathcal{C}(\mathbf{w}_k + \Delta\mathbf{w}) \quad (1.37)$$

$$\implies \nabla\mathcal{C}(\mathbf{w}_k + \Delta\mathbf{w}) = 0 \quad (1.38)$$

And therefore:

$$\nabla\mathcal{C}(\mathbf{w}_k) + \mathbf{H}(\mathbf{w}_k) \Delta\mathbf{w} = 0 \quad (1.39)$$

$$\Delta\mathbf{w} = -\mathbf{H}^{-1}(\mathbf{w}_k) \nabla\mathcal{C}(\mathbf{w}_k) \quad (1.40)$$

Notice how $\mathbf{H}^{-1}$ in Equation 1.40 performs the actions of both the change in search direction away from the steepest gradient, and the scaling in a single step - compared to two separate steps of the conjugate gradient method in Equation 1.33.

A potential pitfall of Newton's method is that if the best quadratic fit to the cost function locally is concave, then it is possible for $\Delta\mathbf{w}$ to actually step *uphill*. The methods which follow help to address this issue.

1.5.3.1 Gauss-Newton

The Gauss-Newton (Nocedal and Wright 2006, Ch.10, pp 254) variant of Newton's method is very useful for cost functions which can be written in the form:

$$\mathcal{C}(\mathbf{w}) = \frac{1}{2} \sum_{i=1}^N y_i^2(\mathbf{w}) \quad (1.41)$$

The gradient is then:

$$\nabla\mathcal{C}(\mathbf{w}) = \sum_{i=1}^N y_i(\mathbf{w}) \frac{\partial y_i}{\partial \mathbf{w}} \quad (1.42)$$

And the Newton Hessian is:

$$\mathbf{H}(\mathbf{w}) = \sum_{i=1}^N \frac{\partial y_i}{\partial \mathbf{w}} \frac{\partial y_i}{\partial \mathbf{w}} + y_i(\mathbf{w}) \frac{\partial^2 y_i}{\partial \mathbf{w}^2} \quad (1.43)$$

The Gauss-Newton method approximates the Hessian by dropping the mixed partial derivative term, such that:

$$\mathbf{H}(\mathbf{w})_{GN} = \sum_{i=1}^N \frac{\partial y_i}{\partial \mathbf{w}} \frac{\partial y_i}{\partial \mathbf{w}}. \quad (1.44)$$

Gauss Newton optimisation tends to have the best convergence properties when applied to functions where $y_i(\mathbf{w}) \approx 0 \forall i$ at the optimum, as this implies that $\sum_{i=1}^N y_i(\mathbf{w}) \frac{\partial^2 y_i}{\partial \mathbf{w}^2} \approx 0$ and therefore $\mathbf{H}_{GN} \approx \mathbf{H}$. When applied to image registration using a mean squared error similarity metric (see Section 1.6.1.1), y_i represents the difference in intensities between voxels in an image which, when correctly aligned, should be approximately zero and lead to good convergence properties.

$\mathbf{H}_{GN}$ is symmetric positive definite, and therefore the quadratic approximation of $\mathcal{C}$ is convex. However, $\Delta \mathbf{w} = -\mathbf{H}_{GN}^{-1}(\mathbf{w}_k) \nabla \mathcal{C}(\mathbf{w}_k)$ can still increase rather than decrease $\mathcal{C}(\mathbf{w})$ if the step overshoots the local minimum, which tends to be more likely further away from the optimum. This final shortcoming is solved by the following method.

1.5.3.2 Levenberg-Marquardt

The Levenberg-Marquardt (Nocedal and Wright 2006, Ch.10, pp258) extension of Gauss-Newton replaces the Hessian with:

$$\mathbf{H}_{LM} = \mathbf{H}_{GN} + \lambda \mathbf{I}. \quad (1.45)$$

For small values of λ , the update $\Delta \mathbf{w}$ behaves more like a Gauss-Newton update, and for large values of λ more like steepest descent. After choosing a sensible starting value for λ_0 (e.g., 0.001), the typical Levenberg-Marquardt iteration then proceeds as follows:

- Calculate a candidate parameter update $\Delta \mathbf{w}$ using Equations 1.45 and 1.40.
- If $\mathcal{C}(\mathbf{w} + \Delta \mathbf{w}) < \mathcal{C}(\mathbf{w})$ accept the update and set $\lambda = \lambda \div 10$.

- If $\mathcal{C}(\mathbf{w} + \Delta\mathbf{w}) > \mathcal{C}(\mathbf{w})$ set $\lambda = \lambda \times 10$ and recompute $\Delta\mathbf{w}$ and $\mathcal{C}(\mathbf{w} + \Delta\mathbf{w})$, repeating until $\mathcal{C}(\mathbf{w} + \Delta\mathbf{w}) < \mathcal{C}(\mathbf{w})$ and then accept the update.

This is the method I prefer when storing of $\mathbf{H}$ is feasible. When it is not, I prefer the following method.

1.5.3.3 MM Algorithms

The Minorise-Maximisation and Majorise-Minimisation (MM) algorithms are two methods for objective function maximisation and minimisation respectively (Hunter and K. Lange 2004). I will focus here on the Majorise Minimisation version as it applies to cost functions. In essence, what this algorithm states is that if $z(\mathbf{w}) \geq y(\mathbf{w}), \forall \mathbf{w}$, with both y and z convex and touching, then $\mathbf{w}_{k+1} = \arg \min_{\mathbf{w}} z(\mathbf{w})$ will either reduce y or leave it unchanged.

The Taylor approximation of $\mathcal{C}(\mathbf{w})$ using $\mathbf{H}_{GN}$ is convex, and serves as y . Chun and Fessler (2018) showed that because the diagonal elements of $\mathbf{H}_{GN}$ are all positive:

$$\mathbf{H}_{MM} = \text{diag}(|\mathbf{H}_{GN}|\mathbf{1}) \succeq \mathbf{H}_{GN} \quad (1.46)$$

Where $|\mathbf{H}_{GN}|$ is the matrix of the absolute values of $\mathbf{H}_{GN}$, $\mathbf{1}$ is column vector of ones and $\succeq$ means that $\mathbf{H}_{MM}$ is at least as positive definite as $\mathbf{H}_{GN}$, and therefore majorises y when used in the Taylor expansion (i.e., the Taylor expansion with $\mathbf{H}_{GN}$ replaced by $\mathbf{H}_{MM}$ can be used as z).

The major advantage of using $\mathbf{H}_{MM}$ is that it is only non-zero along its main diagonal, and therefore requires far less memory to store than $\mathbf{H}_{GN}$. This is particularly important when the number of parameters in $\mathbf{w}$ is large, as it may not be possible to store $\mathbf{H}_{GN}$.

1.6 Image (Dis)Similarity

A method of determining how similar (or dissimilar) two images are is core to the process of image registration. Such measures are often called image (dis)similarity measures, matching criteria or distance measures - but these all refer to the same concept.

Distance measures are generally scalar valued functions of the reference image f and transformed moving image g' , i.e.:

$$M : f, g' \rightarrow \mathbb{R} \quad (1.47)$$

There are generally three groups of distance measures in common use (Sotiras et al. 2013). The first are based on the distance between a discrete set of landmarks in each image. These methods generally require a known correspondence between landmarks in order to estimate the transformation. Identifying such landmarks reliably is not a solved problem however, and even expert annotators may disagree on their placement within an image. Additionally, landmark location tends to be sparse - meaning that the deformation at non-landmark locations are driven entirely by regularisation (see Section 1.7) and interpolation. The second are based directly on the intensity values of each image. These measures tend to be dense - and a meaningful similarity measure can be made at almost all points within the image. However, this increased density brings with it an increase in computational complexity. Some of these measures also tend to be highly non-convex, causing sensitivity to both intensity inhomogeneities and initial alignment of the images. The third are hybrid methods, which combine intensity and landmark based methods - either through joint optimisation, or by treating the landmarks as a constraint.

I will focus exclusively on intensity based measures. They have emerged as the dominant methods applied in neuroimaging, which is the target of my work. Furthermore, the potential issues with intensity based methods can be addressed by the method developer, rather than relying on users as in landmark based methods.

1.6.1 Global Intensity Measures

A global intensity similarity measure is one for which the value at a particular voxel is independent of its neighbours. It assumes that the mapping of intensities in the same tissue type between two images is globally defined (Cachier et al. 2003).

1.6.1.1 Mean Squared Error

The error $\mathbf{e}$ of two vectors $\mathbf{a}$ and $\mathbf{b}$ of length N is defined as:

$$\mathbf{e} = \mathbf{a} - \mathbf{b} \quad (1.48)$$

The mean squared error (MSQ) is then defined as:

$$MSQ(\mathbf{a}, \mathbf{b}) = \frac{1}{N} \sum_{n=1}^N \mathbf{e}_n^2 \quad (1.49)$$

$$= \frac{1}{N} \|\mathbf{e}\|_2^2 \quad (1.50)$$

$$= \frac{1}{N} \mathbf{e}^\top \mathbf{e} \quad (1.51)$$

These vectors are typically made up of samples of images f and g . This makes certain assumptions about the intensities in the images, namely:

- The images are of the same modality, and therefore exhibit (nominally) the same contrast between tissue types
- There is no spatial intensity bias in the image. That is, an object's intensity in an image is independent of its position within that image.

Violating these assumptions can lead to suboptimal alignment. However, when they are satisfied the MSQ has a number of desirable properties - not least of which is its relative simplicity of calculation. Another benefit is the fact that it is a natural candidate for Gauss-Newton optimisation strategies as it is already in the expected sum-of-squares form. Although not unique to MSQ, the fact that each element of the error is independent means it is also a natural candidate for parallelisation.

1.6.1.2 Normalised Mutual Information

Normalised mutual information (NMI) overcomes the shortcoming of MSQ which requires images to have the same nominal contrast. It is an information theoretic approach based on Shannon entropy (Shannon 1948). The Shannon entropy H of a discrete random variable X is defined as:

$$H(X) = - \sum_{m=1}^M p_m \log p_m \quad (1.52)$$

where:

$M =$ the number of discrete values X can take on

$p_m =$ the probability that X takes on value m

Thus, $H(X)$ gives the average “information” or “uncertainty” in a random variable.

This concept can be extended to a *pair* of random variables X and Y , where the *joint* entropy is defined as:

$$H(X, Y) = - \sum_{m=1}^M \sum_{n=1}^N p_{mn} \log p_{mn} \quad (1.53)$$

where:

$N =$ the number of discrete values Y can take on

$p_{mn} =$ the probability that X takes on value m AND Y takes on value n

The voxels in images f and g can be thought of as the random variables X and Y respectively. If the intensities in the images are grouped into m and n histogram bins respectively, then $H(X)$ can be calculated from the normalised histogram of f , $H(Y)$ can be calculated from the normalised histogram of g and $H(X, Y)$ can be calculated from the normalised 2D joint histogram of f and g . The NMI is then defined as (Studholme et al. 1999):

$$NMI(X, Y) = \frac{H(X) + H(Y)}{H(X, Y)} \quad (1.54)$$

Note that the “normalised” in NMI refers to the fact that this metric is normalised for how much of the image is overlapping, and not normalised in the sense that values range from 0 to 1. Although other normalised forms of mutual information exist, this particular definition (which is normalised for the total overlap between images) is very specific to the field of image analysis.

Intuitively, this measure represents the ratio of the total information if the image intensities were independent of each other, over the actual total information. If the images truly are independent (often, but not always, visible as a smooth, non-sparse joint histogram) then the joint entropy will be high, and the NMI low. If the images

are not independent (again often, but not always, visible as a sharp, sparse joint histogram) then the joint entropy will be low, and the NMI high.

Image registration affects NMI primarily through the joint entropy term. As images become better aligned, the intensity in one image better predicts the intensity in the other, and therefore the information in their joint histogram decreases. Importantly, NMI does not assume a linear (or even monotonic) relationship between intensities in the images, and can therefore be used for inter-modal registration. However, it is still sensitive to intensity inhomogeneities - a common attribute of all global metrics.

1.6.2 Local Intensity Measures

In contrast to global intensity measures, a local intensity measure at a particular voxel will depend on some neighbourhood around that voxel. Consequently, these measures are less sensitive to intensity inhomogeneities - but this is generally at the expense of increased computational complexity (Cachier et al. 2003).

1.6.2.1 Local Normalised Cross-Correlation

One local measure which has proven to perform well is Local Normalised Cross-Correlation (LNCC) (Bajcsy and Kovačič 1989; Avants et al. 2008). First, a “locally normalised” version of each image is created, and for an image f is defined as:

$$\bar{f} = f - \mu_n \quad (1.55)$$

where μ_n is the mean intensity in an n^D region around each voxel, with D being the dimensionality of the image. This is entirely equivalent to smoothing f with an n^D box kernel, and then subtracting the smoothed image from f . In the Fourier domain the transformed box kernel is a sinc function, and therefore acts as a low pass filter. The dominant effect of subtracting the smoothed image is to remove the low frequency components of f , and thereby reduce low-frequency intensity inhomogeneities.

The LNCC is then calculated as:

$$LNCC(f, g) = \frac{\langle \bar{f}, \bar{g} \rangle^2}{\langle \bar{f} \rangle \langle \bar{g} \rangle} \quad (1.56)$$

Both the inner product in the numerator, and the means in the denominator are calculated over the same n^D neighbourhood for each voxel.

Some of the computational burden associated with having to work with neighbourhoods has been shown to be offsetable by pre-computing and reusing parts of the calculation which do not vary between iterations, meaning that LNCC can be made computationally tractable.

A potential drawback of LNCC though, is the somewhat “scale-free” nature of the metric. This is due to the product of the means in the denominator of Equation 1.56, which will inflate the penalty in areas with no real contrast (e.g., within the white matter of a T1-weighted image). Therefore these areas will tend to be deformed such that the *noise* correlates well - which is not a biologically plausible thing to do.

1.7 Regularisation

An image similarity measure alone is not necessarily sufficient to find a uniquely optimal transformation. That is, there may be many transformations, and thus sets of transformation parameters, which produce equally similar images. And even if a unique optimum *does* exist, there is no guarantee that the transformation required to reach that optimum is plausible. In other words, a transformation which requires the image to be greatly deformed but produces the best image similarity metric is *not* necessarily better than a less aggressive transformation which leads to a *slightly* worse measure of similarity.

Both of these issues are addressed by regularisation, which assigns a cost to the transformation ϕ itself, rather than to effect which ϕ has on the image similarity. This offers a way to offset a poorer similarity measure by a more plausible transformation.

The total cost function being minimised ($\mathcal{C}_{tot}$) is then the sum of the image similarity cost ($\mathcal{C}_{sim}$) and the regularisation cost ($\mathcal{C}_{reg}$):

$$\mathcal{C}_{tot}(f, g, \phi) = \mathcal{C}_{sim}(f, g, \phi) + \mathcal{C}_{reg}(\phi) \quad (1.57)$$

The cost-function landscape given by $\mathcal{C}_{tot}$ is considerably smoother and has a more central tendency with low values in a valley of plausible transformations.

There are many ways of choosing $\mathcal{C}_{reg}$, and I will discuss some of the most common choices below.

1.7.1 Mechanical

One of the most common families of regularisation methods are those based on modelling the transformation as if it is mechanically deforming an object. These are most easily interpreted in terms of mechanical stress and strain when applied in the small deformation framework, however they may also be applied to velocity fields in the large deformation framework, where their interpretation becomes more akin to viscous fluid resistance. Note that while all of these methods encourage smoothness, they do not ensure diffeomorphism.

1.7.1.1 Membrane Energy

The membrane energy (ME) (Ashburner and Friston 1999), of the transformation is defined as:

$$\sum_{\Omega} \sum_i^D \sum_j^D \left(\frac{\partial \phi_i}{\partial \mathbf{x}_j} \right)^2 \quad (1.58)$$

where:

$i, j = \text{direction}$

$D = \text{transformation/image dimensionality}$

Intuitively, it can be seen that this penalises how quickly the transformation changes across the image

1.7.1.2 Bending Energy

The bending energy (BE) (Bookstein 1997a; Bookstein 1997b) of the transformation is defined as:

$$\sum_{\Omega} \sum_i^D \sum_j^D \sum_k^D \left(\frac{\partial^2 \phi_i}{\partial \mathbf{x}_j \partial \mathbf{x}_k} \right)^2 \quad (1.59)$$

where:

$$i, j, k = \text{direction}$$

$$D = \text{transformation/image dimensionality}$$

BE penalises the second mixed partial derivative of the transformation, leading to smoother deformations than ME.

1.7.1.3 Linear Elasticity

The linear elastic energy (LE) (M. Miller et al. 1993) of the transformation is defined as:

$$\sum_{\Omega} \sum_i^D \sum_j^D \left(\lambda \left(\frac{\partial \phi_i}{\partial \mathbf{x}_i} \frac{\partial \phi_j}{\partial \mathbf{x}_j} \right) + \mu \left(\frac{\partial \phi_i}{\partial \mathbf{x}_j} + \frac{\partial \phi_j}{\partial \mathbf{x}_i} \right)^2 \right) \quad (1.60)$$

where:

$$i, j = \text{direction of transformation}$$

$$D = \text{transformation/image dimensionality}$$

This is again a commonly used form of regularisation, but the analogy to an elastic deformation holds only for small deformations where the stress and strain relationships within an object can be considered linear.

1.7.2 Smoothing

While including regularisation as a cost function during optimisation is an easily understood, principled approach, it does add computational complexity to the optimisation. In an iterative optimisation framework, an alternative approach would be to calculate an update to the transformation based only on the image similarity, followed by an update based only on the regularisation. A very common way of approximating a regularisation update is by smoothing either the update step, the total transformation, or both. Smoothing is usually performed using a Gaussian kernel, although this is not a requirement and different kernels can be used to encourage different characteristics in the transformation similar to the mechanics-derived registration above.

1.7.2.1 Diffusion-like

When the total transformation is smoothed after each update, this is known as diffusion-like regularisation (Thirion 1998; Vercauteren et al. 2009). This has been shown to approximate some mechanics-derived regularisation schemes. As such, it is similarly more suited to smaller deformations, and leads to overall less aggressive deformations.

1.7.2.2 Fluid-like

When the update to the transformation is smoothed before updating the total transformation, this is known as fluid-like regularisation (Avants et al. 2008; Vercauteren et al. 2009). Each update is seen as a small time-step in a velocity-based transformation which is why regularisation of this sort can be interpreted as constraining the flow of a fluid. This is a common choice when large deformations are desired, as updates are only locally constrained, allowing the final deformation to remain quite aggressive.

1.7.3 Hyperelastic

One drawback of the mechanical regularisation models is that they assume that transformations are comparatively small, and therefore the energy required to deform the images scales linearly with the amount of deformation. They are therefore unable to enforce diffeomorphism for larger deformations. A few hyperelastic methods have been proposed which address this to a certain extent, and I will discuss two of the ones most relevant to my work here. These methods have their own drawbacks though as they tend to be far more computationally complex to implement.

1.7.3.1 Log-Jacobian Determinant

A logical method of enforcing diffeomorphism is to ensure that the Jacobian determinant of the transformation is always positive. One of the earliest proposed methods of enforcing this is to penalise a function of the logarithm of the Jacobian determinant

(Rohlfing et al. 2003). Most commonly, this involves cost functions of the form:

$$\mathcal{C}(\phi) = \sum_{\Omega} \log^2 |\mathbf{J}(\phi)| \quad (1.61)$$

This has the desirable quality that the penalty approaches ∞ as $|\mathbf{J}|$ approaches zero, however it does little to encourage smoothness in the transformation. As such it must often be combined with linear mechanical penalties such as bending energy.

1.7.3.2 Riemannian Elasticity

Pennec et al. (2005) proposed an elegant penalty based on their log-Euclidean tensor framework. This is essentially a nonlinear extension of the linear elasticity mechanical regularisation, and is of the form:

$$\mathcal{C}(\phi) = \sum_{\Omega} \text{tr} \left(\log^2 (\mathbf{J}^T(\phi) \mathbf{J}(\phi)) \right) \quad (1.62)$$

Conceptually, this penalises transformations which deviate from rigidity as it is minimised only when J is orthogonal. As a consequence, this is the most similar form of regularisation to the one I present in Chapter 3 of which I am aware.

1.8 Exemplar Methods

In this section I briefly describe some existing registration methods to help contextualise where MMORF fits in.

1.8.1 FNIRT

FMRIB's Nonlinear Image Registration Tool (FNIRT) (Andersson et al. 2007; Andersson et al. 2019) is the immediate predecessor to MMORF, and they therefore share a number of similar characteristics.

FNIRT uses a small deformation, cubic B-spline parametrised free form deformation model. The use of a sum of squared differences cost function means that it is designed for within-modality registration only. FNIRT employs bending energy regularisation by default, but membrane energy can also be used. As FNIRT's transformation and regularisation models do not ensure diffeomorphism by default,

it uses the method proposed by Karacali and Davatzikos (2004) to project any non-diffeomorphic warps onto the closest diffeomorphic warp. FNIRT uses the Levenberg-Marquardt variant of Gauss-Newton to optimise the registration at lower warp resolutions, and the Scaled Conjugate Gradient (Møller 1993) algorithm at higher resolutions. One of the most powerful aspects of FNIRT is the extent to which intensity non-linearities are modelled as part of the registration process. FNIRT allows modelling of within subject intensity bias fields using B-splines, between-subject non-linear intensity mismatches, as well as within-subject nonlinear changes in tissue contrast. In this way, intensity and shape differences between images are simultaneously estimated and optimised.

1.8.2 Demons

Thirion’s Demons (Thirion 1998) treats nonlinear registration as a diffusion process, where the “demons” push on an image depending on local dissimilarity. In essence it is a non-parametric method which produces a dense displacement field which is regularised through Gaussian smoothing. A number of variations of the original algorithm have been proposed over the years, mostly focussed on how the displacement field is updated at each iteration. The original method suggested composing the result from the previous iteration ϕ^k with the update u^k such that $\phi^{k+1} = \phi^k \circ (\mathbf{I} + u^k)$. Although this does not necessarily lead to a diffeomorphic transformation, if u is sufficiently small and sufficiently smooth it can. In the interest of computational efficiency, an update of $\phi^{k+1} = \phi^k + u^k$ has been used, although registration accuracy and convergence tends to suffer as a result (Vercauteren et al. 2009). A diffeomorphic version of the Demons algorithm with symmetric demons forces (i.e., they are independent of the choice of moving and reference image) was proposed by Vercauteren et al. (2008) and Vercauteren et al. (2009). This method uses the update $\phi^{k+1} = \phi^k \circ \exp(u^k)$ where $\exp(u^k)$ is the exponentiation of a velocity field as described by Arsigny et al. (2006a), and can probably be considered the most complete version of the Demons algorithm.

1.8.3 ANTs

The Advanced Normalisation Tools (ANTs) (Avants et al. 2008) has received a lot of attention since being identified as the best performing nonlinear alignment tool in a comparison of 14 methods by A. Klein et al. (2009). Its wide adoption, proven performance and mature implementation mean that it can generally be considered as the current state-of-the-art method. Although ants was originally presented as an LDDMM-based (Beg et al. 2005) diffeomorphic method, the most widely used (and recommended by the tool’s authors) version uses the compositional variation of the Demons algorithm (Joshi et al. 2004), with a few notable differences, and is distributed as part of the Insight ToolKit (Avants et al. 2014). ANTs two main selling points are that the warps it produces are symmetric (or inverse consistent in that $\phi_{f \rightarrow g} \circ \phi_{g \rightarrow f} = \mathbf{I}$), and that the LNCC metric it uses has been shown to perform well even when aligning images with challenging intensity inhomogeneities.

1.8.4 DARTEL

Diffeomorphic Anatomical Registration using Exponentiated Lie algebra (DARTEL) (Ashburner 2007) employs exponentiation of a stationary velocity field using the method of Arsigny et al. (2006a). This method is very elegant in that a diffeomorphic transformation can be calculated without the need for the velocity field to be time varying. This does reduce the space of possible deformations compared to the full LDDMM method, but it does not appear to affect the quality of registration which can be achieved. Additionally, Ashburner uses a fast “scaling and squaring” technique to evaluate the exponential making this method very computationally efficient when combined with a Gauss-Newton optimisation strategy. The default image similarity metric used is the sum of squared differences, and a range of mechanical regularisation terms are available.

1.8.5 Geodesic Shooting

Geodesic Shooting (Ashburner and Friston 2011) is the follow-up tool to DARTEL and provides an efficient method of extending that tool’s methods into a full

LDDMM method. But, instead of trying to optimise the very computationally intensive time-varying warp field, Ashburner shows that any LDDMM registration can be completely specified by an initial momentum field, which is related to the initial velocity field. This tool represents the current best effort to make the elegant and mathematically rigorous LDDMM framework practically applicable to real-world neuroimaging problems.

1.8.6 NiftyReg

NiftyReg (Modat et al. 2010) is similar to FNIRT in that it uses a B-spline parametrised deformation and bending energy regularisation, however it differs in that it uses a normalised mutual information cost function and gradient descent optimisation. The reason for including a description of NiftyReg is that it is one of the only tools which uses GPU parallelisation to accelerate the registration optimisation. They show an order of magnitude speedup of execution times which sets a good benchmark against which to compare this work

1.8.7 Diffusion Methods

Diffusion MRI registration is has its own particular set of considerations which set it apart slightly from the methods presented here. As such, I discuss the most relevant methods by H. Zhang et al. (2006), Yeo et al. (2009) and Irfanoglu et al. (2016) in the context of diffusion registration as a whole in Chapter 4.

1.9 General Purpose Graphics Processing Units

Much of the work in this thesis is only made possible through the use of Graphics Processing Units (GPUs). GPUs provide a relatively affordable method of parallelising computations, with thousands of calculations able to be performed simultaneously on hardware that can fit into a standard laptop or desktop computer.

In order to understand some of the design decisions presented in Chapter 2, I will now introduce a number of GPU-specific concepts. The terminology used is

specific to Nvidia’s CUDA framework (NVIDIA 2019), but the concepts apply more generally.

1.9.1 Hardware Terms

1.9.1.1 Host and Device

Code written with CUDA is designed for heterogeneous computing. In other words, the memory and arithmetic calculations do not all occur on the same device, or even the same type of device. In CUDA terms, the *Host* is the main CPU powered machine. The Host is responsible for coordinating tasks and transferring data. The GPU on which parallel computing occurs is termed the *Device*. Both the Host and the Device have their own memory areas, which are not visible to the other. As such, the Host and Device can only operate on data in their own memory, requiring data be transferred between them across a data bus. Additionally, all calculations on the Device must be initiated from the Host. Execution is, however, asynchronous between Host and Device, and explicit synchronisation is necessary to ensure code executes correctly.

1.9.1.2 Compute Capability

CUDA enabled devices are categorised into various *Compute Capabilities*. The compute capability of a Device is linked to the hardware architecture used in that device, and determines many of parameters which may be used when invoking code on the Device. As such, not all Devices support the same subset of the CUDA programming interface, or may require different parametrisation to execute code with maximum efficiency.

1.9.1.3 Streaming Multiprocessor

A *Streaming Multiprocessor* (SM) is the main on-chip architectural unit within which CUDA code runs. It encompasses processing cores, code schedulers and on-chip memory areas. SM functionality is Compute Capability and Device dependent.

1.9.1.4 Global Memory

Global Memory refers to the Random Access Memory (RAM) on the Device. This is both the largest (typically between 2GB and 32GB) and the slowest memory store on the Device. Typically this is used to store data which the Device will use when executing Kernels (Section 1.9.2.2). This is also the memory area which the Host can copy data from/to.

1.9.1.5 Shared Memory

Shared Memory is a user managed cache located on-chip, and is therefore much faster to access than Global Memory. Shared Memory is, however, limited in size and shared with the Device's L1 cache. Shared Memory is common to a Block (Section 1.9.2.4), and visible for both reading and writing to by all Threads (Section 1.9.2.3) within a Block.

1.9.1.6 Register Memory

Register Memory is the fastest on-chip memory store, and is only visible to an individual Thread. The total number of Registers available to each SM is Device dependent, and may be the limiting factor as to how many Threads can be simultaneously launched by any SM.

1.9.1.7 Texture Memory

Texture Memory is a fixed latency memory cache (i.e., memory for which each read/write access from a Thread takes a fixed time regardless of if other Threads in the same Warp accesses adjacent memory or not) that leverages information about the 1D, 2D or 3D spatial locality of data to reduce slow global memory reads.

1.9.2 Execution Coordination Terms

1.9.2.1 SIMD

SIMD stands for *Single Instruction Multiple Data*. This refers to multiple Threads all executing the exact same logical instruction at the same time, but acting on

different input and output data. This is an example of data-level parallelism and is critical to efficient CUDA code performance.

1.9.2.2 Kernel

Possibly the most fundamental building block when writing CUDA code is the Kernel. A Kernel is a block of code designed to run on the GPU. It is called exclusively as a function from the Host, but executes exclusively on the Device. When used correctly, many instances of the Kernel code should execute simultaneously on the Device, performing the same operations on different data.

1.9.2.3 Thread

Threads form the first of two parallelisation mechanisms available within CUDA. When a Kernel is invoked, it will execute once for each Thread specified in the Kernel launch parameters. Threads are arranged in Blocks, with the maximum number of Threads which can be assigned to a Block depending on a Device's particular Compute Capability. The number of Threads can be declared in 1, 2 or 3 dimensions, with each Thread uniquely identifiable by a combination of its `threadIdx.x`, `threadIdx.y` and `threadIdx.z` values. The maximum number of Threads is dependent on the Compute Capability of the Device. The Threads within a Block have mechanisms to efficiently communicate and synchronise.

1.9.2.4 Blocks

Blocks form the second parallelisation mechanism available with CUDA. When a Kernel is invoked, it will launch the number of Threads requested once for every Block specified. Similarly to Threads, Blocks are arranged in Grids, which may be 1, 2 or 3 dimensional. The Block to which any particular Thread belongs can be uniquely identified by a combination of its `blockIdx.x`, `blockIdx.y` and `blockIdx.z` values.

1.9.2.5 Warp

An often confusing aspect of CUDA is that of *Warps*. A Warp is a logical grouping of Threads which are loaded for execution simultaneously on a SM. In all Compute Capabilities to date, a Warp is 32 Threads in size. Which data a Warp is accessing, and the divergence of any branching code within the Threads of a Warp have major implications for code execution efficiency.

1.9.3 Performance Terms

1.9.3.1 Occupancy

Occupancy is defined as the number of Warps executing concurrently on the GPU, divided by the maximum number of Warps which the GPU is capable of executing. This can be thought of as a measure of how busy the GPU is being kept. Occupancy is one of the first measures which should be optimised when profiling CUDA code, as it has an immediate and dramatic effect on code execution speed, and sets an upper bound on the achievable utilisation of the device.

1.9.3.2 Memory Coalescing

Coalescence is a method of accessing Global Memory, whereby all Threads within a Warp access contiguous memory addresses at any particular point in time (i.e. SIMD on contiguous data). The concept of coalesced memory access is crucial in optimising CUDA code, and it is therefore described in more detail in Figure 1.4

1.9.3.3 Warp Divergence

Warp Divergence occurs when Threads within a Warp do not follow the same branching pattern during execution. As only one instruction can be given to a Warp at a time, each divergent path must be executed serially rather than in parallel.

1.9.3.4 Warp Efficiency

Warp Efficiency is the average percentage of Threads within a Warp that are not serialised, and are therefore able to execute simultaneously. This is equivalent to the

minimum number of instructions performed by any Thread in the Warp compared to the total number of instructions issued to the Warp.

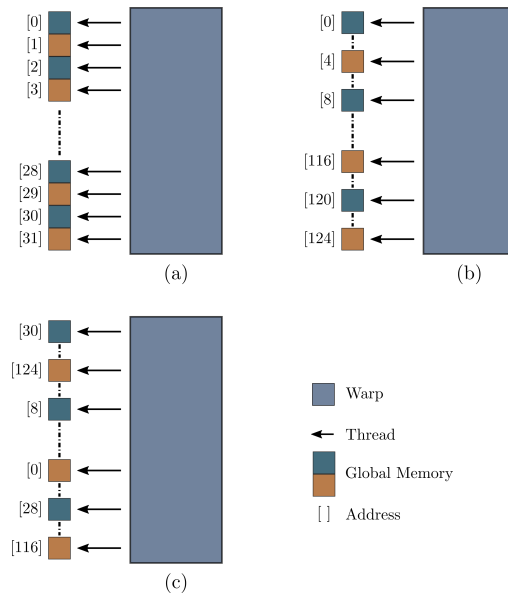


Figure 1.4: Global Memory access patterns. The pattern with which Global Memory is accessed is critical to Kernel performance. Note that Global Memory access patterns are only relevant within a given Warp (group of 32 Threads). In other words, one Warp’s access pattern does not impact on the execution efficiency of any other Warps. Consider a scenario where all Threads access a single byte of data during an instruction cycle (assuming SIMD parallelism). (a) shows the ideal memory access pattern, wherein all Threads within a Warp access a contiguous region of Global Memory. Every time Global Memory is read by a Warp, 32 Bytes of data are returned. If the returned data can be used by multiple Threads within a Warp, then that data does not need to be read again. This is termed *coalesced memory access*. If, however, two Threads attempts to access memory which does not fall within the same 32 Byte block, then the memory access attempts will be serialised. This negatively impacts both the bandwidth and the latency of data transactions. There are two main situations in which a non-coalesced memory read arises. The first is due to strided memory access, and is illustrated in (b). In this case, Threads access memory in an ordered manner, with some constant offset between memory addresses. This is often the case when each Thread performs an operation on a separate row of a matrix which has been stored in row-major order. If the stride is less than 32 Bytes, then a partially coalesced memory access may occur, but beyond this all memory accesses will be forced to execute in a serial manner. Due to the well defined nature of strided memory access, it may be possible to remove the stride and restore coalescing by rearranging how the data is stored in Global Memory (by storing a matrix in column-major order for example). Secondly, uncoalesced memory access can occur due to Threads reading data in a random order, as shown in (c). In this case, changing the storage pattern in Global Memory is unlikely to increase coalescence. If the overall size of the data permits, then storing data in shared memory may be the best alternative. This allows for the possibility of transferring the shared data in a coalesced manner, even if the Threads will not be using it in the same order. Note that these considerations apply to both reading and writing of Global Memory. This may also be extended to the case where a Thread reads more than a single byte of data at a time (for example when accessing a 32 bit float). In this case, all the previous considerations remain unchanged, with the only difference being that a single Warp will perform multiple coalesced memory reads during each instruction cycle.

2

The Multimodal Registration Framework

Contents

2.1	Introduction	45
2.2	Aims of MMORF	45
2.2.1	Assumptions	46
2.2.2	Constraints	46
2.3	Transformation Model	46
2.3.1	Small Deformation Framework	46
2.3.2	Parametrisation	48
2.4	Optimisation Strategy	52
2.5	Image Similarity	54
2.6	Multiresolution Pyramid and Smoothing	55
2.7	Modality Weighting	57
2.8	Software Design	62
2.8.1	UML Notation	62
2.8.2	Object Functionality	63
2.8.3	Object Relationships	67
2.8.4	Object Interactions	68
2.9	Code Optimisation	70
2.9.1	GPU Specific Considerations	70
2.9.2	Profiling Results	76
2.10	Conclusion	82

2.1 Introduction

In Chapter 1 I presented the basic distinguishing features of general registration frameworks, and described some currently available registration tools. In this chapter I will present the registration tool I have developed - discussing the aims of the software, and the choices I have made regarding transformation model and optimisation strategies given the context of Chapter 1. Additionally I will address some of choices common to all registration framework, such as how much smoothing to apply to an image and the weighting of regularisation.

My method is called MMORF, which stands for MultiMOdal Registration Framework. This reflects the *core* aim of this tool - the ability to simultaneously register multimodal datasets. I stress here that I am not referring to *within* subject coregistration of multiple modalities, but rather *between* subject (or subject to template) registration which simultaneously considers the similarity of all available modalities.

2.2 Aims of MMORF

MMORF was developed with the following key aims in mind:

1. MMORF should be capable of handling scalar and tensor image alignment either individually or simultaneously
2. MMORF should produce the most *anatomically plausible* deformation given the available data
3. MMORF should deliver registration accuracy as good as or better than the current state-of-the-art
4. MMORF should be sufficiently computationally efficient to be used for population imaging studies with tens of thousands of subjects
5. The code underpinning MMORF should be designed and written according to good software engineering practices

2.2.1 Assumptions

In order to set boundary conditions on what MMORF should be capable of, I made the following assumptions:

1. Either all intra-subject modalities are already in the same space ...
2. ... or affine transformations between all modalities are available
3. All images are in NIfTI format
4. Vectors and, by extension, tensors are defined in *radiological* orientation
5. DTI volumes conform to the FSL `dtifit` standard¹

2.2.2 Constraints

There is a single major constraint applicable to MMORF, namely that:

1. A CUDA-capable Nvidia GPU must be installed on the device running MMORF

2.3 Transformation Model

MMORF is a volumetric registration algorithm, which produces a dense displacement map as its output. A small-deformation, or elastic framework, was chosen for the deformation model. MMORF therefore solves for a displacement field directly, rather than using an intermediate velocity field.

2.3.1 Small Deformation Framework

There is extensive literature expounding the benefits of large deformation, velocity-based, diffeomorphic approaches (Beg et al. 2005; Ashburner 2007; Avants et al. 2008; Ashburner and Friston 2011; Modat et al. 2011; Irfanoglu et al. 2016; Mang and Biros 2016), and it does often seem as though the field of image registration

¹<https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/FDT/UserGuide#DTIFIT>

has left the small-deformation framework behind. This is unfortunate, as it neglects unique benefits available when working directly with displacement fields.

Consider, for example, if you wish to penalise a measure of local deformation induced by the warp, such as the Jacobian determinant. As a minimum, you require the ability to compute this measure directly in order to assign a penalty for a given displacement field. This is not really a problem for any transformation model - as once you have the final dense deformation field such measures can be readily calculated using finite difference methods. Rather, an issue arises when trying to determine what effect a parameter change would have on the chosen penalty. That is, calculating the derivative of the penalty with respect to the transformation parameters.

In the small deformation framework, you are manipulating the final deformation field directly and therefore analytic derivatives of these measures can be calculated with comparative ease. In the large deformation framework, a parameter change must be propagated through an integral before the effect can be known. Propagating the derivative through this integral is even more complex, and therefore a direct penalty on a deformation measure of interest becomes computationally prohibitive.

That is not to say that it is impossible to do - as Mang and Biros (2016) and others have shown, in a stationary velocity field it is possible to penalise measures such as divergence and curl to achieve the desired form of regularisation. However there are several limitations to this approach - a key one being that the penalty is now coupled to a location in the velocity field, rather than a location in the object being deformed.

Consider the case where a spatially varying regularisation is desired (*e.g.*, because you know a priori that one particular region of the object is likely to be more strongly deformed than another). In the small deformation setting, one may assign a spatially varying weighting to each position in the image to achieve this. However, a spatially varying weighting in the large deformation framework would only change how that point in the *velocity field* deforms the object, rather than constraining any part of the object itself. In other words, you are penalising all parts of the image that move *through* that point during the registration, which cannot be known a priori.

Small deformation frameworks have their own set of limitations though - a major one being the fact that deformations are not guaranteed to be diffeomorphic by default. However, I overcome this through a regularisation term which, amongst other things, constrains the Jacobian determinant to be positive. An analysis of the performance of this method of regularisation compared to an inherently diffeomorphic transformation is presented in Chapter 3.

2.3.2 Parametrisation

Cubic B-splines were chosen to parametrise the deformation. They possess a number of desirable characteristics, such as:

C2 continuity *i.e.*, continuous, analytic derivatives up to second order

Compact support *i.e.*, each spline has only a finite non-zero extent

Spatial separability *i.e.*, an n -dimensional spline (and its spatial derivatives) can be decomposed into the tensor product of n 1-dimensional splines (or differentiated splines)

Although I defined image registration in Chapter 1 in terms of continuous images, in practice I only ever deal with discretely sampled images. One can therefore describe the process of using a set of B-spline basis functions to transform a set of sample coordinates in a reference image f to their corresponding coordinates in a moving image g as follows:

$$f = \text{Reference image} \quad (2.1)$$

$$g = \text{Moving image} \quad (2.2)$$

$$N = \text{Number of sampled voxels in } f \quad (2.3)$$

$$\mathbf{x}/\mathbf{y}/\mathbf{z} = x, y \text{ and } z \text{ coordinates of samples in } f \quad (2.4)$$

$$\mathbf{X} = \underbrace{\begin{bmatrix} \mathbf{x}^\top \\ \mathbf{y}^\top \\ \mathbf{z}^\top \end{bmatrix}}_{3 \times N} = \begin{bmatrix} x_0 & x_1 & \cdots & x_N \\ y_0 & y_1 & \cdots & y_N \\ z_0 & z_1 & \cdots & z_N \end{bmatrix} \quad (2.5)$$

$$M = \text{Number of splines per warp direction} \quad (2.6)$$

$$\mathbf{w}_x/\mathbf{w}_y/\mathbf{w}_z = x, y \text{ and } z \text{ direction warp parameters} \quad (2.7)$$

$$\mathbf{W} = \underbrace{\begin{bmatrix} \mathbf{w}_x^\top \\ \mathbf{w}_y^\top \\ \mathbf{w}_z^\top \end{bmatrix}}_{3 \times M} = \begin{bmatrix} w_{x0} & w_{x1} & \cdots & w_{xM} \\ w_{y0} & w_{y1} & \cdots & w_{yM} \\ w_{z0} & w_{z1} & \cdots & w_{zM} \end{bmatrix} \quad (2.8)$$

$$\mathbf{b}_m = \text{Vectorised } m^{\text{th}} \text{ B-spline at sample positions } \mathbf{X} \quad (2.9)$$

$$\mathbf{B} = \underbrace{\begin{bmatrix} \mathbf{b}_0^\top \\ \mathbf{b}_1^\top \\ \vdots \\ \mathbf{b}_M^\top \end{bmatrix}}_{M \times N} = \underbrace{\begin{bmatrix} b_{00} & b_{01} & \cdots & b_{0N} \\ b_{10} & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ b_{M0} & \cdots & \cdots & b_{MN} \end{bmatrix}}_{\substack{\text{1 voxel per column}}} \left. \vphantom{\begin{bmatrix} b_{00} \\ b_{10} \\ \vdots \\ b_{M0} \end{bmatrix}} \right\} \text{1 spline per row} \quad (2.10)$$

$$\phi(\mathbf{X}, \mathbf{W}) = \text{Transformed sample coordinates} \quad (2.11)$$

$$= \mathbf{X} + \mathbf{WB} \quad (2.12)$$

For simplicity, I use the same configuration of B-splines to define the warps in all 3 directions (*i.e.*, the number and spatial extent of splines defining the displacements in each direction is the same, and they are located identically in space). M is chosen to be the set of splines whose spatial support, wholly or partially, overlaps with the domain over which f is defined. The warp parameters $\mathbf{w}_{x/y/z}$ are then the coefficients of each B-spline, and each parameter only affects the displacement in a single direction.

The compact support of B-splines means that $\mathbf{B}$ is very sparse. Additionally, (disregarding edge cases) each row of $\mathbf{B}$ is simply a shifted version of any other row. I therefore never store $\mathbf{B}$ explicitly, and instead compute $\mathbf{WB}$ using convolution.

Another benefit of the sparsity in $\mathbf{B}$ is that it induces sparsity in the Hessian of cost functions based on this transformation. For an MSQ cost function, each element of the Hessian can be calculated as:

$$\mathbf{H}_{ij} = \sum_{\substack{\mathbf{x} \in \mathbf{b}_i \cap \mathbf{b}_j \\ \text{spline intersection}}} \mathbf{b}_i(\mathbf{x}) \mathbf{b}_j(\mathbf{x}) \underbrace{\frac{\partial g}{\partial xyz(i)}(\phi(\mathbf{x})) \frac{\partial g}{\partial xyz(j)}(\phi(\mathbf{x}))}_{\text{gradient image}}, \quad (2.13)$$

where $\frac{\partial g}{\partial xyz(i)}$ means differentiates with respect to the direction in which B-spline i causes displacement to occur. As each entry in the Hessian represents the interaction between two of the B-spline basis functions, only those combinations of splines which overlap in their spatial support will ever produce non-zero entries. Assuming that the image dimensionality D matches the transformation dimensionality (i.e., you warp a 3D volume in the x , y and z directions), then the number of non-zero entries per row of the Hessian is at most $D \times 7^D$ (equalling 1029 in 3D) for cubic B-splines. Considering that the number of parameters being optimised over can easily exceed 10^6 , this would lead to a matrix which is at least 99.9% sparse. The redundancy increases as the warp resolution increases (the knot-spacing is reduced), which means that the more parameters one attempts to estimate the sparser the Hessian. This means that the memory requirement for the Hessian scales much better with resolution than might initially be feared.

Furthermore, the radial symmetry of B-splines leads to redundancies in the Hessian. Figure 2.1 shows a 2D example of four B-spline pairs which produce the same value. The corresponding locations of the entries in the Hessian are shown in Figure 2.2. This implies that only one of these configurations needs to be calculated, and the value can then simply be stored for each of the other three entries. The redundancy for a value can be up to 2^D , implying that only approximately 1 in 8 entries of the Hessian actually needs to be calculated in the 3D case. Note that in cases where the cost function depends on the spatial derivatives of the B-splines (such as the SPRED regularisation penalty in Chapter 3, and the rotational component of the DTI penalty in Chapter 4), this symmetry (and the associated redundancy) is reduced as the differentiated B-splines are no longer radially symmetric. The degree of sparsity, however, still remains.

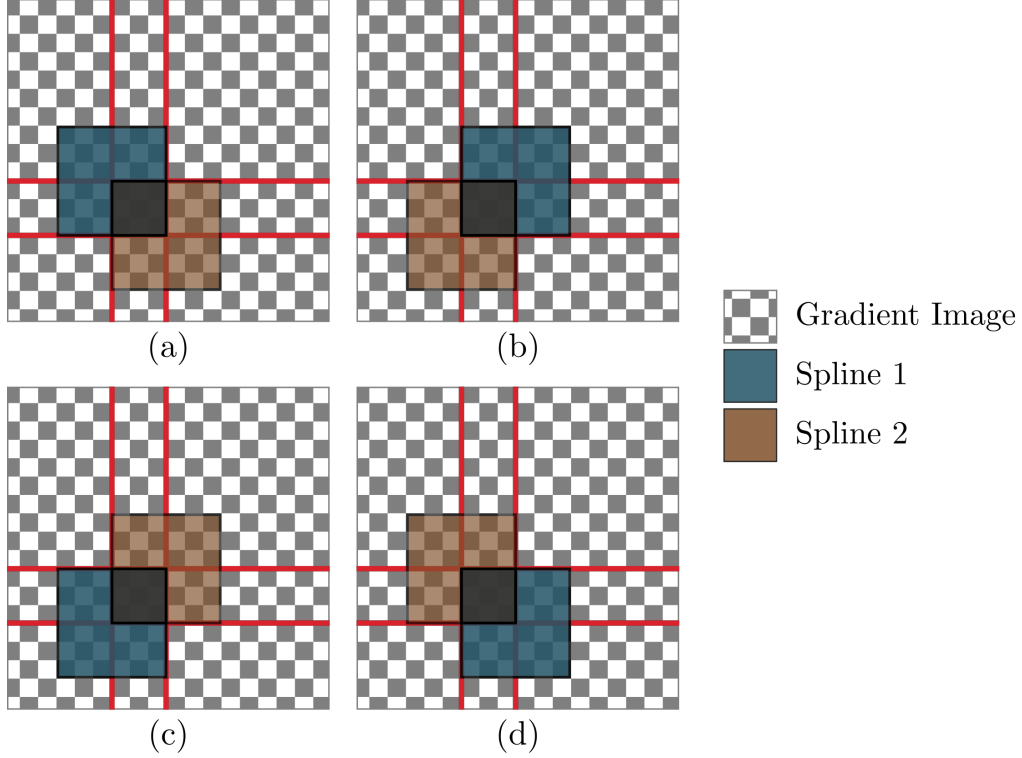


Figure 2.1: 2D example showing emergence of redundant values in the Hessian. Each term in the Hessian (Equation 2.13) corresponds to a configuration similar to that which is shown in (a) to (d) above. The Hessian value is calculated by multiplying the intensity of every point in two partial derivative images with each other, and then by each of two B-splines, and then summing across the entire image. The fact that B-splines have limited support results in zero magnitude over the majority of the image. Therefore, any point where either spline has no support (i.e. its magnitude is zero) will not contribute to the value of the Hessian. Another way of stating this is that only areas where the support of both splines overlap will contribute to the value of the Hessian for a particular configuration. There exist, therefore, multiple configurations of spline overlap for which the area of interest in the *Gradient Image* is the same. (a) and (b) represent the configuration where *Spline 1* and *Spline 2* have had their x -coordinate swapped such that the area of overlap remains the same. Similarly, (a) and (c) represent the configuration where *Spline 1* and *Spline 2* have had their y -coordinates swapped such that the area of overlap remains the same. (a) and (d), therefore, represent the configuration where *Spline 1* and *Spline 2* have had both their x and y -coordinates swapped to achieve the same overlap. Finally, if one recognises that B-splines are radially symmetric, it follows that the product of both splines at each point in the region of mutual support is identical for each configuration in (a) to (d), and therefore the value of the Hessian will be identical as well. This logic extends to the 3D case, except that the addition of a third dimension increases the the number of redundant configurations by a factor of two.

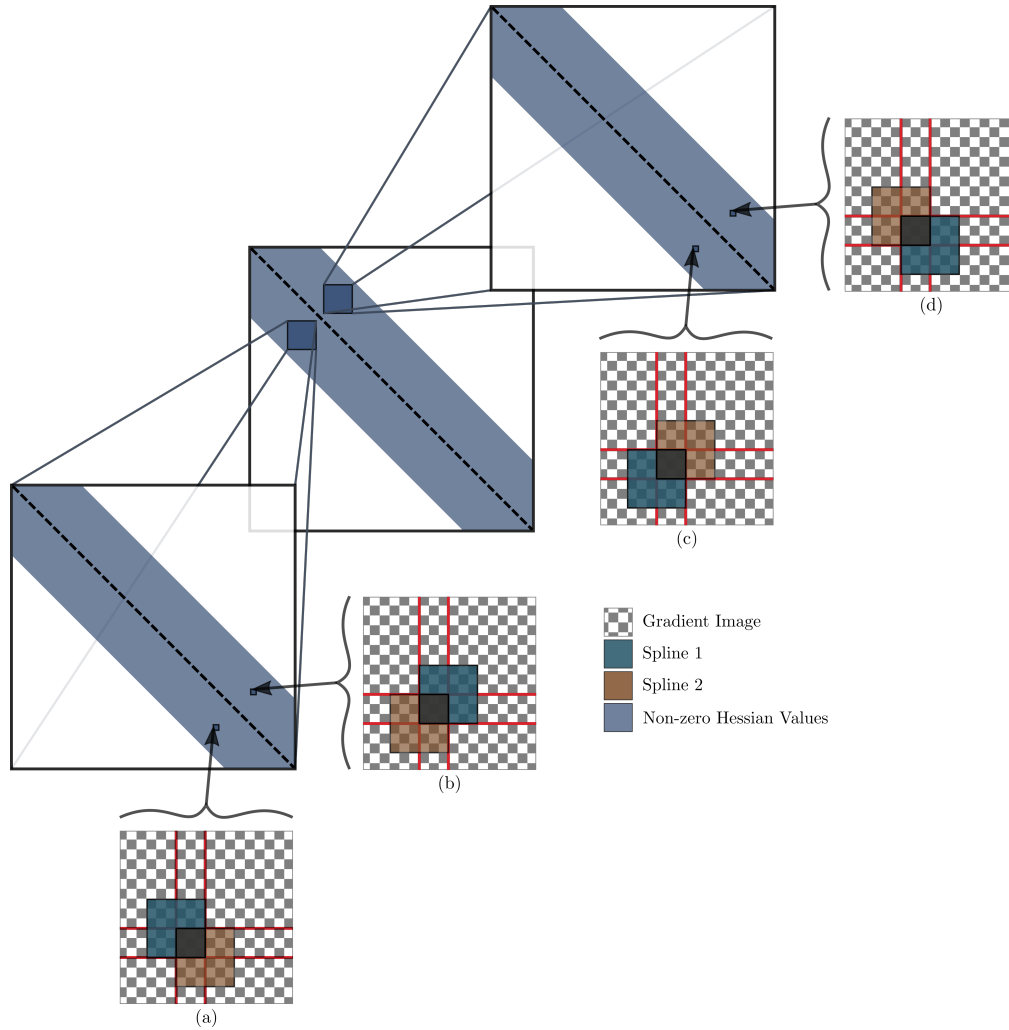


Figure 2.2: Nested levels of symmetry within the Hessian for a 2D image. This shows the practical implications of the spline overlaps of Figure 2.1. Once value (a) is calculated, values (b), (c) and (d) do not need to be calculated - only their indices in the Hessian relative to the position of (a) need to be determined. In 3D there is a third level of symmetry present.

2.4 Optimisation Strategy

I have chosen to use a combination of Levenberg-Marquardt (LM) (Levenberg 1944; Press et al. 2007) and Majorise Minimise (MM) (Hunter and K. Lange 2004) optimisation within MMORF.

When choosing an iterative optimisation strategy for convex functions, a trade-off is made between methods that take many “cheap” (computationally speaking) steps, and methods which take fewer “expensive” steps. Ideally, one wants to choose a method that, on average, minimises the product of the time taken to choose a

step, and the number of steps taken. Broadly speaking, this comes down to choosing between first and second order optimisation strategies.

First order strategies, which use gradient information only, tend to be simple to implement. They do, however, often require choosing some hyperparameters (such as how far to move along the gradient, or how much “momentum” to assign). Second order strategies are, by comparison, naturally more complex. As they use information about function curvature, interactions between parameters must be calculated, which often scales quadratically with parameter size. However, they do tend to have closed form update rules and can potentially converge in far fewer iterations than first order methods.

A general rule of thumb would be that unless a closed-form solution of the second order terms is available then first order optimisation should be preferred as finite difference approximations to second order methods are computationally taxing.

LM applied to B-spline parametrised optimisation has been shown to improve alignment, and require far fewer iterations to converge, when compared to other options such as gradient descent and conjugate gradient (Kybic and Unser 2003). B-splines lend themselves to LM optimisation as the terms of the Hessian matrix can be calculated analytically, eliminating the need for any finite difference approximations. Additionally, the finite support of B-splines leads to Hessians with a high level of sparsity (Andersson et al. 2019). These two factors mitigate, to an extent, the additional computational cost associated with utilising LM optimisation. The use of GPU parallelisation to speed up the Hessian calculation and the iterative solving of the resulting system of linear equations means that the benefits of LM optimisation can be had without paying the penalty of longer runtimes.

However, above a certain warp resolution, the Hessian becomes too large to store in computer memory despite its sparsity. In these instances, MMORF switches to an MM algorithm which replaces the LM Hessian with a diagonal approximation to it which is at least as positive definite, and requires on the order of 1000 times less memory to store. In practice, this method takes longer to converge than LM, but requires no other changes to the registration algorithm.

2.5 Image Similarity

In MMORF I have chosen to use the mean squared error (MSQ) cost function for scalar modalities, and the related squared Frobenius norm for tensor modalities. As I only consider intra-modal image similarity, the use of more complex cost functions capable of comparing inter-modal image similarity (such as normalised mutual information) is not a requirement. I perform a linear intensity mapping between images to account for differences in overall image intensity, but it may be that in future a global nonlinear mapping (such as that implemented by Andersson et al. (2019)) is added to improve registration when differences in acquisition parameters have caused substantial differences in the relative intensities across tissue types for scans that are nominally of the same contrast.

MSQ is also ideally suited to the Gauss-Newton family of optimisation strategies (including both LM and MM), which have an implicit assumption that the cost function being optimised is some form of squared difference. Provided that the function being optimised is guaranteed to be non-zero everywhere, other cost functions can still be used within this framework through a change of variables (as shown in Chapter 3, Section 3.2.1.3) - but it is convenient not to have to go down that path unless it is entirely necessary.

MSQ does, however, have a potential drawback in that it assumes the *intensity* of an object is independent of its *position* in the image. Unfortunately, spatial inhomogeneities in the transmission and reception of radio frequency pulses in MRI often manifests as spatially varying image intensities (Vovk et al. 2007). In other words, the intensity of an object in the image depends both on its “intrinsic” intensity, and a spatially varying “bias”. This bias tends to change smoothly, and can be estimated with relatively low degrees of freedom.

If the bias field is not explicitly accounted for, optimising an MSQ cost function can lead to differences in image *intensity* being incorrectly treated as incomplete image *alignment*. To make this concrete, consider a registration of a T1 image, g , to a T1-template, f , that is unaffected by bias field. Furthermore, imagine that g is affected by a bias field such that a region of the brain is brighter than it should

be (has higher signal). This will lead to an intensity mismatch between f and g , which would persist even if they were in perfect register. The registration will then try to “hide” the high signal by shrinking the tissue type with higher signal (white matter) and expanding the one with lower signal (grey matter).

Andersson et al. (2019) have demonstrated how a multiplicative bias field and a warp field can be estimated simultaneously in order to overcome this issue. A consequence of this approach is that it changes the diagonal structure of the Hessian, due to interactions between the bias field and warp parameters. This would have a highly detrimental effect on MMORF’s performance as efficient parallelisation of the Hessian calculation is reliant on exactly that diagonal structure (see Section 2.9.1.2 for details).

I have therefore chosen to also use a multiplicative bias field estimation, but carry out the estimation in an interleaved fashion with the warp field estimation. In this regime, the bias field is estimated, then the warp field is estimated, then the bias field estimation is updated, then the warp field estimation is updated, etc. This has proven to be an effective strategy as an accurate estimation of the bias field is reached before the warp is at a sufficiently high resolution to be effected by it.

I utilise the same representation and optimisation framework to calculate the bias field as I do the warp field. The primary difference is that the bias is represented by a single B-spline field as opposed to separate x , y and z B-spline fields for the warp. Furthermore, the field is kept at a single, coarser resolution with fairly high regularisation, encouraging smooth, slowly varying fields. This combination reduces the complexity of the optimisation to the level where it has a completely negligible effect on overall runtime, while greatly improving registration accuracy.

2.6 Multiresolution Pyramid and Smoothing

Coarse-to-fine multiresolution approaches to image registration (Bajcsy and Kovačič 1989; Szeliski and Coughlan 1997) have proven highly effective in practice, regardless of the transformation model used (Kybic and Unser 2003; H. Zhang et al. 2006; Andersson et al. 2007; Ashburner 2007; Avants et al. 2008; Modat et al. 2010;

Irfanoglu et al. 2016). Robustness is aided by varying the smoothing of the images - which allows low frequency content to be aligned first, followed by high frequency details. Computational efficiency is aided by subsampling, after smoothing, the original image.

The exact choice of resolutions for the pyramid is somewhat arbitrary. Many tools define the pyramid in terms of voxels, with the finest resolution set at the original voxel resolution. In MMORF however, different modalities may have different native resolutions, and therefore I made the decision to rather define the resolution pyramid in mm. As 1 mm isotropic is a very common resolution for structural MRI modalities, I tend to use this as the finest resolution. Each prior resolution of the pyramid is a factor of 2 coarser. In testing I have found that a coarsest resolution of 32 mm isotropic gives consistently good performance. This leads to a total of 6 levels in the pyramid.

It is common practice to smooth the images being aligned in step with the coarser levels of the pyramid. From a scale space theoretic perspective, smoothing with a Gaussian allows the optimisation to be sensitive to coarse details at first, and finer details as the registration converges (Witkin 1984; Babaud et al. 1986). In terms of signal processing, Gaussian smoothing also removes higher frequency components of the image, allowing the images to be subsampled without violating the Nyquist criterion.

One decision which is rarely discussed is how much to smooth an image for a given subsampling level. My approach to this is to have the user specify the amount of smoothing, and calculate the subsampling based on ensuring that the Nyquist criterion is met. Consider a Gaussian smoothing kernel with standard deviation σ . The Fourier transform of the Gaussian is then also a Gaussian, and

the following relationships hold:

$$\sigma = \frac{FWHM}{2\sqrt{2\ln 2}} \quad (2.14)$$

$$\begin{aligned} \sigma_f &= \frac{1}{2\pi\sigma} \\ &= \frac{\sqrt{2\ln 2}}{\pi FWHM} \end{aligned} \quad (2.15)$$

$$\begin{aligned} f_c &= \sqrt{2\ln 2}\sigma_f \\ &= \sqrt{2\ln 2} \frac{\sqrt{2\ln 2}}{\pi FWHM} \\ &= \frac{2\ln 2}{\pi FWHM} \end{aligned} \quad (2.16)$$

$$\begin{aligned} f_N &= 2f_c \\ &= \frac{0.8825}{FWHM} \end{aligned} \quad (2.17)$$

where FWHM is the Full Width at Half Maximum, σ_f is the standard deviation of the Fourier transformed Gaussian, f_c is the cutoff frequency (defined here as the Half Width Half Maximum the Fourier transformed Gaussian), and f_N is the Nyquist frequency associated with the filter.

From this, a simple choice of sampling frequency which satisfies Nyquist is $1/FWHM$. Therefore, in MMORF the sampling frequency is chosen to be $\max(1/FWHM, 1/\text{knot-spacing})$. Empirically, I found that a FWHM of $1/4$ the B-spline knot-spacing gave the best registration results, and I therefore used this as the default in MMORF.

2.7 Modality Weighting

Having a method which is capable of registering multiple modalities simultaneously begs the question of how best to combine the information across modalities. I have chosen to treat each modality as an additive term to the overall cost function. Each modality can then be individually weighted to balance the similarity and regularisation terms. This weighting can either be constant across the volume, or vary spatially if a particular modality should be up or down-weighted in certain

regions. The total cost for N modalities is then:

$$\mathcal{C}_{tot} = \sum_{\Omega} \left(\sum_{n=1}^N \lambda_n \odot \mathcal{C}_n + \lambda_{reg} \odot \mathcal{C}_{reg} \right), \quad (2.18)$$

where $\odot$ is the Hadamard, or element-wise, product which allows for both uniform and spatially varying λ s. This form is convenient as the gradient and Hessian are then similarly the weighted sums of the individual modalities' gradients and Hessians. I have implemented spatially varying weighting by treating λ as a soft mask defined in the reference image space.

Next arises the question of how to choose the weighting of each modality. For unimodal registration, the problem reduces to choosing the relative weighting between λ_1 and λ_{reg} . This problem is completely ubiquitous in image registration, and at the same time has no universally accepted solution - although some progress has been made towards inferring the required amount of regularisation using Bayesian inference (Simpson et al. 2012). A pragmatic approach is to set $\lambda_1 = 1$ and then vary λ_{reg} until some measure of registration accuracy is maximised.

However, these relative weightings are most likely only optimal for the particular dataset (and the particular modality) on which they were tuned. My favoured method of dealing with this is to follow the Bayesian approach outlined by Ashburner et al. (1999). Bayes' rule can be applied to image registration as follows:

$$p(\mathbf{w}|f, g) \propto p(f, g|\mathbf{w})p(\mathbf{w}), \quad (2.19)$$

where $p(\mathbf{w}|f, g)$ is the probability of observing warp parameters $\mathbf{w}$ given images f and g (posterior), $p(f, g|\mathbf{w})$ is the probability of observing images f and g given the warp parameters $\mathbf{w}$ (likelihood), and $p(\mathbf{w})$ is the *a priori* probability of observing warp parameters $\mathbf{w}$ (prior). Image registration can then be seen as finding the *maximum a posteriori* (MAP) estimate $\arg \max_{\mathbf{w}} (p(\mathbf{w}|f, g))$. Now, according to the Gibbs measure (or Boltzmann distribution):

$$p(\mathbf{w}|f, g) \propto e^{-E(\mathbf{w}|f, g)} \quad (2.20)$$

where $E(\mathbf{w}|f, g)$ is the energy associated with the state $(\mathbf{w}|f, g)$. Therefore, the MAP estimate corresponds to:

$$\begin{aligned} \arg \max_{\mathbf{w}} (p(\mathbf{w}|f, g)) &= \arg \min_{\mathbf{w}} (E(\mathbf{w}|f, g)) \\ &= \arg \min_{\mathbf{w}} (E(f, g|\mathbf{w}) + E(\mathbf{w})) \end{aligned} \quad (2.21)$$

where $E(f, g|\mathbf{w})$ is the Gibbs energy of the likelihood, or image similarity measure, and $E(\mathbf{w})$ is the Gibbs energy of the prior, or regularisation measure.

Under the assumption that g is only a spatially transformed version of f with the addition of Gaussian noise, the likelihood energy becomes (Ashburner et al. 2000):

$$E(f, g|\mathbf{w}) = \frac{1}{2\sigma^2} \sum_{\Omega} (f - g \circ \phi(\mathbf{w}))^2, \quad (2.22)$$

which is just the sum of squared differences between the reference and transformed moving image, scaled by half of the inverse of the variance of the noise σ^2 .

The takeaway is that this variance scaling can be used to normalise the likelihood of different datasets and modalities - ensuring that varying levels of noise and intensity distributions in images are accounted for. The true value of σ^2 is unknown, however a good approximation to it is the residual sum of squared differences after registration. In fact, the sum of squared differences of the previous iteration of an iterative optimisation strategy can be used to approximate σ^2 .

This methodology has an additional desirable effect. Early iterations will have a larger sum of squared differences than later iterations - thereby favouring the prior (or regularisation) term in the cost function. As the registration converges, the estimate of σ^2 decreases, effectively up-weighting the image similarity term in line with a growing confidence in the alignment. The relatively higher initial regularisation weighting has the benefit of helping to smooth out the optimisation landscape and avoid local minima. The relatively lower final regularisation weighting has the benefit of allowing the more complex deformations needed to match local image details. For these reasons I have therefore incorporated this variance scaling into MMORF.

Choosing a sensible default value for λ_{reg} was carried out for a unimodal T1 registration with $\lambda_{T1} = 1$. A grid search over a range of candidate values of λ_{reg} was

performed. I selected the best result as measured by the overlap of expert-segmented cortical regions from the Non-rigid Image Registration Evaluation Project (NIREP) (Christensen et al. 2006). Further details are provided in Chapter 3, Section 3.2.5.4. This serves as the default values of λ_{reg} for any registration using MMORF, however it can still be freely changed by the user if desired.

But the question still stands about what to do when adding another modality. I will start by considering some approaches adopted by other tools. ANTs (Avants et al. 2008) uses only global modality weightings, and ensures that, for N modalities:

$$\sum_{n=1}^N \lambda_n = 1$$

with a default weighting of $\lambda_n = 1/N$. Little advice is given regarding how relative weightings should be chosen, however.

DR-TAMAS (Irfanoglu et al. 2016) allows for locally varying weighting, and similarly ensures that:

$$\sum_{n=1}^N \lambda_n(\mathbf{x}) = 1.$$

DR-TAMAS also has a number of different weighting options depending on whether directional or intensity information should be preferentially optimised.

Ensuring that the weights always sum to 1 is sensible given the gradient descent with fixed step size optimisation approach of ANTs and DR-TAMAS. Consider a registration scenario in the absence of regularisation. In the case where two modalities are strongly spatially correlated (e.g., T1 and T2 images), their individual contributions to the cost and gradient of optimisation will generally be additive. If, for example, the weighting of each modality was allowed to be set to 1, then the optimiser would take larger steps (due to a larger gradient) and could become unstable.

However, restricting the weighting to sum to 1 comes with potential disadvantages too. Consider the case where two modalities contain highly orthogonal spatial information - e.g., one modality only has contrast within grey matter, and one only within white matter. The gradient of the cost function for each modality

would only be non-zero in those areas with non-zero contrast, and so provide complementary information for driving the alignment. By constraining the weights to always sum to 1, the gradient would be unnecessarily reduced and optimisation may be slowed leading to long convergence times.

In MMORF neither of these scenarios would cause an issue, as the Levenberg-Marquardt optimisation automatically compensates for larger gradients with larger Hessians - leading to reasonable step sizes. Furthermore, the variance scaling ensures that the same λ value is approximately equally meaningful when applied to any modality, and therefore unless there is a strong reason to do otherwise, the weighting of each modality can be kept equal.

The question then still remains as to whether or not the modality weights should sum to 1 in the presence of regularisation. At first it would seem that logic dictates that, as above, this should depend on whether the modalities provide confirmatory (correlated, therefore yes) or complementary (orthogonal, therefore no) information. But how much of an effect would incorrectly adding correlated modality weights *actually* cause? In reality, that would depend on how the regularisation scales with increased deformations. If the regularisation is linear (bending energy, Gaussian smoothing, etc.), then adding another modality with similar information will result in a deformation which is twice as large. Alternatively, if the regularisation cost scales nonlinearly with the size of the deformation, then the effect of “over-weighting” the similarity in some regions will be reduced. The default regularisation used by MMORF, and presented in Chapter 3, is *highly* nonlinear, and therefore should be robust to correlated cost functions while simultaneously being responsive to orthogonal cost functions.

As such I have chosen to leave all cost functions equally weighted by default, without enforcing the total weighting to sum to 1. However this is an entirely free parameter which users could vary for each modality separately, or even alter for each iteration individually.

2.8 Software Design

A key aim when developing MMORF was to ensure that it does not end up relegated to the catalogue of biomedical imaging tools which prove a scientific point well, but have no practical use beyond a journal article or PhD thesis. Furthermore, I wanted to be able to easily test the effect of switching out certain components of the optimisation framework without having to rewrite extensive regions of unrelated code.

The key method by which I achieved this was the concept of dependency inversion (Martin 2014). This states that:

1. High-level objects should not depend on low-level objects. Instead, they should both depend on abstractions.
2. Abstractions should not depend on details. Instead, details should depend on to abstractions.

These abstractions can be thought of as contracts between objects defining how they are allowed to interact.

In C++, the way to achieve this is through the use of abstract classes, also known as interfaces. Abstract classes define the expected parameters and return types, without specifying how the functions should be implemented.

Algorithms can then be defined and coded in terms of the capabilities of these abstract classes. Concrete realisations of the algorithms can then be created by substituting objects which implement the functions defined in the abstract classes.

In the following sections I will use Unified Modelling Language (UML) to describe the interfaces and their implementations, their relationships and interactions (OMG Object Management Group 2017).

2.8.1 UML Notation

UML uses a standard set of notation to describe the functionality, relationships and interactions between components of a model. Figure 2.3 provides a reference for the subset of UML notation used here.

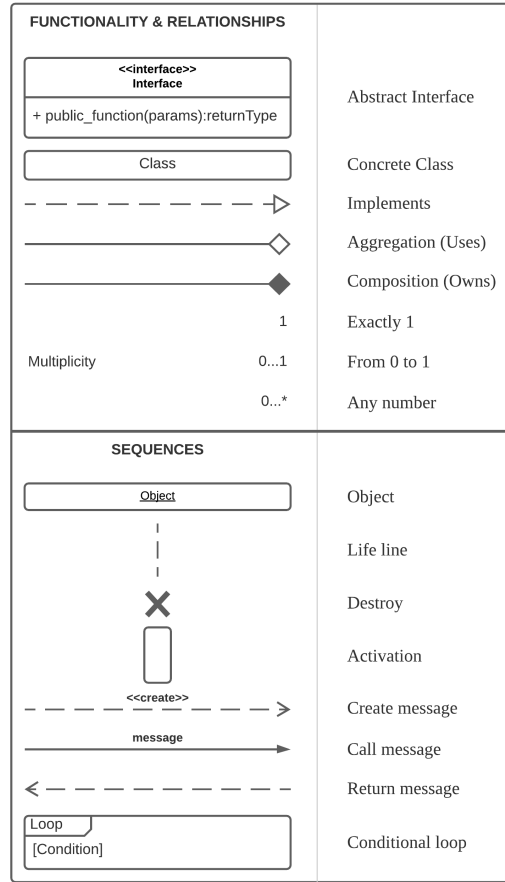


Figure 2.3: Basic UML symbols used to represent the functionality, relationships and interactions between objects.

2.8.2 Object Functionality

The software classes defining the objects upon which MMORF is built mirror the conceptual building blocks of any registration method, namely: transformations, cost functions, and optimisers. Here I will present some of the more important interfaces which define the functionality of the model elements, as well as some concrete class implementations thereof.

The central interface around which MMORF was designed is the `WarpField`, shown in Figure 2.4. The `WarpField` defines the parameter space controlling the registration and the physical extents over which deformations can occur and cost functions should be evaluated. No assumptions are made about what the parameters represent, only that they are accessible for reading and updating, which allows

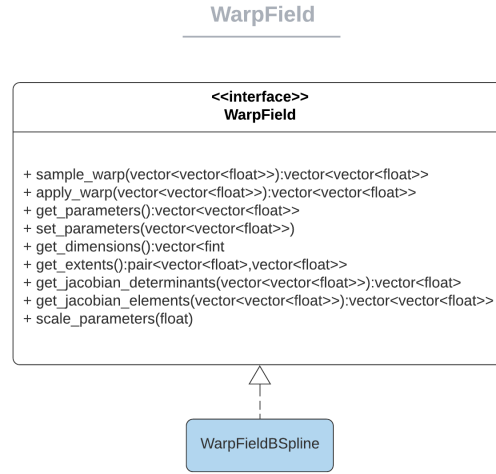


Figure 2.4: *WarpField* interface and a concrete implementation thereof. Even though I have only created one concrete implementation of the *WarpField* interface in the form of the *WarpFieldBSpline* class, MMORF has been designed and coded to be agnostic of how the warp field is being implemented internally. All functions associated with a warp field, such as sampling it, applying it or calculating the Jacobian determinant, do not depend on any underlying implementation decisions.

for a wide range of possible transformation models to be implemented if required. *WarpFields* are also responsible for actually applying a transformation to a set of points, or for sampling characteristics of the warp field (such as the local Jacobian matrix or determinant). One class adhering to the *WarpField* interface has been implemented - the *WarpFieldBSpline*. This class uses an underlying cubic B-spline representation, and the parameters control the B-spline coefficients. Internally all spline-related operations (e.g., sampling or differentiating of the warp field) are carried out using the excellent GPU accelerated *CubicInterpolationCUDA* library by Ruijters and Thevenaz (2012), making this class highly computationally efficient.

Figure 2.5 shows the *CostFxn* interface, as well as three groups of classes which implement it. The first (top left, orange) represent data consistency (or image similarity) cost functions for scalar and tensor images. The second (right, blue) represent regularisation cost functions. These are both “simple” cost functions as they implement the *CostFxn* interface directly. The third (bottom left, green) represent “compound” cost functions which combine one or more similarity and regularisation *CostFxn* objects into a single class. This is an important model in

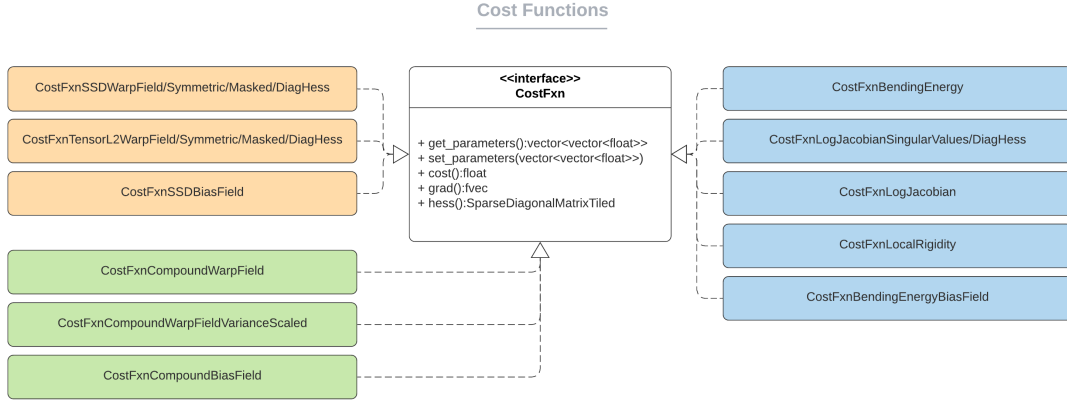


Figure 2.5: `CostFxn` interface and concrete implementations thereof. The orange classes (top 3 left) represent implementations associated with image similarity or data consistency. The blue classes (right) represent implementations associated with regularisation or statistical priors. These are both examples of “simple” cost functions - that is, these cost functions are each only associated with a single metric. The green classes (bottom 3 left) represent implementations associated with combining multiple metrics into a single cost. They are therefore “compound” cost functions as internally they may contain references to multiple simple cost functions. As an example, a regularised cost function based on the sum of squared differences and bending energy can be created as a `CostFxnCompoundWarpField` object which internally utilises a `CostFxnSSDWarpField` object and a `CostFxnBendingEnergy` object.

MMORF as it allows for a single, consistent way of calculating the cost, gradient and hessian regardless of how many modalities are being considered.

The `CostFxn` interface has been designed to be registration agnostic, and therefore does not feature any warp or image related functionality. The only application specific component of it is the use of the `SparseDiagonalMatrixTiled` return type for the Hessian. In practice, any matrix can be stored in this format, but the memory footprint will be suboptimal for non-sparse matrices. In this sense, MMORF should be seen as a sparse-matrix algebra optimised framework - which is a reasonable constraint as any cost functions which produce non-sparse Hessians quickly become unusable at all but the lowest resolution parametrisations (Ashburner and Friston 1999).

The main objects which interact with `CostFxn`s are those which implement the `Optimiser` interface, shown in Figure 2.6. An `Optimiser` only needs a single function to be implemented - appropriately called `optimise`. This takes a cost

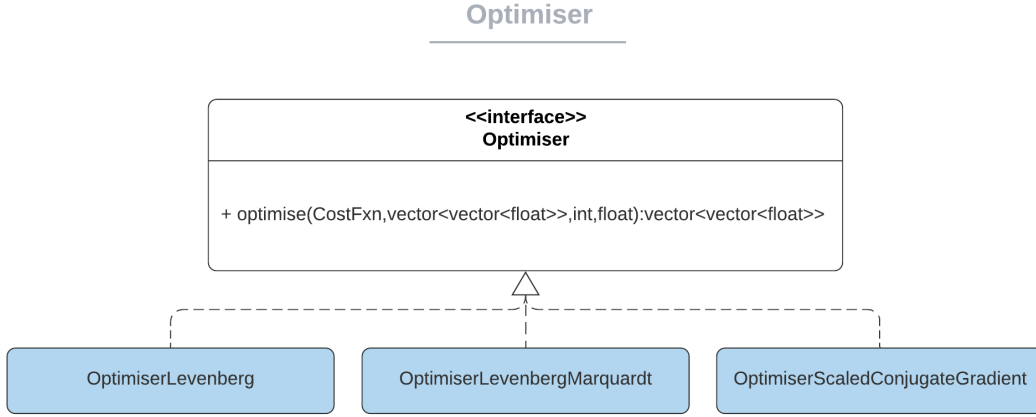


Figure 2.6: **Optimiser** interface and concrete implementations thereof. All class which implement the **Optimiser** interface need only support a single **optimise** function. This function takes as input a **CostFxn**, initial parameters, and the maximum number of iterations and relative tolerance used to test for convergence. Three implementations are currently available based on Levenberg (2nd order), Levenberg-Marquardt (2nd order) and scaled conjugate gradient (1st order) strategies.

function and initial parameter estimate and returns a set of optimised parameters. Again, this interface is entirely registration agnostic.

Internally, classes which implement the **Optimiser** interface using second order methods (such as **OptimiserLevenberg**) utilise a **LinearSolver** to compute parameter updates $\Delta \mathbf{x}$ by solving the system of linear equations defined by $\mathbf{H}\Delta \mathbf{x} = \mathbf{b}$. Figure 2.7 shows the **LinearSolver** interface, as well as two concrete classes which implement it. This is an example of how GPU acceleration can be selectively applied to parts of the overall registration framework. **LinearSolverCusp** uses the GPU accelerated CUSP library (Bell and Garland 2012) to iteratively solve for $\Delta \mathbf{x}$, whereas **LinearSolverArmadillo** uses the CPU only Armadillo linear algebra library (Sanderson and Curtin 2016) to perform the same task. In this way, taking advantage of the orders of magnitude speed up which CUSP provides over Armadillo is as simple as changing which class of linear solver an **Optimiser** uses.

There are numerous other interfaces within the MMORF framework, but these provide a good idea of how I carried out modelling of object functionality.

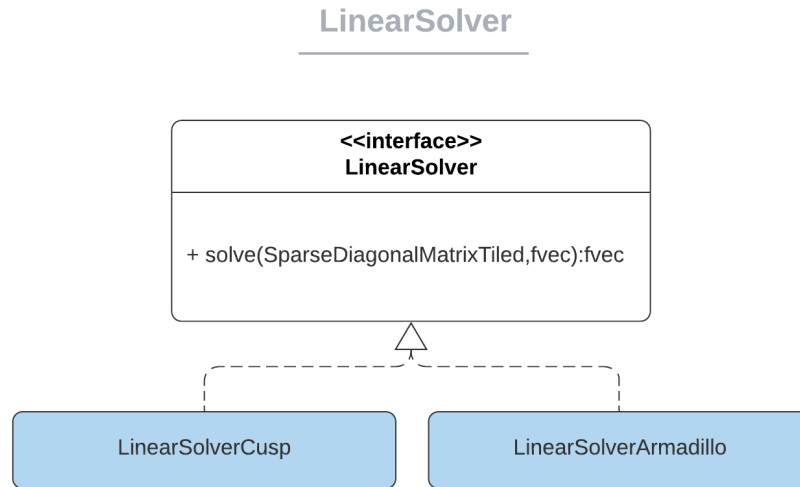


Figure 2.7: **LinearSolver** interface and concrete implementations thereof. Used to solve linear systems of the form $\mathbf{Ax} = \mathbf{b}$. It includes 2 implementations, one which uses the GPU accelerated Cusp library, and one which uses the highly efficient Armadillo CPU library. In practice, **LinearSolverCusp** is orders of magnitude faster than the CPU implementation and is used by default in MMORF.

2.8.3 Object Relationships

I will now give an example of the relationships between objects in MMORF, presented as a UML relationship diagram in Figure 2.8. This diagram includes some of the interfaces and classes presented in Section 2.8.2, as well as a few others. Interfaces are coloured in orange, classes which include GPU parallelisation are coloured in green, and classes which do not include parallelisation are coloured in blue. Two groupings of objects are highlighted by rectangular boxes, and represent objects associated with either warp field estimation, bias field estimation, or both.

At the top of the hierarchy is the **RegistrationCoordinatorMultimodal** class. Objects of this class are mainly responsible for creating the cost functions associated with the registration during their construction, and then for carrying out the registration algorithm primarily through calls to **Optimisers**. For the sake of brevity I have only shown the associative link (i.e. “uses a”) to the **Optimiser** interface, but note that the **RegistrationCoordinatorMultimodal** is dependent on all objects shown.

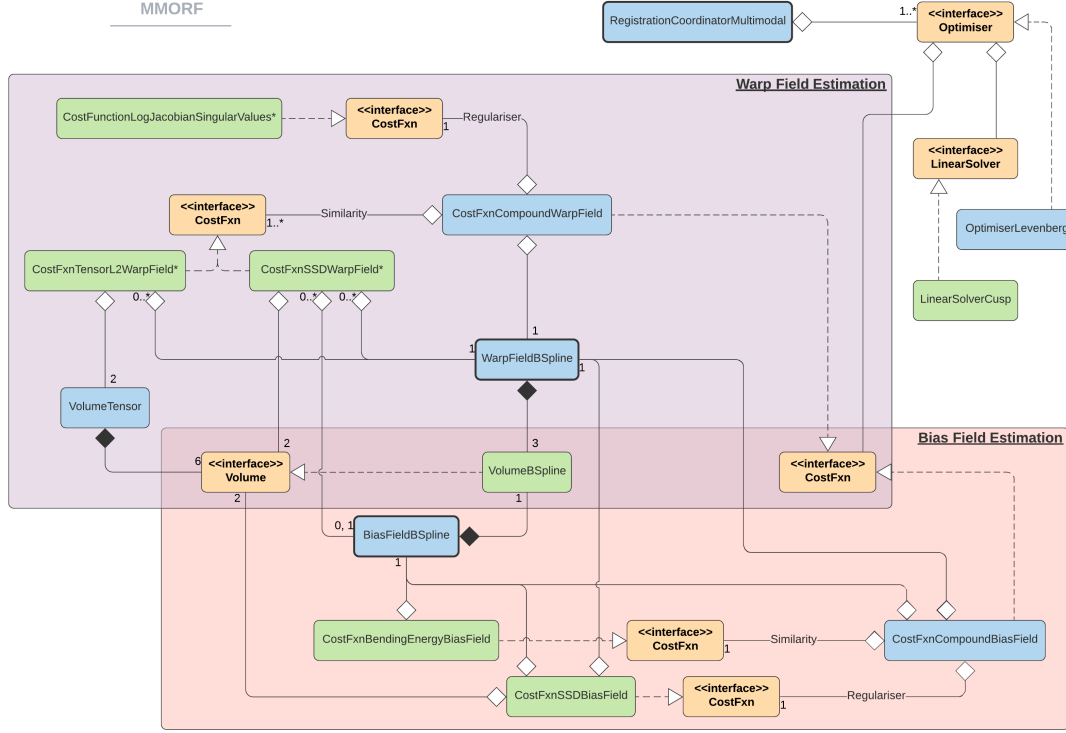


Figure 2.8: UML relationship diagram showing the relationships and dependencies between key classes during registration with MMORF. The two highlighted groupings demonstrate how the process of bias field estimation and warp field estimation are both posed in terms abstract interfaces. The `Optimiser/LinearSolver/CostFxn` interactions define a contract for optimising a set of parameters, which is realised for the two use cases by the different specific class implementations. Interfaces are orange, GPU parallelised classes are green, and non-parallelised classes are blue.

The important aspect to note here is that, as far as possible, the relationships between higher and lower level objects are mediated through abstract interfaces. For example, the `OptimiserLevenberg` object is related to the `CostFxnCompoundWarpField` object *only* through the contract established between the `Optimiser` and `CostFxn` interfaces. This epitomises the dependency inversion principle, and has been key to effectively implementing the mixed CPU/GPU architecture of MMORF, as well as flexibly changing the number, and types, of modalities being used.

2.8.4 Object Interactions

Sequence diagrams provide additional details about the interactions which make up the object relationships presented in Section 2.8.3. Figure 2.9 shows a sequence

2.9 Code Optimisation

In Chapter 3 I present an approach to parallelising the calculations associated with incorporating a novel regularisation penalty into the Gauss-Newton optimisation framework of MMORF, as well as providing an intuitive 2D example of the parallelisation process. However, there are a number of GPU-specific considerations that must be taken into account for the parallelisation to actually execute efficiently. Here I will demonstrate how I have addressed those considerations, as well as characterise the resulting code performance. Note that all GPU code output was tested for correctness against CPU code (written in MATLAB) that has previously been used to validate FNIRT (Andersson et al. 2007).

2.9.1 GPU Specific Considerations

I will begin by very briefly recalling some of the basic Nvidia CUDA terminology, described in more detail in Chapter 1, Section 1.9:

Host: The CPU and its associated memory

Device: The GPU and its associated memory

Kernel: The code to be executed in parallel on the GPU

Thread: The smallest logical execution unit of the device

Block: A group of threads

Warp: A set of 32 threads from the same block that are executed simultaneously on the device

Latency: The time taken to execute an instruction or read from memory

Global Memory: The largest but slowest memory space on the device, accessible to all threads

Register Memory: The fastest, most local memory, owned by a single thread

Shared Memory: Fast access memory shared between all threads within a block

Texture Memory: Fixed latency global memory cache that leverages 1D, 2D or 3D spatial locality of data

Occupancy: The ratio of the number of active warps to the maximum number of warps the device can handle

Memory Coalescing: Threads within a warp accessing consecutive location within memory

Warp Divergence: When threads within a warp do not follow the same branching pattern and must therefore have their execution serialised

Warp Efficiency: The average percentage of threads within a warp that are not serialised, and which are therefore simultaneously active

2.9.1.1 Occupancy

Occupancy is a critical factor in overall GPU code performance, as it directly determines the upper bound on how efficiently the GPU's compute elements are being utilised. A dominant factor in determining occupancy is the amount of register memory required by a particular kernel. If the kernel requires less than 32 registers per thread, then theoretical occupancy will always be 100 %. In such cases, kernels can be executed with the maximum number of threads per block supported by a particular device (usually 1024).

Complications arise when register counts exceed 32 though. In such cases, executing a kernel with the maximum number of threads per block results in only a single block being able to execute on the device at any one time, as two blocks would require more registers than are physically present on the device, and all threads within a block must be loaded at once, or not at all. This dramatically reduces occupancy, and subsequently code performance. It may therefore be better to reduce the number of threads per block thereby allowing maximal use of the available device resources.

Kernel register counts can be determined at compile time, and block sizes set explicitly to account for this, however hard-coding this may lead to brittle code. There is also no guarantee that future versions of compilers and devices might not alter what the optimum configuration is. I therefore made use of the built-in CUDA function `cudaOccupancyMaxPotentialBlockSize` to dynamically determine the best configuration at runtime.

Using this approach, all kernels have a maximum occupancy of between 62.5% and 100%, with achieved occupancies near the theoretical limits. I did experiment with forcing the compiler to never exceed 32 registers per thread, however the increased latencies due to register spillage (i.e. using global memory as a swap space) nullified any gains in occupancy. Note also that the lowest occupancy kernel still had a factor of 300 speedup, and the most critical kernel in terms of execution times had a theoretical occupancy of 75%.

2.9.1.2 Memory Access

I have spent considerable time ensuring memory access is as optimised as possible, and I will therefore present some of the key design decisions that I found to improve performance.

The first consideration was memory coalescing when saving results of gradient and Hessian calculations. This is trivial when writing the result of gradient calculations if the parallelism is performed in a “thread per gradient element” regime and therefore this was an obvious approach to take.

Calculating the Hessian is considerably more complicated problem though. As the Hessian is incredibly sparse, the choice of matrix format will have a significant impact on the pattern of memory access. I therefore selected a sparse diagonal format, where values along any non-zero diagonal of the Hessian are stored sequentially in memory. For sparse matrices with a largely diagonal structure, this format has been shown to be highly efficient (Bell and Garland 2008). Details are shown in Figure 2.10. By assigning each diagonal to a block, I could then ensure that the threads within that block were always writing to sequential memory addresses resulting

in near-perfectly coalesced memory write-access. Note that I choose to store all non-zero diagonals, rather than only the symmetrically unique ones. While this increases the memory requirements for storing the Hessian, it does not significantly increase computation time as symmetric values are still only calculated once. The gain is that Hessians with the same sparsity pattern, but different amounts of redundancy, can be stored in the same format and can therefore be operated on (e.g., added together) more easily. This is a situation which occurs frequently when using cost functions which are functions of derivatives of B-splines rather than B-splines themselves (see Chapters 3 and 4)

An additional benefit of the sparse diagonal format is that it allows for highly efficient matrix by vector multiplication with perfect coalescence - an operation which is performed many times when solving sparse systems of linear equations to compute warp field updates. Note that while I did not end up using my own implementation of sparse matrix vector multiplication implementation (as the existing methods in the CUSP library were sufficiently efficient), I was able to achieve a $5\times$ performance increase over a CPU implementation.

The second consideration was reading of the transformation basis functions. Many portions of the code require a representation of the cubic B-splines, or potentially a spatially differentiated version thereof. Calculating the gradient of the mean-squared error in a “thread per element” approach is equivalent to a strided convolution of the B-spline basis function with a spatially differentiated moving image. As such the B-spline need only be stored once per block as its values are identical irrespective of where the convolution is occurring. This is a perfect use case for shared memory, allowing the values of the spline to be read from global memory only once (in a perfectly coalesced manner) and then shared by all threads. Furthermore, the separable nature of B-splines was used to reduce memory usage by storing only 1D splines and then performing tensor-product multiplication on-the-fly. The same approach was used in the Hessian calculation, where the values along any diagonal can be similarly modelled as a strided convolution between the spatial overlap of two splines and a gradient image.

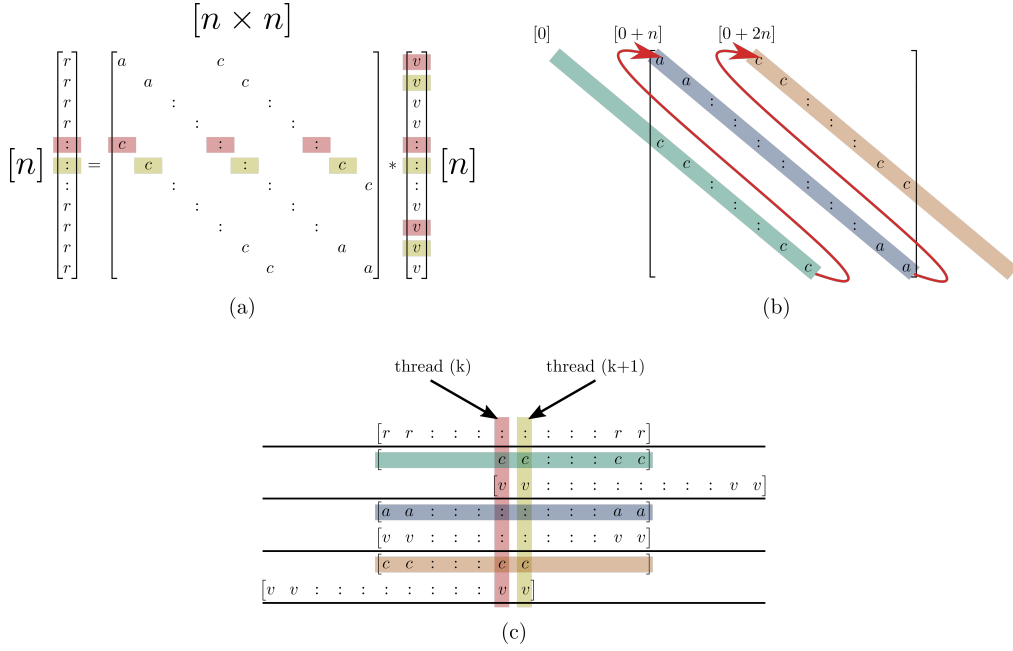


Figure 2.10: Global Memory access using a sparse diagonal matrix. Consider a sparse, diagonal matrix of dimension $n \times n$, multiplied by a right hand side (rhs) vector of dimension $n \times 1$, the product of which is an $n \times 1$ result vector. Assume too that this matrix is symmetric about its main diagonal. This configuration is shown in (a). Now, consider the case where only the main diagonal, and a single diagonal above and below are non-zero. One such configuration is shown in (b), where only the a and c diagonals are highlighted. Next, assume that both the rhs and result vectors are stored in separate, contiguous areas of Global Memory. Finally, imagine that the non-zero diagonals of the matrix are stored sequentially in Global Memory, with zero padding before (below main diagonal) or after (above main diagonal) such that the offset between elements in the same row of adjacent diagonals is n . If each thread launched is responsible for calculating one element of the result vector, then the data access pattern for each thread is as shown in (c). By looping sequentially through each diagonal's contribution to the result vector (which is represented by the values between the solid horizontal lines in (c)), adjacent threads will always be reading from, and writing to, contiguous areas of Global Memory. Therefore, all memory operations are coalesced, and Global Memory efficiency is maximised.

The final consideration was reading in the volumetric gradient images during the convolutions. Strided memory access immediately precludes coalescence unless the stride is accounted for pre-emptively in the way the data is stored. For example, for 3 sequential threads to access the true sequence:

$$\{1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

in a coalesced manner with a stride of 3, it would need be stored in memory as:

$$\{1, 4, 7, 2, 5, 8, 3, 6, 9\}$$

Whilst this is doable, I chose to employ a more elegant approach using texture memory. Texture memory is a fixed latency cache that is optimised for spatially local data, and therefore well suited to the problem of strided convolutions. Fixed latency implies that a memory read always takes the same amount of time regardless of coalescing - but a cache hit still saves data from having to be transferred from global memory. Therefore, for perfectly coalesced memory access texture memory would always be slower than global memory. However, for non-coalesced memory access this is offset by the fact that a cache hit reduces the burden on global memory bandwidth - which would otherwise quickly become the limiting factor in most GPU kernels with strided memory access patterns. Furthermore, by launching sufficient warps (*i.e.* when combined with high occupancy), latency can effectively be automatically hidden by allowing other warps to execute whilst one warp waits for data. I tested the effectiveness of this approach by replacing all texture memory reads with constant values (effectively eliminating the latency, but maintaining the number of computations) and saw minimal impact on the overall performance, indicating that the latency/bandwidth trade-off had the desired effect. Note that for some more recent GPU architectures, L1 caching may provide similar performance improvements for non-coalesced data access, however the method I have employed here will be effective across the vast majority of the GPUs currently in use.

2.9.1.3 Warp Efficiency

Warp divergence is a critical aspect to minimise in order to attain high levels of computational throughput, and the decision to use a sparse diagonal matrix format for the Hessian benefits this. Consider the non-zero elements of one row of the Hessian. Each one can be mapped to a pattern of spatial overlap of the two B-splines corresponding to that point in the matrix. When traversing a row, the spatial overlap pattern changes for each non-zero element, and therefore would require a different pattern of data access. If threads within a warp were arranged per-row, then the control flow within each thread would differ, and the calculations would be serialised. In the worst case scenario, this could lead to an up to 32 times reduction

in performance. However, it can be noticed that the pattern of spatial overlap along any diagonal of the Hessian is constant. Therefore, if the threads within a warp are assigned to a diagonal then they require exactly the same control flow to correctly index into the B-splines, and warp divergence is effectively eliminated. By employing this strategy I achieve average warp efficiencies close to the theoretical optimum.

2.9.1.4 Additional Considerations

There are a number of smaller optimisations that have been made that I will now briefly describe for completeness.

Image sampling and interpolation: All image sampling and interpolation (including image gradients and warp sampling) is performed on the GPU device using the `CubicInterpolationCUDA` code by Ruijters and Thevenaz (2012).

Data transfer: The images being registered, the warp representation and the Hessian are all stored on the GPU device, reducing the need for expensive host-to-device and device-to-host data transfers.

Linear Solver: Gauss-Newton update steps are calculated in a parallelised manner on the device, and employ the CUSP (Bell and Garland 2012) iterative linear solver to solve the required system of linear equations. Note that CUSP uses a compressed sparse row matrix representation, and I therefore implemented a kernel for converting to this format from my sparse diagonal Hessian representation.

2.9.2 Profiling Results

It is worth quantifying some aspects of MMORF’s performance through code profiling, to demonstrate the degree to which execution time benefits from GPU parallelisation. To this end, I utilised the NVIDIA Visual Profiler (NVVP) to time the execution of select kernels, as well as an exemplar full registration at a variety of warp resolutions.

A single, scalar image pair was registered using a mean-squared (MSQ) similarity metric and the SPRED regularisation penalty described in Chapter 3. Note that increasing the number of scalar image pairs would linearly increase the runtime of the kernels associated with scalar cost functions. The nature of the tensor cost function used to align DTI images (described in Chapter 4) is such that the kernels used are identical to those for the SPRED penalty, and therefore adding tensor modalities would increase the runtime proportional to the kernels associated with the regularisation term in this comparison.

CPU code was executed on an Intel Core i7-6700HQ processor with 16GB of RAM. GPU code was executed on an NVIDIA 960M GPU with 2GB of RAM.

I assessed both the overall proportion of the total registration runtime assigned to each calculation type (gradient, Hessian etc.) as well as the speedup in performance.

2.9.2.1 Overall Performance

As I have used a multiresolution registration approach in MMORF, the proportion of time spent on each calculation changes depending on the current warp resolution. I present here the overall performance for isotropic knot spacings of 40, 30, 20 and 10 mm in Table 2.1 The key points to take away from these results are:

- The high GPU utilisation (particularly for higher resolution warps) suggests that the major computational bottlenecks have been effectively assigned to the GPU.
- The SPRED/DTI penalty, and in particular the Hessian calculation, is by far the most computationally intensive part of the algorithm.
- The algorithm is not being bottlenecked by excessive data transfers.
- Solving the system of linear equations to determine the Gauss-Newton update step (via a GPU parallelised iterative linear solver) is not a computational bottleneck.

Knot Spacing (mm)	Runtime (s)	GPU Utilisation (%)	Data Transfer (%)	SPRED/DTI Hessian (%)	MSQ Hessian (%)	Gradient (%)	Sparse Matrix Convert (%)	Linear Solver (%)
40	3.5	46.0	22.8	91.2	0.2	1.3	3.5	1.4
30	5.8	59.0	14.4	91.6	0.1	0.7	3.8	1.4
20	9.4	76.9	7.4	96.5	0.1	0.4	0.8	0.7
10	45.0	83.2	3.3	93.2	0.1	0.2	3.0	1.3

Table 2.1: Performance for 5 Gauss-Newton iterations of optimisation at varying isotropic warp resolutions (or knot spacings). Key points to note are that the GPU utilisation increases with warp resolution, whilst the proportion of time needed for data transfer decreases. This indicates that the GPU is being used most effectively when it is likely to have the biggest impact. Additionally calculating the Hessian of the SPRED/DTI penalty is by far the most computationally intensive portion of the algorithm. Iterative inversion of the Hessian (via a GPU parallelised iterative solver) accounts for less than 2% of the GPU compute time. This rises to approximately 5% of the compute time when factoring in a necessary conversion from my own internal sparse diagonal format to the solver’s required compressed sparse row format. Note that the somewhat different 20 mm performance is a result of, entirely by chance, requiring fewer Levenberg iterations at that resolution (i.e., update steps more often led to a decrease in the cost function without having to increase the LM λ value and recalculate the update), and consequently fewer sparse matrix conversions and linear solver iterations were performed.

2.9.2.2 Kernel Specific Performance

Given an understanding of the relative importance of the GPU kernels within the entire registration, I examined the performance of selected kernels compared to reference CPU implementations. The CPU testing was coordinated in MATLAB, but utilises highly optimised compiled C code to perform the actual calculations. I have verified that the outputs of both methods are identical. Note that I did not implement the entire registration algorithm for the CPU, merely selected functions for testing purposes.

First, observe in Figure 2.11 that the gradient calculation ($\partial \mathbf{C} / \partial \mathbf{w}$ for both the similarity and regularisation metrics) has a speedup in the region of an order of magnitude. This is in line with a similar spline-based GPU method by Modat et al. (2010). It should be noted that the overall performance is not overly dependent on this calculation in a Gauss-Newton framework.

Second, I consider the major computational bottleneck identified above, namely the calculation of the Hessian. Note that I have used two separate algorithms, one

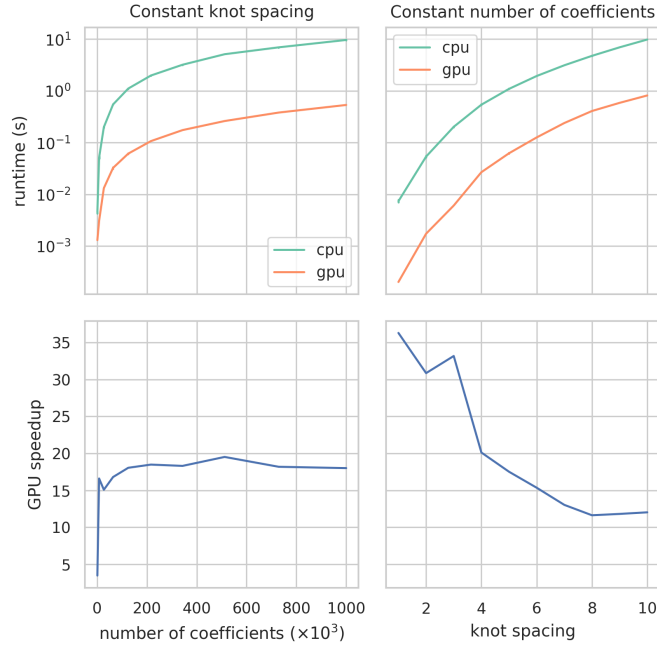


Figure 2.11: Runtime and speedup of GPU vs CPU implementations of the gradient calculation with respect to the B-spline warp parameters. The left column shows results when knot spacing is kept constant at 5 mm isotropic, and the number of spline coefficients in each dimension are varied. The right column shows results when the number of spline coefficients in each dimension is held constant at 50 isotropic, and the knot spacing is varied. In all cases the images are sampled at 1 mm isotropic.

for the MSQ metric and one for the SPRED/DTI metric. The difference between the methods is due to the fact that I am able to exploit additional levels of symmetry in the MSQ Hessian that are not present in the SPRED/DTI Hessian. Observe in Figure 2.12 that the speedup is in the region of $30 - 40\times$ for the MSQ Hessian calculation. Consequently the MSQ Hessian is not rate limiting to the registration.

There are a number of reasons why the SPRED/DTI Hessian is by far the most important calculation in terms of execution time. Firstly, as seen in Figure 2.13, the speedup is only about half that of the MSQ Hessian at around $15 - 20\times$ for practical combinations of knot spacing and coefficient dimensions. This is largely attributable to two reasons, namely:

- The MSQ Hessian has a theoretical 100 % GPU utilisation that is reduced to 75 % for the SPRED/DTI Hessian due to the second algorithm requiring more registers per thread.

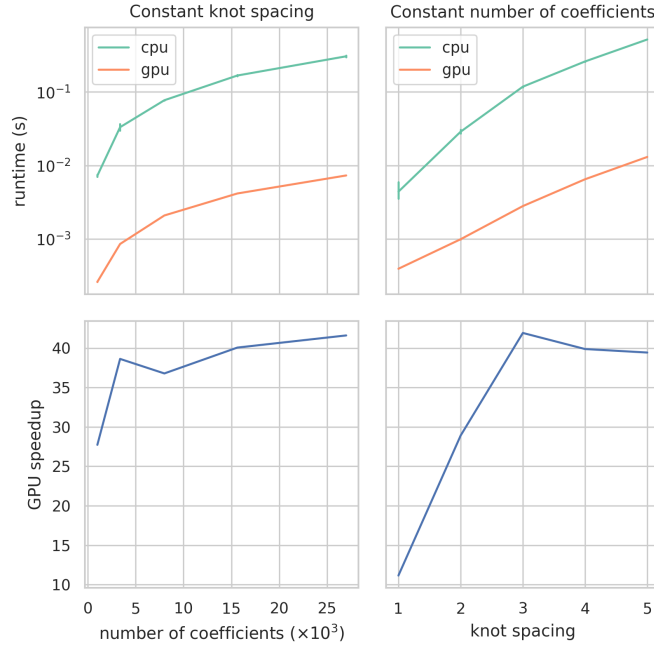


Figure 2.12: Runtime and speedup of GPU vs CPU implementations of the Hessian calculation for the MSQ metric. The left column shows results when knot spacing is kept constant at 2 mm isotropic and the number of spline coefficients in each dimension are varied. The right column shows results when the number of spline coefficients in each dimension is held constant at 15 isotropic and the knot spacing is varied. In all cases the images are samples at 1 mm isotropic.

- The SPRED/DTI Hessian requires more memory reads per element of the Hessian, which introduces additional latency costs on the GPU.

Secondly, the SPRED/DTI Hessian calculation is called $9\times$ more often than the MSQ Hessian, due to cross-terms introduced by the partial derivative of the cost functions with respect to the Jacobian matrix $\frac{\partial \mathbf{C}^\top}{\partial \mathbf{J}} \frac{\partial \mathbf{C}}{\partial \mathbf{J}}$.

Finally, the reduction in symmetry introduces $8\times$ more calculations per SPRED/DTI Hessian compared to the MSQ Hessian. In future, I will look at increasing the speedup of the SPRED/DTI Hessian further, potentially by deconstructing the algorithm into multiple smaller computational units that would be better able to hide the inherent latencies in the calculations.

The result of these optimisations is that, for isotropic 1 mm human brain data, a unimodal scalar registration can be completed in under 15 min, a unimodal tensor

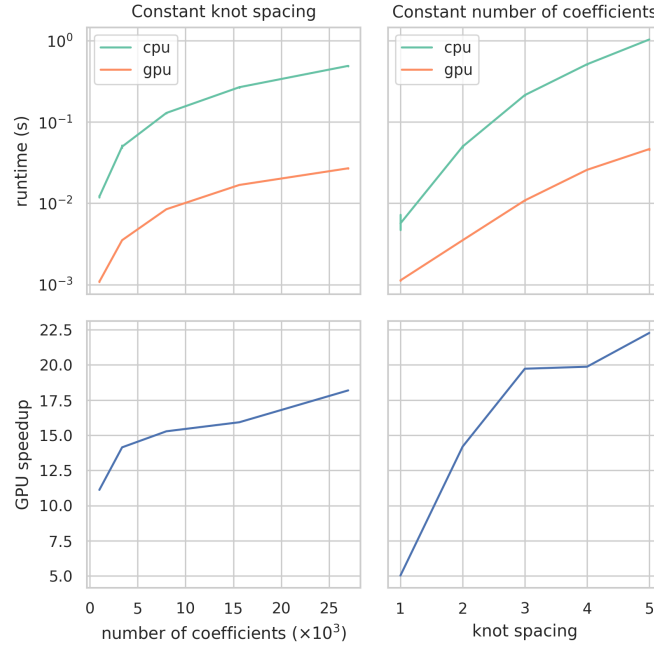


Figure 2.13: Runtime and speedup of GPU vs CPU implementations of the Hessian calculation for the SPRED/DTI regularisation metric. The left column shows results when knot spacing is kept constant at 2 mm isotropic and the number of spline coefficients in each dimension are varied. The right column shows results when the number of spline coefficients in each dimension is held constant at 15 isotropic and the knot spacing is varied. In all cases the images are samples at 1 mm isotropic.

registration in under 30 minutes and a multimodal scalar and tensor registration in under 45 minutes.

Overall, I am pleased with the computational performance of MMORF - particularly with respect to how GPU parallelism has been used to overcome computational bottlenecks. CPU parallelism, using either OpenMP² or MPI³, could be used to improve the relative performance of the CPU implementation, and this would be an interesting avenue to consider in the future. Converting my current implementation to a CPU parallelised version would be a complex task though, and as such has not been carried out at this time. However, the software engineering approach I have taken while developing MMORF means that if a future CPU version is sufficiently in demand, then it can be included in the same framework by ensuring it conforms to the same interface models I have used. Other potential

²<https://www.openmp.org/>

³<https://www.mpi-forum.org/>

future avenues of investigation include the relative energy and monetary costs of GPU versus CPU implementations - the former becoming an increasingly important metric given the current climate crisis.

While there are still opportunities to further speed up execution, the current implementation strikes a good balance between efficiency and maintainability. The codebase is both sufficiently performant and robust enough for me to confidently recommend it be used for real neuroimaging studies.

2.10 Conclusion

In this chapter I presented several of the key design and implementation decisions that I have made while writing the software for MMORF. I have taken a software engineering driven approach at all levels of the design - breaking up the registration optimisation problem into abstract logical objects with well defined functionality, relationships and interactions. The resulting software framework is highly modular, with very few implementation-specific dependencies, allowing components which share a common interface to be easily added or swapped out. This means I have been able to focus the complex task of GPU parallelisation only at those places where where it will be *maximally* effective in relieving computational bottlenecks.

The end result is a well engineered registration method which is capable of aligning an arbitrary number of scalar and tensor modalities in a computationally efficient GPU parallelised framework. In Chapter 3 I will show how the increased computing power offered by GPUs has allowed me to implement a highly complex regularisation strategy to improve the anatomical plausibility of registration results - a penalty that would otherwise have been too computationally complex to be of practical use. In Chapter 4 I will show how the same principles can be applied to the registration of tensor-valued data leading to warps which are informed by both intensity and directionality information in a computationally efficient manner. In Chapter 5 I will show how MMORF can be used to create an unbiased multimodal template, and give examples of templates I have created for the lemur, macaque and human. Finally, in Chapter 6 I will then demonstrate how the

combination of these methods leads to an overall improvement in anatomical consistency following registration.

3

A Symmetric Prior for the Regularisation of Elastic Deformations

Contents

3.1	Introduction	85
3.2	Methods	89
3.2.1	Theory	89
3.2.2	Intuition	96
3.2.3	Implementation	97
3.2.4	2D Example	100
3.2.5	Testing	101
3.3	Results	111
3.3.1	Overlap Scores	111
3.3.2	Jacobian Determinant Distributions	115
3.3.3	CVAR Distributions	118
3.3.4	Spatial Maps	119
3.4	Discussion	126
3.4.1	Relationship to Earlier Work	128
3.4.2	Beyond Enforcing Diffeomorphic Warps	129
3.4.3	Other Parallelised Algorithms	133
3.4.4	Registration Accuracy	134
3.4.5	Anatomical Plausibility	134
3.4.6	Performance Summary	136
3.5	Conclusion and Outlook	136

3.1 Introduction

This chapter is based largely on the work published in F. J. Lange et al. (2020a). As I was the lead author on that paper, much of what appears here is identical to what can be found there.

As stated in earlier chapters, existing volume based methods can broadly be categorised based on:

Construction of warps As part of the iterative process a warp can be updated by adding an update, or by warping the previous warps by an update. The former is often referred to as an *elastic* or a *small deformation* framework, as opposed to a *plastic* or *large deformation* framework for the latter (M. Miller et al. 1997). Note that for infinitesimally small displacements, composition of warps can be extended to the concept of integrating a velocity field (Beg et al. 2005).

Regularisation of warps Warps can be *regularised*, i.e., have smoothness imposed on them, by adding a penalty term to the cost-function, or by explicitly smoothing the updates.

Similarity measure This is a scalar measure that assesses the similarity/difference between the images and whose maximisation/minimisation drives the registration.

The present chapter is concerned mainly with the second of these, the regularisation. I will, however, touch upon the construction of warps and the effect of similarity metric choice as well. A thorough review of these categorisations can be found in Sotiras et al. (2013).

Historically one important aspect of regularisation in the small deformation framework has been to attempt to ensure that the warps stayed one-to-one and onto i.e., that each point in image A mapped to a unique point in image B, and that each point in image B could be reached from image A. This also means that the

mapping is invertible. This was primarily done by adding a penalty term to the cost-function such that one simultaneously minimised the difference between the images (dissimilarity) and some function of the warps (regularisation). That function would often be borrowed from mechanics, such as bending energy (Bookstein 1997b), membrane energy (Amit et al. 1991) or linear-elastic energy (M. Miller et al. 1993) (whose formulations are summarised in Ashburner and Friston (1999)). However, none of those functions had any specifically biological relevance to the problem of mapping individual brains to each other. Moreover, they do not explicitly guarantee invertibility. So it was common practice to either ignore the issue of invertibility, or to empirically calibrate the relative weights of the dissimilarity and regularisation penalties so that the resulting warps were (almost) always invertible. But that typically meant a quite high weight for the regularisation term which resulted in an inability to properly model large deformations.

This limitation led to the development of the large deformation framework where warps are updated by warping (resampling) the previous warp by an update warp (equivalent to a composition of multiple small warps). If both the previous and the update warps are invertible, this guarantees that the composed warp is also invertible. Therefore, as long as one ensures that each update warp is invertible the end result will be too. Furthermore, certain methods (such as LDDMM (Beg et al. 2005) and Geodesic Shooting (Ashburner and Friston 2011)) seek solutions where the overall set of composed warps (or integrated velocity field) are as “smooth” (according to some criterion) as possible, rather than greedily smoothing at each step. The result is then that these methods are able to model large deformations while remaining invertible by taking many small update steps, leading to better matching than could be achieved within the small deformation framework.

However, invertibility is not the be-all and end-all of well regularised warps. It *does* restrict the space of allowed warps, but some form of regularisation is still needed for choosing the optimal warp within that space. Moreover, it is no more clear in the large than in the small deformation framework how exactly that regularisation of the warps/velocity fields should be performed. Even when maintaining invertibility

one can, and frequently does, still obtain warps that correspond to implausibly large local changes in volume and/or shape (Ashburner and Ridgway 2013; Mang and Biros 2016; Coalson et al. 2018). For a more detailed discussion of some of the tools which address invertibility, see Section 3.4.2.

Here I show how using the small deformation framework of MMORF, together with a penalty function that explicitly penalises changes in both volume and shape, leads to state-of-the-art registration quality and warp regularity. I will refer to this penalty as the Symmetric Prior for the Regularisation of Elastic Deformations (SPRED). It centres on penalising the singular values s_i of the local Jacobian matrix $\mathbf{J}$ of the transformation ϕ according to:

$$\mathcal{C} = \sum_{i=1}^3 \log^2 s_i (\mathbf{J}(\phi)) \quad (3.1)$$

This penalty approaches infinity as relative volumes approach zero (which would break invertibility), which means that it can be given a small weight in order to allow large deformations and still guarantee diffeomorphic warps. Hence, it addresses the same issue of invertibility as LDDMM based methods *as well as* the issue of how to find warps with desirable characteristics within that space. It should be stressed at this point that the penalty I propose is not the same, or even particularly similar to, previously suggested penalties based on some function of the Jacobian determinant (see for example Rohlfing et al. (2003) or Leow et al. (2007)). I should also be clear that even though I refer to this as an elastic deformation, the prior is not based on any linear elastic model with the accompanying small deformation restriction.

The mathematical formulation and theoretical basis of the penalty are explained in Section 3.2.1. It has been proposed in the past (Ashburner et al. 1999; Ashburner et al. 2000), but the computational cost is such that at the time it was not practically feasible to consider using it as part of a real registration tool. With the advent of *general-purpose computing on graphics processing units* (GPGPUs) that is no longer the case, and I describe my implementation, using NVIDIA’s CUDA framework (NVIDIA 2019), of a registration algorithm based on this penalty function in Section 3.2.3.

The rationale behind the suggested penalty function is twofold,

- To minimise geometric distortions, *i.e.* changes in both shape and volume, and to ensure the warps are one-to-one and onto.
- To be symmetric in the sense that a geometric (lineal, areal or volumetric) change by a factor of 2 is penalised the same as one by a factor of 0.5.

The first property comes from the notion that brain tissue implements function, be that function integration of signal in the grey matter or transmission of signal in the white matter. The implementation of that function will necessarily occupy some physical space and it would seem unlikely that a given function can be implemented in a very small fraction of space in one brain compared to in another. A penalty should therefore favour no volume changes at all. It should also rapidly increase as the relative volumes become unrealistic, approaching infinity as the volume changes get close to breaking the one-to-one and onto criteria, thereby never allowing negative relative volumes. But it should also prevent unrealistic shape changes. Given what is known about the structure of both grey and white matter it is hard to picture a situation where a function implemented in a $1 \times 1 \times 1 \text{ mm}^3$ cube in one brain might occupy a $0.01 \times 10 \times 10 \text{ mm}^3$ sheet or a $0.1 \times 0.1 \times 100 \text{ mm}^3$ stick in another. It is here that I note an important observation - that this can all be reduced to penalising lineal changes since that would automatically also penalise shape and volume changes.

The second property, symmetry, comes from the desire that a registration algorithm should be inverse consistent, *i.e.*, that the transform that is obtained from registering image A to image B is the inverse of that obtained from registering B to A (Christensen 1999). A necessary, but not sufficient, condition for this is a prior that is symmetric in terms of penalising an expansion from A to B equally to the corresponding contraction from B to A.

3.2 Methods

3.2.1 Theory

Here I present the formal concepts underpinning the SPRED penalty, as well as the mathematics for its incorporation into MMORF's optimisation strategy. For an intuitive explanation of these concepts, refer to Section 3.2.2.

3.2.1.1 Jacobian Matrix

Consider the generic registration problem presented in Chapter 1, where a moving image $g(\mathbf{x})$ is registered to a reference image $f(\mathbf{x})$. Here I will specifically consider the case where $\mathbf{x} \in \mathbb{R}^3$, such that:

$$\mathbf{x} = [x, y, z]^\top$$

Any fixed, nonlinear deformation $\phi(\mathbf{x}) = \mathbf{x} + \mathbf{d}(\mathbf{x})$ can then be written as:

$$\begin{aligned}\phi(\mathbf{x}) &= [\phi_x(\mathbf{x}), \phi_y(\mathbf{x}), \phi_z(\mathbf{x})] \\ &= [x + d_x(\mathbf{x}), y + d_y(\mathbf{x}), z + d_z(\mathbf{x})]\end{aligned}$$

If $\phi(\mathbf{x})$ defines a continuous, differentiable function on $\Omega \in \mathbb{R}^3$, then one can calculate a local Jacobian matrix $\mathbf{J}(\mathbf{x})$ at any point in Ω , with:

$$\mathbf{J}(\mathbf{x}) = \frac{\partial \phi(\mathbf{x})}{\partial \mathbf{x}} \quad (3.2)$$

$$= \begin{bmatrix} \frac{\partial \phi_x(\mathbf{x})}{\partial x} & \frac{\partial \phi_x(\mathbf{x})}{\partial y} & \frac{\partial \phi_x(\mathbf{x})}{\partial z} \\ \frac{\partial \phi_y(\mathbf{x})}{\partial x} & \frac{\partial \phi_y(\mathbf{x})}{\partial y} & \frac{\partial \phi_y(\mathbf{x})}{\partial z} \\ \frac{\partial \phi_z(\mathbf{x})}{\partial x} & \frac{\partial \phi_z(\mathbf{x})}{\partial y} & \frac{\partial \phi_z(\mathbf{x})}{\partial z} \end{bmatrix} \quad (3.3)$$

$$= \begin{bmatrix} \left(1 + \frac{\partial d_x(\mathbf{x})}{\partial x}\right) & \frac{\partial d_x(\mathbf{x})}{\partial y} & \frac{\partial d_x(\mathbf{x})}{\partial z} \\ \frac{\partial d_y(\mathbf{x})}{\partial x} & \left(1 + \frac{\partial d_y(\mathbf{x})}{\partial y}\right) & \frac{\partial d_y(\mathbf{x})}{\partial z} \\ \frac{\partial d_z(\mathbf{x})}{\partial x} & \frac{\partial d_z(\mathbf{x})}{\partial y} & \left(1 + \frac{\partial d_z(\mathbf{x})}{\partial z}\right) \end{bmatrix} \quad (3.4)$$

Where $\partial d_i / \partial j$ is the partial derivative of the i^{th} component of $\mathbf{d}$ in the j^{th} direction.

$\mathbf{J}$ then describes how the image is locally scaled and rotated by the action of ϕ . These actions can be decoupled by applying singular value decomposition (SVD) to $\mathbf{J}$ (Ashburner et al. 1999; Ashburner et al. 2000), such that:

$$\mathbf{J} = \mathbf{U}\mathbf{S}\mathbf{V}^T \quad (3.5)$$

$\mathbf{U}$ and $\mathbf{V}$ are unitary matrices describing the rotational effect of $\mathbf{J}$, and will not be considered further as they do not geometrically distort the image in any way. $\mathbf{S}$ is a diagonal matrix containing the singular values and represents the orthogonal scaling effect of $\mathbf{J}$. It is the individual elements s_i of $\mathbf{S}$ which will be penalised in the regularisation, and I will refer to them as the Jacobian Singular Values (JSVs).

3.2.1.2 The SPRED Penalty Function

The penalty is based on the prior belief that the JSVs are drawn from a lognormal distribution. Such a penalty meets the biological plausibility arguments set out in Section 3.1 as:

- $s_i = 1$ is most likely
- $s_i = 0$ and $s_i = \infty$ are infinitely unlikely
- $s_i = a$ and $s_i = \frac{1}{a}$ are equally likely

Now, I define $C(\mathbf{w})$ as the penalty (or *cost*) associated with $\mathbf{w}$:

$$C(\mathbf{w}) = \sum_{n=1}^N c_n(\mathbf{w}) \quad (3.6)$$

$$= \sum_{n=1}^N v(1 + |\mathbf{J}_n|) \sum_{i=1}^3 \log^2 s_i \quad (3.7)$$

Where:

C = total cost/penalty

$\mathbf{w}$ = transformation parameters

N = total voxels in image

c_n = cost/penalty at voxel n

v = voxel volume

$\mathbf{J}_n = \mathbf{J}_n(\mathbf{w})$ = Jacobian matrix at voxel n

$s_i = s_i(\mathbf{w}) = i^{\text{th}}$ singular value of $\mathbf{J}_n$

In the same way that penalising s_i^2 is equivalent to assuming a normally distributed prior for s_i , recognise that the $\log^2 s_i$ term in Equation 3.7 represents a *lognormal* prior on the JSVs. However this prior is based on the distribution of singular values only in f . Therefore I also include the term $v(1 + |\mathbf{J}_n|)$ to ensure that the penalty is truly symmetric in g by accounting for the total volume in both images being penalised.

In practice, computation of the logarithm in Equation 3.6 (and its associated effect on the gradient and Hessian) may add significant computational overhead (Ashburner et al. 2000). I therefore apply the following approximation:

$$\log^2 x \approx x + \frac{1}{x} - 2 \quad (3.8)$$

Recognising that:

$$\log^2 x = \frac{\log^2(x^2)}{4} \quad (3.9)$$

and that:

$$\text{tr}(\mathbf{A}^T \mathbf{A}) = \sum_{i=1}^3 \lambda_i(\mathbf{A}^T \mathbf{A}) = \sum_{i=1}^3 (s_i(\mathbf{A}))^2 \quad (3.10)$$

where $\lambda_i(\mathbf{A}^T \mathbf{A})$ denotes the i^{th} eigenvalue of $\mathbf{A}^T \mathbf{A}$ and where $s_i(\mathbf{A})$ denotes the i^{th} singular value of $\mathbf{A}$, one can then write:

$$C(\mathbf{w}) \approx \sum_{n=1}^N v(1 + |\mathbf{J}_n|) \sum_{i=1}^3 \left(s_i^2 + \frac{1}{s_i^2} - 2 \right) / 4 \quad (3.11)$$

$$= \sum_{n=1}^N v(1 + |\mathbf{J}_n|) \text{tr} \left(\mathbf{J}_n^T \mathbf{J}_n + (\mathbf{J}_n^{-1})^T \mathbf{J}_n^{-1} - 2\mathbf{I} \right) / 4 \quad (3.12)$$

An important benefit of this approximation is that the cost function is now an explicit function of only the elements of $\mathbf{J}$, and therefore the full SVD of $\mathbf{J}$ need never be calculated. However, the rationale behind Equation 3.8 is not immediately obvious, and is therefore worthy of further explanation.

Let:

$$y(x) = x + \frac{1}{x} - 2 \quad (3.13)$$

And:

$$z(x) = \log x \quad (3.14)$$

$$\therefore x(z) = e^z \quad (3.15)$$

Then:

$$y(z) = e^z + e^{-z} - 2 \quad (3.16)$$

Now, taking the Taylor expansion of $y(z)$ about $z = 0$:

$$\begin{aligned} y(z) |_{z=0} &\approx y(0) + \frac{y'(0)}{1!} (z - 0) + \frac{y''(0)}{2!} (z - 0)^2 \\ &= 0 + 0 + z^2 \\ &= z^2 \end{aligned} \quad (3.17)$$

And finally, using Equation 3.14:

$$\therefore y(x) \approx \log^2 x \quad (3.18)$$

Figure 3.1 graphically demonstrates the relationship between the exact and approximate penalty functions. It is clear that the approximation preserves the same desirable properties as the exact penalty.

This approximation highlights an interesting interpretation of the SPRED penalty as a form of local rigidity-based regularisation. As shown in Equation 3.12, utilising the approximation leads to a penalty in terms of the local Jacobian containing the term:

$$y = \text{tr} \left(\mathbf{J}^T \mathbf{J} + \left(\mathbf{J}^{-1} \right)^T \mathbf{J}^{-1} - 2\mathbf{I} \right) \quad (3.19)$$

Which may be rearranged as follows:

$$y = \text{tr} \left(\mathbf{J} \mathbf{J}^T - 2\mathbf{I} + \left(\mathbf{J}^{-1} \right)^T \mathbf{J}^{-1} \right) \quad (3.20)$$

$$= \text{tr} \left(\mathbf{J} \mathbf{J}^T - \mathbf{J} \mathbf{J}^{-1} - \left(\mathbf{J}^{-1} \right)^T \mathbf{J}^T + \left(\mathbf{J}^{-1} \right)^T \mathbf{J}^{-1} \right) \quad (3.21)$$

$$= \text{tr} \left(\left(\mathbf{J} - \left(\mathbf{J}^{-1} \right)^T \right) \left(\mathbf{J}^T - \mathbf{J}^{-1} \right) \right) \quad (3.22)$$

$$= \text{tr} \left(\left(\mathbf{J}^T - \mathbf{J}^{-1} \right)^T \left(\mathbf{J}^T - \mathbf{J}^{-1} \right) \right) \quad (3.23)$$

$$= \left\| \mathbf{J}^T - \mathbf{J}^{-1} \right\|_F^2 \quad (3.24)$$

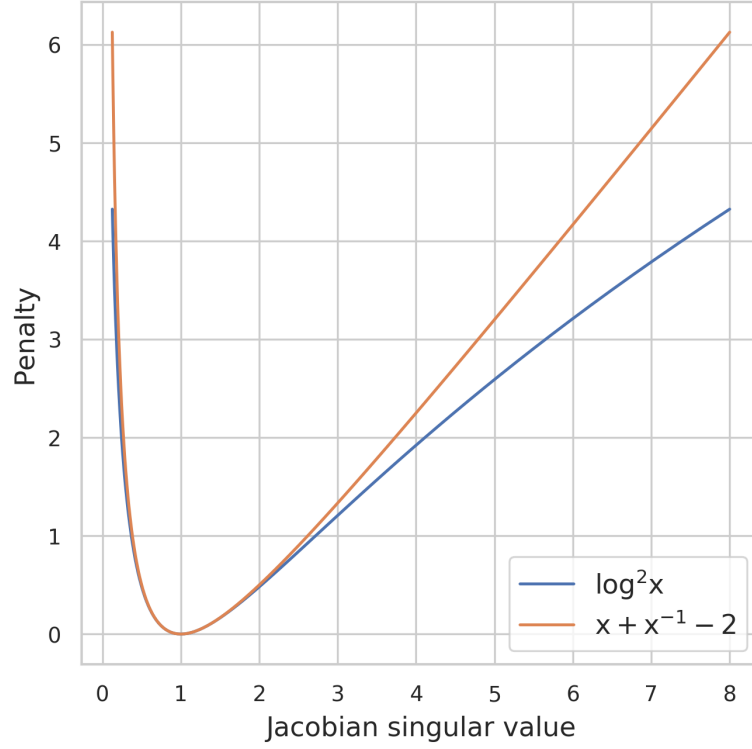


Figure 3.1: Exact vs approximate SPRED penalty - Effect of the approximation in Equation 3.8 on the SPRED penalty. The approximation is almost indistinguishable from the exact penalty for singular values near 1. For singular values between 0.25 and 4 the approximation is accurate to within 17 %, and within 42 % between 0.125 and 8. Note that the approximation maintains the desirable symmetry for expansions and contractions. Note too that the approximation majorises the exact penalty (*i.e.*, the approximation is always greater than or equal to the exact penalty), and therefore maintains the exact penalty’s ability to ensure diffeomorphism.

where $\|\cdot\|_F^2$ is the squared Frobenius norm.

This penalty is minimised to 0 when $\mathbf{J}^T = \mathbf{J}^{-1} \iff \mathbf{J}$ is an orthogonal matrix. This provides an intuitive alternative interpretation of the approximate SPRED penalty as one that penalises transformations for which the local deformation deviates from a pure rotation. Under that interpretation, the similarity to the *Local Rigidity* penalty

$$\|\mathbf{J}^T \mathbf{J} - \mathbf{I}\|_F \quad (3.25)$$

as used by (Loeckx et al. 2004; Staring et al. 2007) becomes apparent.

The primary difference between these two methods is that Equation 3.24 includes $\mathbf{J}^{-1}$, and consequently tends to ∞ as $|\mathbf{J}|$ tends to 0. The SPRED approximation is

therefore able to preserve diffeomorphism, whereas Local Rigidity is not.

3.2.1.3 Gradient and Hessian

The penalty function as defined in Section 3.2.1.2 is included into the Gauss-Newton style optimisation framework of MMORF, and therefore will require calculation of the gradient and the Gauss-Newton approximation to the Hessian of Equation 3.11. Note that for a valid application of Gauss-Newton (i.e., for using the Gauss-Newton approximation of the Hessian to be allowable) the function being minimised must be of the form $y = \frac{1}{2}a(x)^2$, which Equation 3.12 is not (Chen 2011). However, as the penalty is always ≥ 0 , by making the substitution $a(x) = \sqrt{2c(x)}$:

$$C(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^N a_n^2(\mathbf{w}) \quad (3.26)$$

$$= \sum_{n=1}^N c_n(\mathbf{w}) \quad (3.27)$$

it becomes of that form. This means that:

$$\nabla C(\mathbf{w}) = \sum_{n=1}^N \frac{\partial a_n}{\partial \mathbf{w}} a_n(\mathbf{w}) \quad (3.28)$$

$$= \sum_{n=1}^N \frac{\partial c_n}{\partial \mathbf{w}} \quad (3.29)$$

and therefore that the Gauss-Newton Hessian approximation becomes:

$$\mathbf{H}(\mathbf{w}) = \sum_{n=1}^N \left(\frac{\partial a_n}{\partial \mathbf{w}} \right)^T \frac{\partial a_n}{\partial \mathbf{w}} \quad (3.30)$$

$$= \sum_{n=1}^N \left(\frac{1}{\sqrt{2c_n(\mathbf{w})}} \frac{\partial c_n}{\partial \mathbf{w}} \right)^T \left(\frac{1}{\sqrt{2c_n(\mathbf{w})}} \frac{\partial c_n}{\partial \mathbf{w}} \right) \quad (3.31)$$

$$= \sum_{n=1}^N \frac{1}{2c_n(\mathbf{w})} \left(\frac{\partial c_n}{\partial \mathbf{w}} \right)^T \frac{\partial c_n}{\partial \mathbf{w}} \quad (3.32)$$

i.e., it introduces a factor $1/2c_n(\mathbf{w})$ compared to how one would “normally” calculate the Gauss-Newton Hessian.

The gradient $\nabla C(\mathbf{w})$ is a $3M \times 1$ column vector (where M is the number of B-splines used to represent one directional component of the warp-field), whose m^{th} element is of the form:

$$\nabla C_m = \sum_{n \in V_m} \underbrace{\begin{bmatrix} \frac{\partial c_n}{\partial \mathbf{s}_n} \end{bmatrix}}_{1 \times 3} \underbrace{\begin{bmatrix} \frac{\partial \mathbf{s}_n}{\partial \mathbf{J}_n} \end{bmatrix}}_{3 \times 9} \underbrace{\begin{bmatrix} \frac{\partial \mathbf{J}_n}{\partial w_m} \end{bmatrix}}_{9 \times 1} \quad (3.33)$$

Where:

$$V_m = \{\text{voxels in } f \text{ where B-spline } m \text{ has support}\},$$

and where $n \in V_m$ means that the summation is over all voxels n for which the B-spline m has support. $\mathbf{J}_n$ is a vectorised version of the Jacobian at the n th voxel and $\mathbf{s}_n$ is a vector of the three singular values of $\mathbf{J}_n$.

Similarly the Gauss-Newton Hessian is a $3M \times 3M$ matrix whose jk^{th} element is:

$$\mathbf{H}_{jk} = \sum_{n \in \{V_j \cap V_k\}} \frac{1}{2c_n(\mathbf{w})} \underbrace{\begin{pmatrix} \begin{bmatrix} \frac{\partial c_n}{\partial \mathbf{s}_n} \end{bmatrix} \begin{bmatrix} \frac{\partial \mathbf{s}_n}{\partial \mathbf{J}_n} \end{bmatrix} \begin{bmatrix} \frac{\partial \mathbf{J}_n}{\partial w_j} \end{bmatrix} \end{pmatrix}}_{\text{scalar}} \underbrace{\begin{pmatrix} \begin{bmatrix} \frac{\partial c_n}{\partial \mathbf{s}_n} \end{bmatrix} \begin{bmatrix} \frac{\partial \mathbf{s}_n}{\partial \mathbf{J}_n} \end{bmatrix} \begin{bmatrix} \frac{\partial \mathbf{J}_n}{\partial w_k} \end{bmatrix} \end{pmatrix}}_{\text{scalar}} \quad (3.34)$$

Where:

$$V_j = \{\text{voxels in } f \text{ where B-spline } j \text{ has support}\}$$

$$V_k = \{\text{voxels in } f \text{ where B-spline } k \text{ has support}\}$$

and where $n \in \{V_j \cap V_k\}$ denotes summation over all voxels n for which both B-spline j and k have support. It should be noted that although $3M \times 3M$ can be a very large number, especially for high warp-resolutions/small knot-spacings, the vast majority of elements are zero because most spline combinations jk have no overlap in support.

I will demonstrate the practicalities of actually calculating these entities in the sections which follow.

3.2.2 Intuition

In this section I detail how to calculate the suggested penalty, and why it is an ideal, albeit challenging, candidate for Single Instruction Multiple Data (SIMD) style parallelisation on a GPU (NVIDIA 2019). For the estimation of the warp parameters I use the Gauss-Newton method, and so need to calculate the gradient of the penalty with respect to the spline coefficients of the warps and a Gauss-Newton approximation to the Hessian.

I attempt now to give an intuitive outline of the technical/mathematical formalism which follows in Section 3.2.1. At each voxel one can calculate a Jacobian matrix using the values of the surrounding warp spline-coefficients. The singular values of this matrix represent the orthogonal scalings of that voxel due to the warp. The SPRED penalty (described in Section 3.2.1.2) is a function of these singular values, and it is therefore possible to calculate each voxel's contribution to the total penalty independently. Those calculations are a perfect match for SIMD parallelisation with one computational thread per *voxel*, where each thread performs identical calculations but on different data. The individual voxel contributions are subsequently summed (a *reduction* in GPU parlance) to obtain the total penalty for the warp.

In order to calculate the gradient of the penalty, one needs to apply the chain rule, which says that:

$$\frac{d}{dx} [f(g(x))] = \frac{df}{dg} \frac{dg}{dx} \quad (3.35)$$

Remembering that for example g might be a vector-valued function of a vector-valued argument x which would make df/dg a vector and dg/dx a matrix.

In this case the contribution to an element of the gradient from a given voxel is a scalar valued function of the vector of singular values of the local Jacobian matrix. These in turn are each a function of the vector of elements of the Jacobian matrix. Finally, these elements are functions of the vector of spline-coefficients whose support includes that voxel. The first two factors are again calculated using one thread per

voxel, while the third factor uses one thread per *spline-coefficient*. Similarly each element of the total gradient is calculated by one thread per *spline-coefficient*.

To understand how the Hessian matrix is calculated consider a matrix $\mathbf{A}$ where each row corresponds to a voxel and each column to a warp spline-coefficient, and where $\mathbf{A}_{ij}$ is the rate of change of the penalty contribution from voxel i with respect to coefficient j . The Gauss-Newton approximation to the Hessian then becomes $\mathbf{A}^T \mathbf{A}$. However, for practical reasons, $\mathbf{A}$ is never explicitly calculated or stored and instead the Hessian $\mathbf{H}$ is calculated directly. In this instance I use one thread per non-zero *element of $\mathbf{H}$* for the final calculation, but there are a number of preceding steps which employ the same one thread per *voxel* approach as in the gradient calculations.

3.2.3 Implementation

I now present how I achieved computational tractability of the SPRED penalty, its gradient, and GN Hessian.

3.2.3.1 Parallelising Calculation of Gradient and Hessian

As with all but the simplest optimisation strategies, calculating the value of the SPRED penalty itself is not the computationally costly part of this algorithm. Rather, it is the corresponding gradient and Hessian calculations which are massively computationally complex. See, for example, Equation 3.33, where there are multiple applications of the chain rule to vector valued functions of vectors, leading to intermediate matrices. These intermediates then need to be multiplied together, further increasing the complexity. It is easy to see that this complexity is compounded even further in Equation 3.34 when calculating the Hessian.

If these calculations were all performed sequentially the problem would become computationally intractable, and the suggested form of regularisation would be a mathematical nicety with no practical impact.

However I will demonstrate how the problem can be framed in terms of the application of multiple massively parallelised SIMD operations, rendering it realistic to use.

As an example, consider how one might parallelise the gradient calculation in Equation 3.33. Firstly, note that the part:

$$\underbrace{\underbrace{\begin{bmatrix} \frac{\partial c_n}{\partial \mathbf{s}_n} \end{bmatrix}}_{1 \times 3} \underbrace{\begin{bmatrix} \frac{\partial \mathbf{s}_n}{\partial \mathbf{J}_n} \end{bmatrix}}_{3 \times 9}}_{1 \times 9} \quad (3.36)$$

depends only on which voxel n is currently being considered. Each of these 1×9 vectors can therefore be calculated and stored independently by one GPU thread. I represent these pre-calculated vectors as an $N \times 9$ matrix (where N is the total number of voxels in image $\mathbf{f}$) which denote $[\partial c / \partial \mathbf{J}]$ where the n th row is:

$$\left[\frac{\partial c}{\partial \mathbf{J}} \right]_{n*} = \underbrace{\begin{bmatrix} \frac{\partial c_n}{\partial \mathbf{s}_n} \end{bmatrix} \begin{bmatrix} \frac{\partial \mathbf{s}_n}{\partial \mathbf{J}_n} \end{bmatrix}}_{1 \times 9} \quad (3.37)$$

The columns of $[\partial c / \partial \mathbf{J}]$ can be thought of as *partial derivative images* and can be visualised as demonstrated in Section 3.2.4. Note also that in these calculations one can use the approximation given in equation 3.12 to avoid explicit computation of the JSVs.

The next part:

$$\underbrace{\begin{bmatrix} \frac{\partial \mathbf{J}_n}{\partial w_m} \end{bmatrix}}_{9 \times 1} \quad (3.38)$$

is “constant” in the sense that its elements are given by the spatial derivatives of the B-spline basis-functions and can be calculated (analytically) once and for all for a given knot-spacing. If I denote a column-vectorised version of a single 3D B-spline basis-function (whose dimensions are given by the knot-spacing) by $\mathbf{B}$, I can create a matrix $[\partial \mathbf{J} / \partial w]$ containing all the elements needed as:

$$\left[\frac{\partial \mathbf{J}}{\partial w} \right] = \underbrace{\begin{bmatrix} \mathbf{B}^{(x)} & \mathbf{B}^{(y)} & \mathbf{B}^{(z)} \end{bmatrix}}_{N_S \times 3} \quad (3.39)$$

where $\mathbf{B}^{(*)}$ denotes the spline basis-function spatially differentiated in the i^{th} direction and N_S denotes the total size of the 3D spline (determined by the knot-spacing).

Finally each element ∇C_m of the gradient can be calculated by independent threads as:

$$\nabla C_m = \sum_{n \in V_m} \sum_{k \in L_m} \left[\frac{\partial c}{\partial \mathbf{J}} \right]_{nk} \left[\frac{\partial \mathbf{J}}{\partial w_m} \right]_{n'k'} \quad (3.40)$$

Where:

$$V_m = \{\text{voxels in } f \text{ where B-spline } m \text{ has support}\}$$

$$L_m = \{\text{elements of } \mathbf{J}_n \text{ where B-spline } m \text{ has an effect}\}$$

$k \in L_m$ denotes that k will select the three columns of $[\partial c / \partial \mathbf{J}]$ that contribute to the m^{th} element of the gradient. To be concrete, if $1 \leq m \leq M$ the element pertains to the x -component of the warp-field and k will select the columns of $[\partial c / \partial \mathbf{J}]$ corresponding to the first row of $\mathbf{J}$ (see equation 3.2), and if $M < m \leq 2M$ it will select the columns of $[\partial c / \partial \mathbf{J}]$ corresponding to the second row etc. The ' superscripts on n' and k' indicate transformed versions of n and k such that n' takes into account the offset of the m^{th} spline into the image, and k' will select the corresponding column (regardless of row) in $\mathbf{J}$.

Equation 3.40 serves as an illustration of some of the challenges of SIMD parallelisation. I have chosen a strategy where each element of ∇C is calculated by one thread. That choice means that I avoid having to synchronise writes to ∇C or having to perform reductions, both of which reduce occupancy and speed. The downside of that choice is that the different threads within a warp will access global memory in a non-coalesced fashion. This is because each thread will access a 3D sub-volume in each of the nine "partial derivative images" defined in equation 3.37, where these sub-volumes are offset by an integer multiple of the knot-spacing in one or more dimensions.

So, in summary the calculation of the gradient can be divided into three parts:

- One which is separable over voxels (or more generally "sample points") where the calculations for one voxel can be performed and stored independently by one GPU thread.

- One which is “constant” and can be performed once and for all and stored where the results can be accessed by all threads.
- One which is separable over warp parameters (spline coefficients) where calculations for one warp parameter can be performed and stored independently by one GPU thread.

Following this line of reasoning, the Hessian calculation of Equation 3.34 can be similarly divided into portions which are constant, separable over samples, and separable over warp parameters.

As this process is central to how using the SPRED penalty is made tractable, I provide an intuitive understanding of what implementing the parallelisation of equation 3.33 actually looks like in practice by considering a simple 2D example in the following section.

3.2.4 2D Example

I begin by defining a displacement warp field and its values in the underlying image space. The image space is a square with dimensions 21×21 . The cubic B-splines are placed with their control points (knots) 5 samples apart. Each spline has a spatial support radius of 2 knots (10 samples), and therefore I place an extra ring of splines around the perimeter of the image space, such that each point in the image has an equal number of splines with support there. As a result, the x -warp and y -warp parameters are of dimension 7×7 . I initialise the warp with a random set of parameters. Figure 3.2 shows the warp parameters, the resulting warp in image space, and the spatial partial derivatives of the warp in image space.

Based on the spatial partial derivatives of the displacement fields in Figure 3.2, the local Jacobian matrix can be calculated at each point in space according to Equation 3.4.

The elements of $\mathbf{J}$ for each point in the sample space, as well as the resulting Jacobian determinant, $|\mathbf{J}|$, are shown in Figure 3.3, which now contains all of

the information necessary to calculate the voxelwise parallelised part (Equation 3.37) of the gradient (Equation 3.33).

Based on the elements of $\mathbf{J}$, Figure 3.4(a) shows the resulting contribution to the total cost from each point in the image. 3.4(b) to 3.4(e) then constitute the gradient of the cost with respect to the elements of $\mathbf{J}$ at each point in the image, i.e., the columns of $[\partial c / \partial \mathbf{J}]$ (Equation 3.37). Each point in the four partial derivative images can be calculated completely independently of every other point. The calculation of these images therefore constitutes the first parallelised step of the algorithm

I now move onto visualising the result of Equation 3.40. Note that Equation 3.39 is constant across the image and identical over the support of every spline. Therefore, when calculating the gradient in a parallel-per-parameter regime, the operation conforms exactly to the SIMD paradigm and may be very efficiently implemented on GPUs. In practice, I perform one such calculation for each combination of $\mathbf{J}$ element gradient image (Figure 3.4(b) to 3.4(e)) and warp direction (x or y), the results of which are shown in Figure 3.5(b) to 3.5(e). Finally, I sum over these combinations for each warp direction, and the result is then the gradient of the penalty with respect to the warp parameters. This is shown in 3.5(f) and 3.5(g).

3.2.5 Testing

Meaningfully assessing something like the anatomical plausibility of a deformation is a challenge, and I will here present how I have approached it.

To facilitate a quantitative comparison of registration aggressiveness one requires measures of changes in both volume and shape. I use two main indices for assessing what I consider to be anatomical plausibility.

First, the Jacobian determinant is useful in that it provides a single number quantifying volumetric changes. I have analysed both the histogram of Jacobian determinants within the brain - which provides information about how aggressively the warp is contracting or expanding the volume on average - and how those Jacobian determinants are spatially distributed - i.e., whether the deformation appears to be spatially sensible.

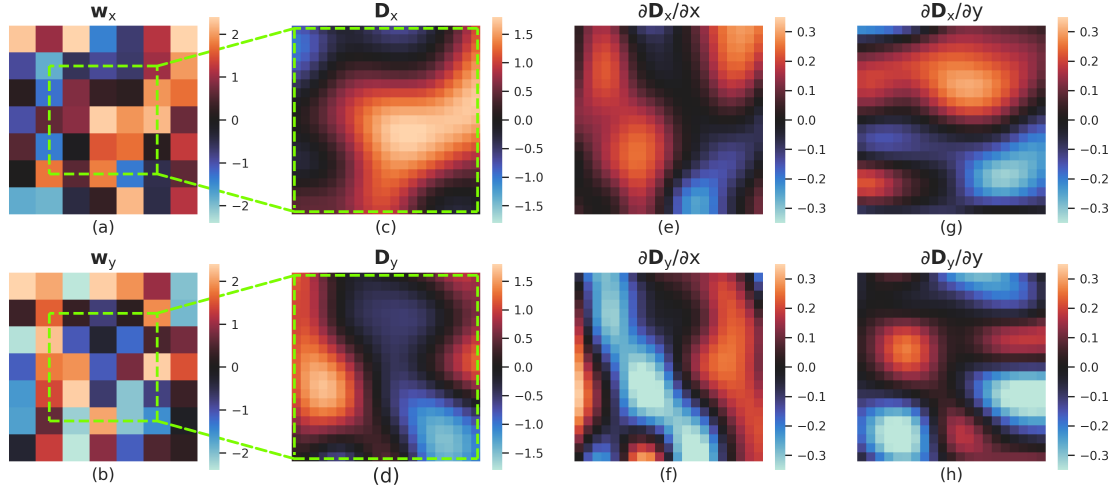


Figure 3.2: 2D B-spline displacement field and spatial derivatives - Example of a warp (or technically *displacement*) field for a random set of parameters (cubic B-spline coefficients). The positive x -direction is defined from left to right, and the positive y -direction from top to bottom. The x -warp parameters $\mathbf{w}_x$ (a) and y -warp parameters $\mathbf{w}_y$ (b) are arranged on a regular 7×7 grid. The resulting x -displacement field $\mathbf{D}_x$ (c) and y -displacement field $\mathbf{D}_y$ (d) corresponding to $\mathbf{w}_x$ and $\mathbf{w}_y$ are evaluated on a regular grid with knot spacing of 5×5 . Note that only those parts of the warps with full support of the surrounding B-splines are shown (i.e., the area outside of the green square is considered to be *outside* of the image being warped). This leaves a square of 21×21 samples in image space. The spatial partial derivatives of the x -warp in the x -direction (e), y -warp in the x -direction (f), x -warp in the y -direction (g) and y -warp in the y -direction (h) are presented, as these values are used in determining the local Jacobian matrices of the warp field. Cubic B-spline parametrisation ensures that both the warps and their spatial derivatives have analytical solutions and are smooth-continuous.

Secondly, I use the Cube-Volume Aspect Ratio (CVAR) as a single number to describe the deviation in shape of any cuboid from a regular cube (W. Smith and Wormald 1998). CVAR in 3 dimensions is defined as: “*The cube-root of the ratio of the volume of the smallest regular cube which can fully enclose the cuboid, to the cuboid’s own volume*”. Alternatively, for a deformation one can equally well define it as: “*The cube-root of the ratio of the largest Jacobian singular value cubed, to the Jacobian determinant*”.

$$CVAR = \sqrt[3]{\frac{s_{i-max}^3}{s_1 \times s_2 \times s_3}} \quad (3.41)$$

CVAR therefore takes on a value of 1 for a regular cube, and is greater than 1 for any other shape. Similarly to the Jacobian determinant, I have analysed both

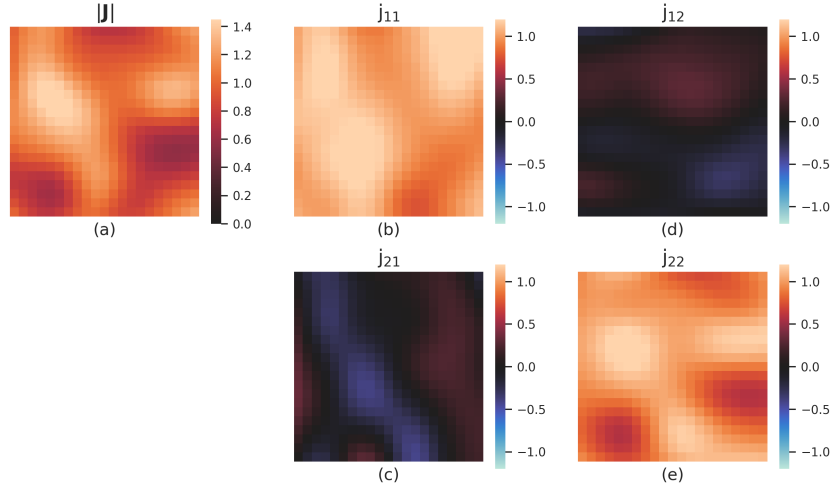


Figure 3.3: 2D Jacobian matrix fields - Elements of the local Jacobian matrix of the warp. At each position $[x, y]$ in image space a local Jacobian matrix $\mathbf{J}$ is calculated according to Equation 3.4 by adding the identity matrix $\mathbf{I}_2$ to the 2×2 matrix composed of the elements of Figure 3.2 (e) to (h) at that position. The resulting 2×2 2D fields (each showing how one element of $\mathbf{J}$ varies across the image) are presented here in sub-figures (b) to (e). From these a local Jacobian determinant field (a) can be calculated. The SPRED penalty is based on the singular values of $\mathbf{J}$, and is therefore a function of fields (b) to (e).

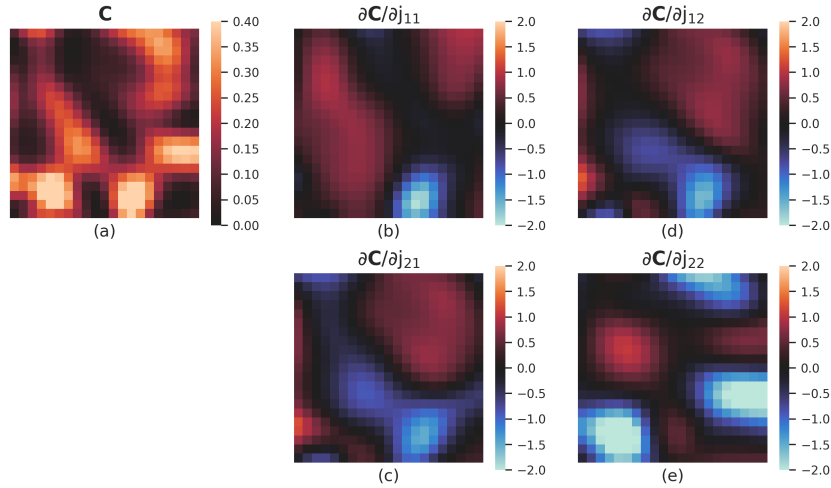


Figure 3.4: 2D cost field and Jacobian matrix gradient - From the singular values of the local Jacobian matrices of Figure 3.3 a cost field $\mathbf{C}$ (a) can be evaluated at each position $[x, y]$ of the image. $\mathbf{C}$ then represents the contribution to the total cost from each point in image space. The partial derivatives of $\mathbf{C}$ with respect to the elements of $\mathbf{J}$ are shown in sub-figures (b) to (e). These partial derivative fields (which together make up the overall $\partial\mathbf{C}/\partial\mathbf{J}$ term), can be calculated independently for each position $[x, y]$ as they depend *only* on the values of $\mathbf{J}$ at $[x, y]$, making this step ideal for a parallel-per-voxel approach.

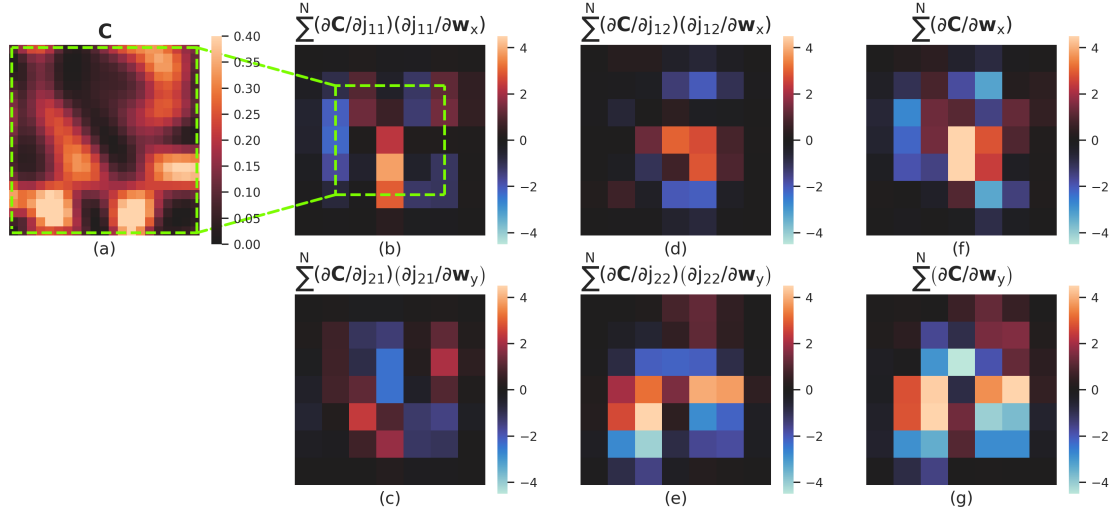


Figure 3.5: 2D cost field and warp parameter gradient - The final, and most computationally expensive, step is calculating the gradient of the total SPRED penalty (being the sum of $\mathbf{C}$ over all x and y) with respect to the warp parameters. Given the partial derivatives of $\mathbf{C}$ with respect to $\mathbf{J}$, knowing the gradient of the resulting partial derivative fields (i.e., Figure 3.4 (b) to (e)) with respect to the warp parameters would allow application of the chain rule to calculate the total gradient $\partial\mathbf{C}/\partial\mathbf{w}$. I proceed by recognising that the piece-wise polynomial nature of B-splines means that the partial derivative of $\mathbf{J}$ with respect to each warp parameter is constant. Additionally, at position $[x, y]$ it is exactly 0 for any B-spline which does not have support at that position. Thus, I can calculate the total gradient with respect to each warp parameter by summing the result of the chain rule across each position $[x, y]$ where that spline has support. (b) to (e) illustrates this in terms of the contributions to the overall gradient due to the effect of the warp parameters $\mathbf{w}_x$ and $\mathbf{w}_y$ on each element of the $\mathbf{J}$ fields. These partial derivatives can be calculated independently for each warp parameter, making this step ideal for a parallel-per-parameter approach. The end result of the process is then the gradient of the total penalty with respect to $\mathbf{w}_x$ (f) and $\mathbf{w}_y$ (g).

the histograms and the spatial distributions of the CVAR.

However, such metrics are only meaningful when considered jointly with registration accuracy. In other words, the correct way to assess biological plausibility is to condition it on registration accuracy. This raises a second point, which is that a sensible measure of registration accuracy is therefore required. It is known that image similarity measures are a poor choice for evaluating registration accuracy, as they are only a proxy for what one is truly interested in, i.e., alignment of common anatomical structures (Rohlfing 2012). Image-wide tissue maps are not a good option either, as they are global in nature and only really test tissue classification rather than anatomical alignment (Rohlfing 2012). I therefore chose

the overlap of manually segmented cortical regions as the data-consistency measure - which has been established by the image registration community as a viable solution to the above concerns (A. Klein et al. 2009). However this method is not perfect, as it does not account for errors in manual segmentation. This means that absolute measures of overlap are not meaningful on their own - but as a comparison *between* methods it is entirely valid.

For any registration method, regardless of the specifics of the regulariser, there is a trade-off between the image similarity term and regularisation term. That trade-off is determined by their relative weights, something that is typically calibrated empirically (however see Simpson et al. (2012) for an attempt to determine it probabilistically). Therefore, in order to compare two methods on the basis of plausible warps it is crucial to calibrate both methods so that they achieve (close to) identical overlap scores.

3.2.5.1 Registration Tools

In order to gain as clear a picture as possible of how MMORF performs with the SPRED penalty, I chose to compare to both different regularisation penalties which are all implemented *within* MMORF, as well as comparing to several different registration tools entirely.

The four registration tools chosen were:

FLIRT (Jenkinson and S. M. Smith 2001) I chose to include a linear registration method in order to provide a yardstick by which to compare the overlap scores, and in particular the differences in overlap, of the other methods.

ANTs (Avants et al. 2008; Avants et al. 2014) ANTs has been shown to perform well in a direct comparison with other methods (A. Klein et al. 2009; Ou et al. 2014), and represents the current state-of-the-art for registration accuracy. A method performing significantly worse would not be of any practical interest, so it provides a minimum accuracy bar for the other methods in the comparison. ANTs was run with both cross correlation (ANTs-CC, Avants et al. (2011))

and mean sum of squares image similarity metrics (ANTs-MSQ). In both cases the greedy SyN transformation model was used.

FNIRT (Andersson et al. 2007; Andersson et al. 2019) I included FNIRT because it uses the same image similarity and deformation model (B-Splines) as MMORF, but employs a different strategy to ensure invertibility (Karacali and Davatzikos (2004)) and a different regulariser (bending energy).

MMORF Implements the SPRED regulariser. In order to demonstrate just how different the SPRED behaviour is to a Jacobian Determinant regulariser (for example Rohlfing et al. (2003) or Leow et al. (2007)) I have also implemented a regulariser based on the sum of squares of log-determinants of the Jacobians. A third regulariser, based on preserving local rigidity (Loeckx et al. 2004) is also tested, due to the mathematical similarities between this penalty and SPRED.

While there are myriad other tools that may also have been included in this comparison (e.g., SPM Unified Segmentation (Ashburner and Friston 2005), DARTEL (Ashburner 2007), elastix (S. Klein et al. 2010), MIRTk (Rueckert et al. 1999; Schnabel et al. 2001), Diffeomorphic Demons (Vercauteren et al. 2009), or ART (Ardekani et al. 2005)), the chosen methods represent a broad enough overview of the types of tools available to adequately characterise the performance of MMORF with the SPRED penalty.

The Jacobian determinant regularisation tested within MMORF implements the penalty:

$$C_{LJ}(\mathbf{w}) = \sum_{n=1}^N v(1 + |\mathbf{J}_n|) \log^2 |\mathbf{J}_n| \quad (3.42)$$

Given that I believe the distribution of Jacobian determinants is a good indicator of anatomical plausibility, it is natural to ask why I do not just penalise that directly then. In the introduction (Section 3.1) I made the point that a cube being transformed into a sheet or stick in order to preserve its volume is undesirable from an anatomical plausibility perspective. The results of using this regularisation will

demonstrate this occurring in practice. By penalising the squared logarithm of the Jacobian, Equation 3.42 is equally as capable of preserving diffeomorphism as compared to the SPRED penalty, but unable to control changes in shape.

The local rigidity regularisation tested within MMORF implements the penalty:

$$C_{LR}(\mathbf{w}) = \sum_{n=1}^N v \left\| \mathbf{J}_n^T \mathbf{J}_n - \mathbf{I} \right\|_F \quad (3.43)$$

As explained in Section 3.2.1.2, this penalty is a simpler form of penalising warps which deviate from rigid body translations and rotations, but without the benefit of preserving diffeomorphism.

Together, these regularisation strategies present simpler solutions to address some of the same concerns as the SPRED penalty, and allow for an interesting evaluation of the benefits of the extra complexity introduced by SPRED.

3.2.5.2 Test Dataset

I have chosen to use the Non-rigid Image Registration Evaluation Project (NIREP) (Christensen et al. 2006) as the dataset for comparing the registration methods. This dataset consists of the T1-weighted scans of 16 healthy subjects, eight male (average age 32.5 years) and eight female (average age 29.8 years). Each subject's scan has been brain extracted, bias corrected, and 32 cortical regions (16 left hemisphere, 16 right) have been expertly hand-segmented. The segmented regions provide a *ground truth* for anatomical correspondence of a finer granularity than simple tissue maps.

3.2.5.3 Evaluation Strategy

The comparison between tools was evaluated using pairwise registrations of each combination of the 16 subjects (240 warps in total). The within MMORF comparison of regularisation strategies was performed using only registration to subject 1 (15 warps in total).

Each of the 32 segmented cortical regions were transformed through these warps and the Jaccard Coefficient overlap score was calculated. The distributions of these

overlaps in each region were then compared between registration methods. The Jaccard coefficient for two regions A and B is defined as:

$$\frac{A \cap B}{A \cup B}, \quad (3.44)$$

and ranges from 0 for no overlap to 1 for perfect overlap.

Additionally, the distribution of Jacobian determinants and of cube-volume aspect ratios were calculated in order to gain some insight into how aggressive each method is in terms of volume and shape changes. These were inspected and compared as histograms, spatial maps and by statistics such as min, max, mean, 5th and 95th percentiles.

3.2.5.4 Registration Parameter Selection

An important aspect when comparing registration algorithms is ensuring that the user-selectable parameters are as optimal as possible (A. Klein et al. 2009).

I aimed to ensure a high level of registration accuracy by using an optimised configuration of ANTs (Avants et al. (2008)) as a benchmark. To make sure warp comparisons were meaningful, the other methods were calibrated to try and match the performance of ANTs. In this way, I could compare anatomical plausibility conditioned on state-of-the-art registration accuracy. I will already note here that not all methods were capable of matching the ANTs benchmark, despite my concerted efforts. I did not, however, feel I should “de-tune” ANTs, as MMORF *was* able to match it for registration accuracy - and doing so may have raised the common question of bias when comparing one’s own method against the state-of-the-art.

For both ANTs-CC and ANTs-MSQ the steps in the resolution pyramid were set to 10, 6, 4, 2 and 1 mm isotropic. Smoothing was performed with Gaussian kernels of sigma 5, 3, 2, 1 and 0 mm isotropic as per the tool’s author’s recommendations. This left three parameters over which to optimise, namely:

- Gradient step (λ_g)
- Update field variance (λ_u)

- Total field variance (λ_t)

λ_g determines how far each point is allowed to move on each iteration, and increasing this value allows for higher frequency deformations. λ_u determines how much to smooth the gradient field between updates and increasing this value increases smoothness in the velocity field. λ_t determines how much to smooth the total displacement field.

A grid-search of these parameters was performed using a subset of the NIREP dataset, and the parameters which led to the best overlap scores were selected. Note that despite my best efforts, I was unable to get ANTs-MSQ to match the registration accuracy of ANTs-CC. The best results for ANTs-MSQ were achieved with:

$$\lambda_g = 0.85$$

$$\lambda_u = 2$$

$$\lambda_t = 0$$

And for ANTs-CC with:

$$\lambda_g = 0.5$$

$$\lambda_u = 2$$

$$\lambda_t = 0$$

λ_t was found to always decrease performance and was therefore left at the default value of 0. The final values for λ_g and λ_u are more aggressive than the default (fairly conservative) values of $\lambda_g = 0.1$ and $\lambda_u = 3$ normally used for both methods, but I found that this was necessary to achieve the optimal registration accuracy and therefore a meaningful comparison. Note also that the ANTs-CC parameters are very similar to those supplied by the tool's author for use in similar analyses such as A. Klein et al. (2009).

FNIRT was run with an optimised strategy proposed by the tool’s author. This consisted of a multi-resolution, multi-smoothing level registration, to a final knot spacing of 1 mm isotropic, and is explained in detail in Andersson et al. (2019).

For MMORF, I fixed the number of steps in the warp resolution pyramid (40, 20, 10, 5, 2.5 and 1.25 mm isotropic), and then optimised over three parameters, namely:

- Smoothing as a function of warp resolution (λ_s)
- Initial regularisation (λ_{ri})
- Rate of decrease of regularisation (λ_{rd})

The FWHM of the smoothing kernel was defined as the current warp resolution divided by λ_s . The initial regularisation λ_{ri} is in arbitrary units and its exact value is not of interest. λ_{ri} was determined independently for each regularisation penalty. The total regularisation at each step of the resolution pyramid was $\lambda_{ri} \times (\lambda_{rd})^{it}$ where it is the current position in the resolution pyramid (i.e. 0 to 5).

The parameter values were chosen via a grid search of the parameter space. Half of the NIREP dataset (8 subjects, 56 warps) were used, and overlap scores calculated as per Section 3.2.5.3. The parameters which resulted in the best overlap scores were then used for the rest of the analysis. The final parameters common to all regularisation methods were:

$$\lambda_s = 4.0$$

$$\lambda_{rd} = 0.75$$

These parameters are relatively aggressive, but as the aim was to match the overlap scores of ANTs-CC this is to be expected.

Note that while significant effort has been made to ensure that each tool performs as well as possible, the overarching goal was not to definitively classify which tool performed best in terms of overlap scores. Rather it was to investigate how the SPRED penalty affects the nature of the deformations when compared to similar performing tools. Hence, the main goal was to calibrate their respective performances in terms of overlap accuracy so that they were not excessively different.

3.3 Results

I present the results under two general groupings. The first is a comparison between MMORF with SPRED and other registration tools - in particular ANTs-CC as the current state-of-the-art. This will establish the combination of MMORF with SPRED as a new benchmark of anatomical plausibility. The second is a comparison of different regularisation schemes within MMORF. This will establish that it is the SPRED penalty itself which imparts the desirable qualities seen in the tool comparison results.

3.3.1 Overlap Scores

3.3.1.1 Tools

Combined left/right hemisphere Jaccard Coefficient overlap scores are shown for the various registration tools in Figure 3.6. FLIRT, as expected, achieves the lowest levels of overlap, with mean overlaps varying depending on the region being considered. The “gap” in overlap between FLIRT and the nonlinear methods provides a useful yardstick against which to gauge the differences between the methods. All of the nonlinear methods, again as expected, improve the degree of overlap considerably. ANTs-MSQ has the lowest average overlap of the nonlinear methods. FNIRT has the second lowest overlap scores on average, although in certain areas it is outperformed by ANTs-MSQ. Note that the FNIRT results are significantly better than in previously reported comparisons (e.g. Ou et al. (2014) who used the parameters from A. Klein et al. (2009), which were unsuited for skull stripped data), but are in line with those from the tool’s author (Andersson et al. 2019). ANTs-CC and MMORF perform the best and equally well on average, but one or the other performs better in individual areas. These observations are supported when one considers the distribution of overlap scores combined across the entire brain in Figure 3.7 which display the same trends.

It is perhaps surprising that the two versions of ANTs produce such different results, given that they differ only in their choice of similarity metric. However, this is in line with what the creators of ANTs report themselves (Avants et al. (2011))

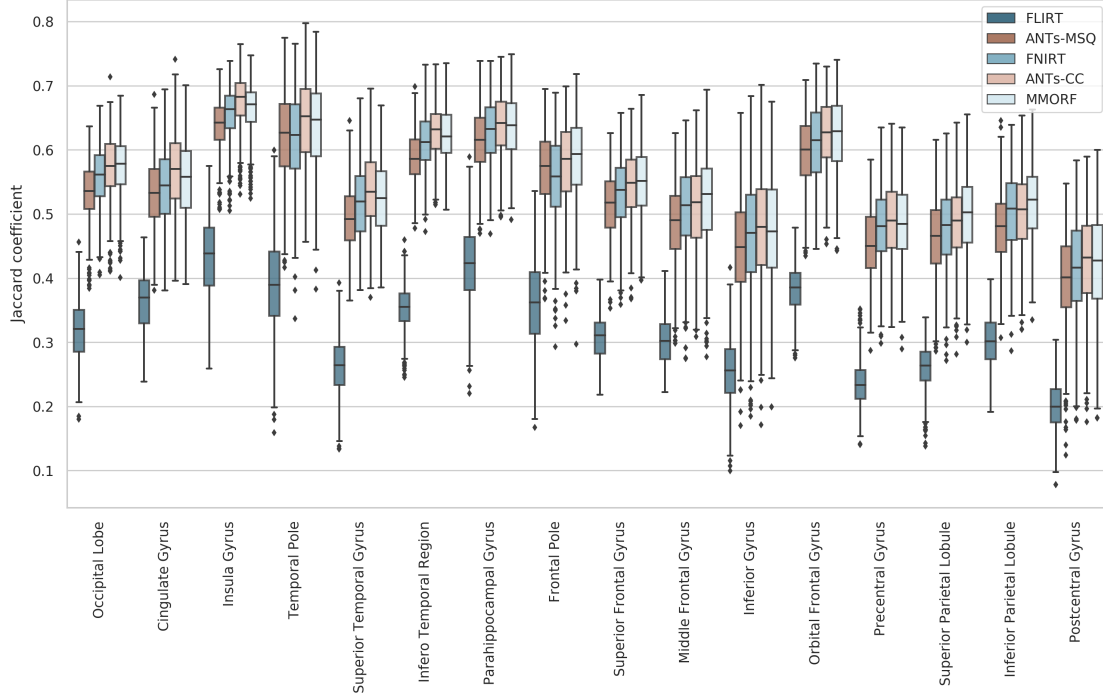


Figure 3.6: Jaccard Coefficient region-wise comparison (tools) - Jaccard coefficients of 16 regions (32 original cortical segmentations combined left/right) for each of the 4 nonlinear and 1 linear registration tools. The FLIRT result provides a baseline for evaluating the improvement in overlap between linear and nonlinear methods. Overall ANTs-CC and MMORF perform similarly. In some areas ANTs-CC produces better results (e.g., cingulate gyrus), and in some area MMORF performs better (e.g., inferior parietal lobule). In all cases, both methods outperformed ANTs-MSQ and, to a lesser extent, FNIRT.

on a different segmentation data set. I therefore believe this to be an accurate representation of their respective performance capabilities.

Because the best overlap performance I could achieve with ANTs-MSQ was significantly worse than that of MMORF and ANTs-CC I did not consider it meaningful to include it further in the comparison of warp metrics (Jacobian determinant and CVAR).

I have, however, included FNIRT in one of the warp-comparisons, even though it did perform slightly worse than ANTs-CC and MMORF. The reason for this is twofold. First, because the difference in performance was considerably less than for ANTs-MSQ. Second, because it uses the same image similarity metric and warp-construction as MMORF (differing only with respect to the strategy for ensuring invertibility and regularisation) I consider it to be the state-of-the-art

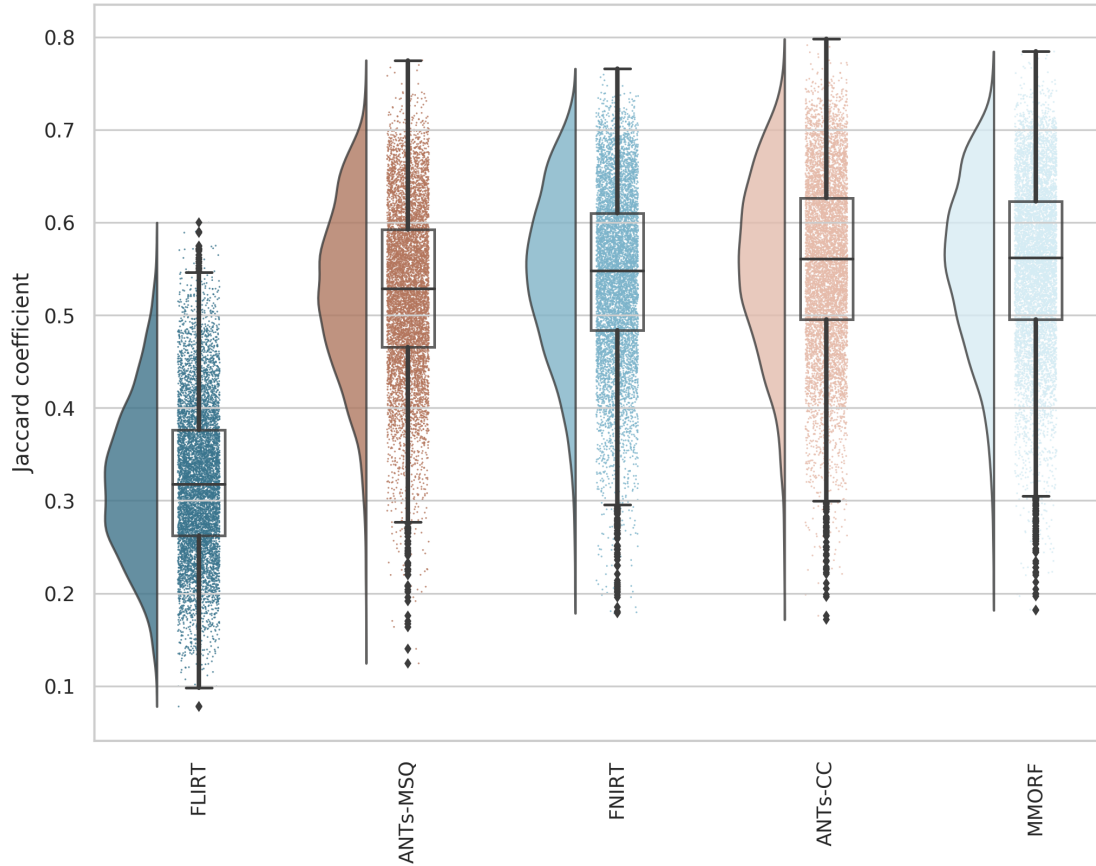


Figure 3.7: Jaccard Coefficient combined comparison (tools) - Distribution of Jaccard coefficients combined across all regions and subject pairs for each of the 4 nonlinear and 1 linear registration tools. This confirms the trends observed in Figure 3.6, with MMORF and ANTs-CC performing best, followed by FNIRT and then ANTs-MSQ. The MMORF and ANTs-CC distributions are remarkably similar, despite the slight region-to-region differences in performance.

for what can be achieved within the small deformation framework when using conventional regularisation schemes.

I now present a more detailed comparison between MMORF with SPRED and ANTs-CC, in order to demonstrate the performance of the main method of interest against the state-of-the-art, before presenting how different regularisation schemes within MMORF compare.

The joint distribution of overlap scores for MMORF and ANTs-CC for all regions and warps shown in Figure 3.8 confirms that the two methods are on average very comparable in terms of registration accuracy. This allows for a meaningful comparison of anatomical plausibility, conditioned on registration accuracy.

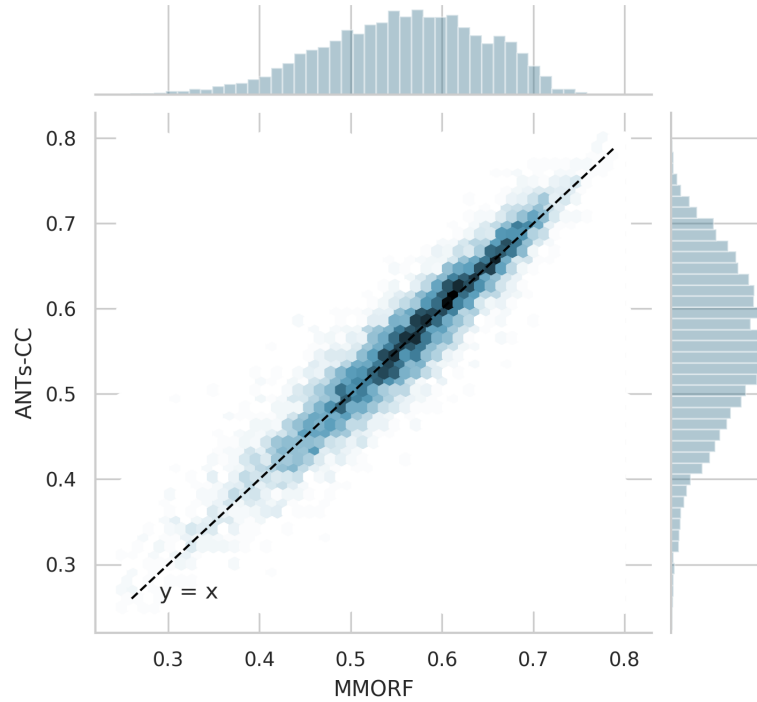


Figure 3.8: MMORF vs ANTs-CC Jaccard coefficients - Joint histogram of Jaccard coefficients across all regions and warps for MMORF and ANTs-CC. Note the tight, linear relationship, supporting the observation from Figure 3.6 that MMORF and ANTs-CC have very comparable overall performance across the NIREP dataset.

3.3.1.2 Regularisers

Combined left/right hemisphere Jaccard Coefficient overlap scores are shown for the three regularisation methods implemented within MMORF in Figure 3.9. No one method consistently outperforms the others - similar to what was seen in the between-tools results. This is confirmed when considering all regions combined across subjects, as shown in Figure 3.10. All three methods have approximately the same median overlap score. The Jacobian determinant penalised distribution displays slightly more of a central tendency, while the SPRED and local rigidity distributions are remarkably similar.

From these results it is clear that the choice of regulariser need not negatively impact registration accuracy, and therefore other aspects of the deformations can be meaningfully compared

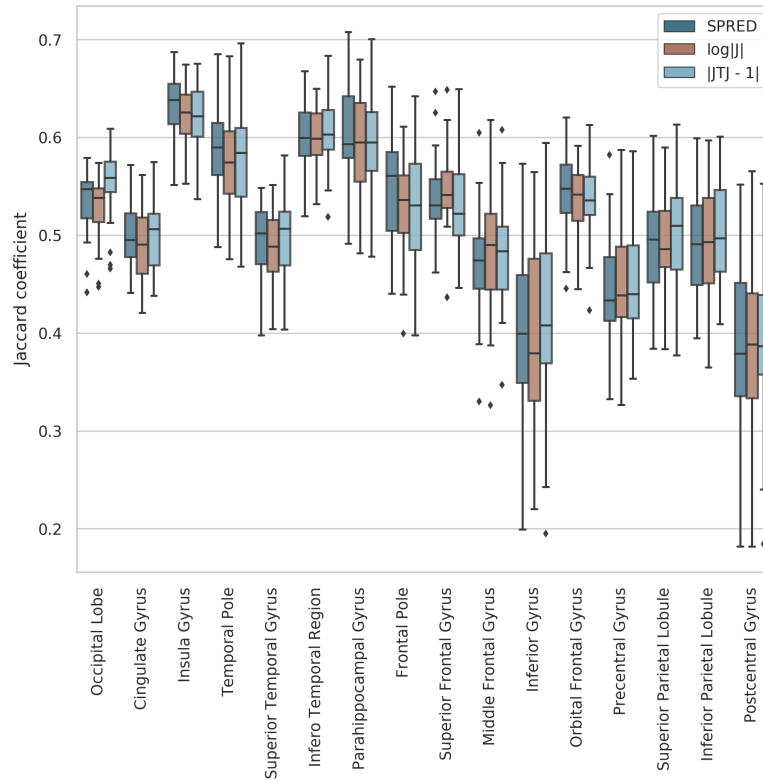


Figure 3.9: Jaccard Coefficient region-wise comparison (regularisers)1 - Jaccard coefficients of 32 cortical segmentations for registration to subject NA01 of the NIREP dataset. No one method was consistently superior or inferior across all regions leading to comparable results between the methods in terms of overlap scores.

3.3.2 Jacobian Determinant Distributions

3.3.2.1 Tools

Figure 3.11 shows the Jacobian determinant distribution of the warps generated with MMORF, ANTs-CC and FNIRT for a randomly selected image pair. I show both the non-log and log distributions since they highlight slightly different behaviours. It should be noted that although these warps represent a single pair of subjects, I could have chosen any of the 240 warps and the relationship would remain essentially the same.

The Jacobian determinant distribution for FNIRT has a shape that is quite distinct from that of MMORF and ANTs-CC. It is seen very clearly in the non-log distribution where instead of tapering off smoothly towards zero there is an almost linear drop. This is most likely due to FNIRT projecting its warps onto the

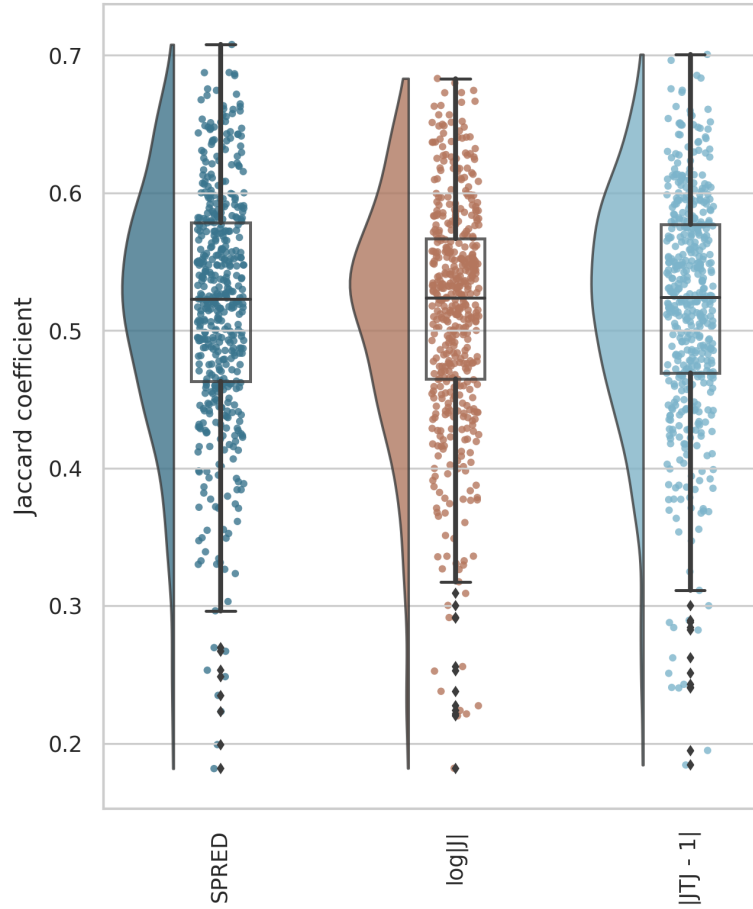


Figure 3.10: Jaccard Coefficient combined comparison (regularisers) - Distribution of Jaccard coefficients combined across all regions and subjects pairs for each regularisation penalty. This confirms what is seen in Figure 3.9, that all methods have comparable overlap accuracy on average.

nearest B-Spline field which has no negative Jacobian determinants (Karacali and Davatzikos (2004)). The shapes of the distributions for MMORF and ANTs-CC are more similar, but with ANTs-CC having a notably greater dispersion.

Of the three methods that had the most similar overlap scores it is clear that FNIRT causes substantially greater volume distortions, in addition to having slightly lower overlap scores. I have therefore dropped FNIRT from further analysis at this stage and focus on MMORF and ANTs-CC.

From the distribution of log-Jacobian determinants, the range between the 5th and 95th percentiles for each method can be calculated. This measure was selected as it is the *width* of the distribution, rather than its *mean*, which acts as a meaningful

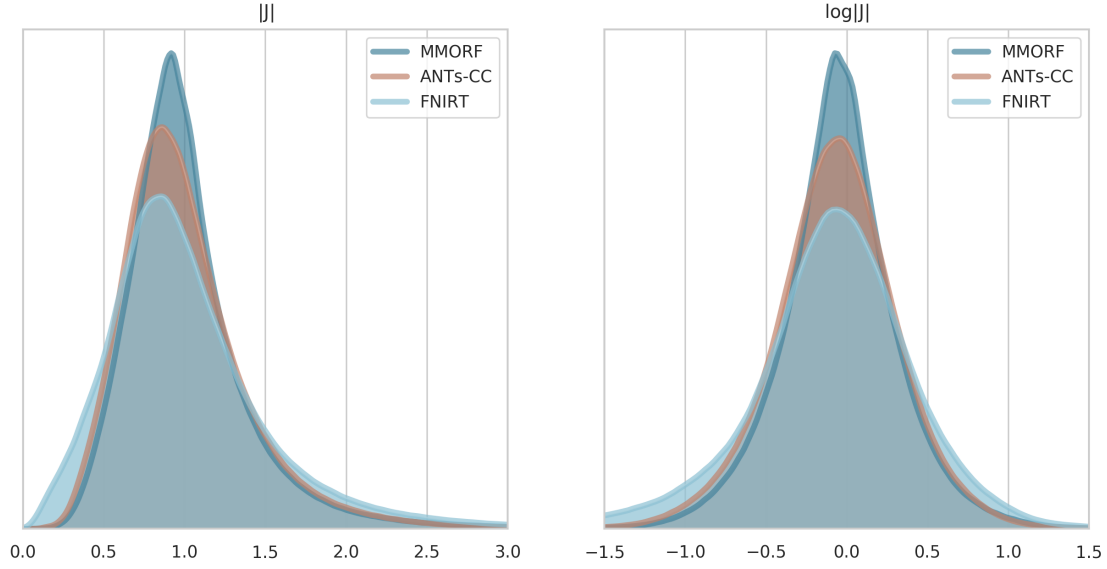


Figure 3.11: Jacobian determinant distributions - Jacobian determinant and log-Jacobian determinant distribution of a randomly selected registration pair from the NIREP dataset. Only values within the brain itself have been considered. As desired, all three methods produce Jacobian determinants with few to no negative values. There are distinct differences in appearance though. FNIRT is the most different, with a visibly heavier negative log-Jacobian tail. This is likely an effect of projecting the Bending-Energy regularised registration onto a field without negative Jacobian determinants. ANTs-CC displays the next widest log-Jacobian distribution, but with more evenly weighted tails.

indication of how aggressively the image is being deformed. This was done for every warp, and the results are summarised in Figure 3.12. It can be seen that both the mean and the dispersion of the ranges are considerably greater for ANTs-CC than for MMORF. Figure 3.13 shows a scatter plot of the log 5th-95th percentile range for each of the 240 registrations for ANTs-CC and MMORF. In all but 4 cases the percentile range is smaller for MMORF.

3.3.2.2 Regularisers

The 5th to 95th percentile ranges for the three regularisation methods are presented in Figure 3.14. Not unexpectedly, the regularisation method which penalises the Jacobian determinant directly has the consistently lowest range. The local rigidity penalty displays the poorest performance, with consistently more extreme Jacobian determinant ranges. The SPRED penalty falls between the other two methods.

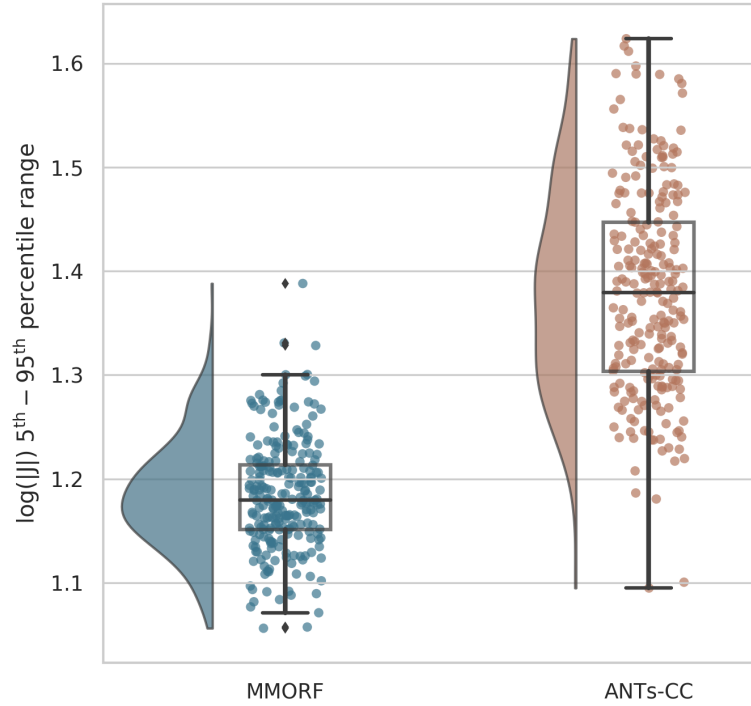


Figure 3.12: Log-Jacobian determinant range distributions (tools) - The 5th to 95th percentile is used as a summary measure for the log-Jacobian determinant distributions in Figure 3.11. The distribution of this percentile range over all 240 registrations is shown for the two best performing methods. ANTs-CC displays a significantly higher range on average, as well as a greater dispersion of ranges when compared to MMORF.

3.3.3 CVAR Distributions

3.3.3.1 Tools

Unlike the Jacobian determinant, the mean (across sample points within the brain) of the CVAR is a meaningful statistic. I therefore use that as the summary measure of *shape* distortion for a given warp. Figure 3.15 demonstrates that the mean is both lower on average, and has smaller dispersion for MMORF compared to ANTs-CC.

3.3.3.2 Regularisers

The mean CVAR values for the three regularisation methods are presented in Figure 3.16. It is immediately clear that penalising the log-Jacobian determinant directly has resulted in significantly more shape distortion than the other two methods. The local rigidity penalty has led to less distortion than the log-Jacobian,

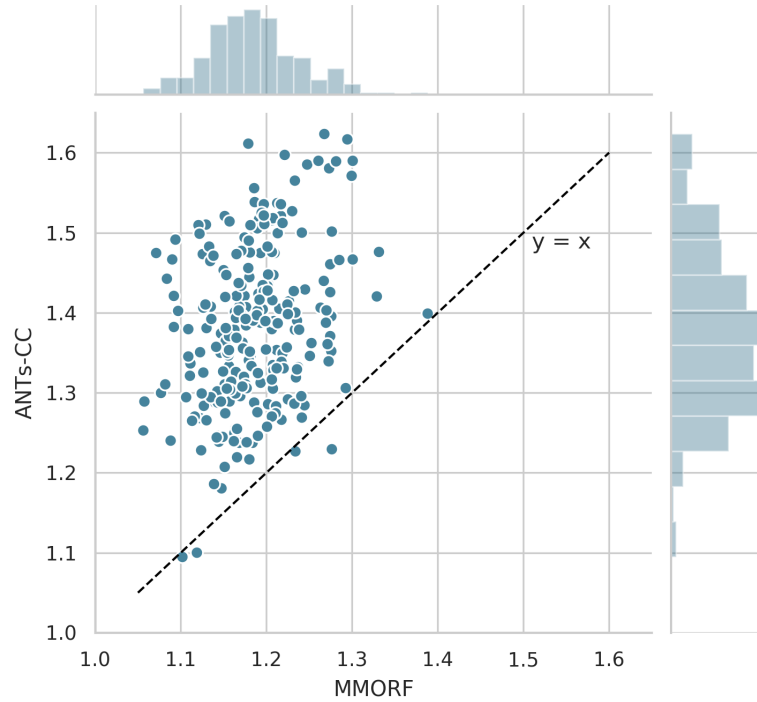


Figure 3.13: MMORF vs ANTs-CC log-Jacobian determinant range - Scatter plot of MMORF and ANTs-CC log-Jacobian determinant 5th to 95th percentile ranges. The key feature here is that ANTs-CC produces systematically wider ranges compared to MMORF, with only 4 of the 240 registrations favouring ANTs-CC. This is despite ANTs-CC and MMORF producing comparable Jaccard coefficients overall, as shown in Figure 3.8.

but it is still consistently higher than the SPRED penalty. SPRED is best able to control shape distortions.

3.3.4 Spatial Maps

3.3.4.1 Tools

In the previous section I showed that of the methods with comparable overlap scores ANTs-CC caused substantially more volumetric distortions than MMORF. In order to better understand the source of that difference I present the spatial maps of the Jacobian determinants and of the CVAR. Figure 3.17 shows maps of Jacobian determinants and figure 3.18 of CVAR for the same randomly selected pair of subjects as Figure 3.11. There are clear visual differences between the results of the two methods. In areas where the T1-weighted signal intensity has strong contrast and carries information about the tissue type, such as along the cortex,

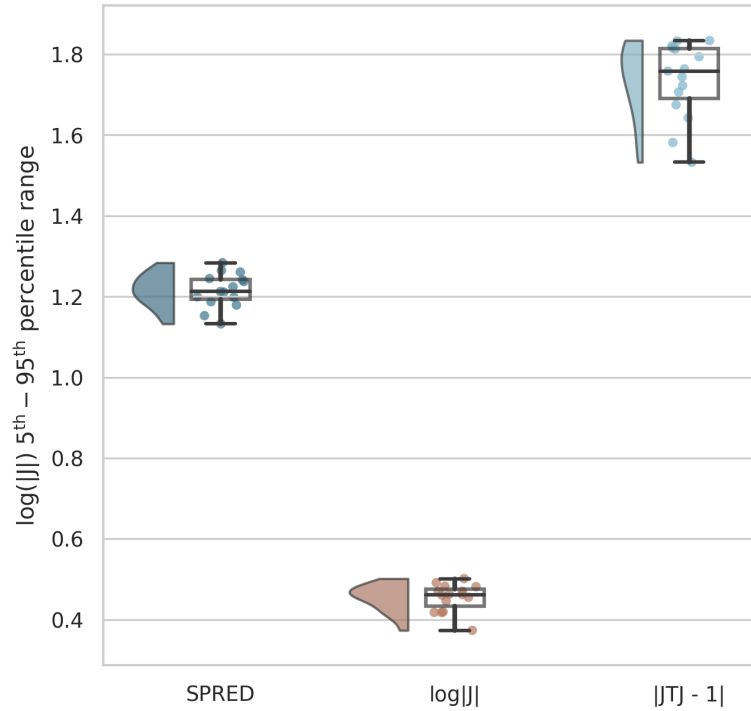


Figure 3.14: Log-Jacobian determinant range distributions (regularisers) - 5th to 95th percentile range of Jacobian determinants for warps from all subjects to NA01 of the NIREP dataset. As would be expected penalising the log-Jacobian determinant directly leads to a much tighter distribution of Jacobian determinants about 1 than penalising the JSVs. The local rigidity penalty performs worst in this regard, with consistently more extreme Jacobian determinants.

both methods produce similar looking maps with varying amounts of expansion and compression. Where they differ in appearance is predominantly within areas displaying a relatively flat T1-weighted signal profile, such as in white matter. Here MMORF produces smoothly changing values with magnitudes close to 1. ANTs-CC by contrast displays highly oscillatory behaviour, with a higher proportion of values deviating away from 1.

As a T1-weighted volume contains minimal information within the white matter, there is little reason to believe that the deformation should be anything other than smooth within these regions. Clearly this has not detrimentally affected the performance of ANTs-CC in terms of achieving high cortical overlap scores. However, if this warp was used to transform a modality rich in white-matter information (such as diffusion tensor imaging) the results could potentially be very deleterious.

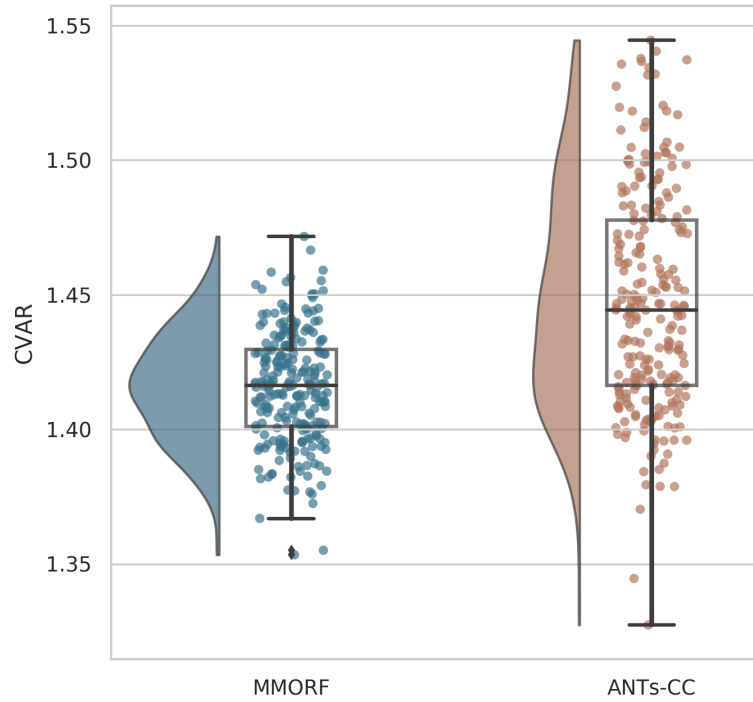


Figure 3.15: Mean CVAR distributions (tools) - Distribution of mean CVAR within the brain over all 240 registrations is shown for the two best performing methods. ANTs-CC displays a higher mean CVAR on average, as well as greater variance when compared to MMORF.

By contrast, the MMORF result is more balanced. In areas where there is a large amount of information to drive the deformation (tissue boundaries) the warps are correspondingly larger and more variable, whereas areas with minimal information are left relatively unchanged.

When looking at figure 3.18 it should be noted that the particular (randomly selected) registration I chose for displaying spatial maps happens to be one of 65 (out of 240) for which the mean CVAR was greater for MMORF than for ANTs-CC.

3.3.4.2 Regularisers

Spatial maps of the Jacobian determinant of the warp between a random subject and subject NA01 are shown in Figure 3.19 for the three regularisation methods implemented within MMORF. These give a more qualitative understanding of the distributions in Figure 3.14. The map for the log-Jacobian determinant regulariser is remarkably flat, demonstrating that this method was able to very effectively control

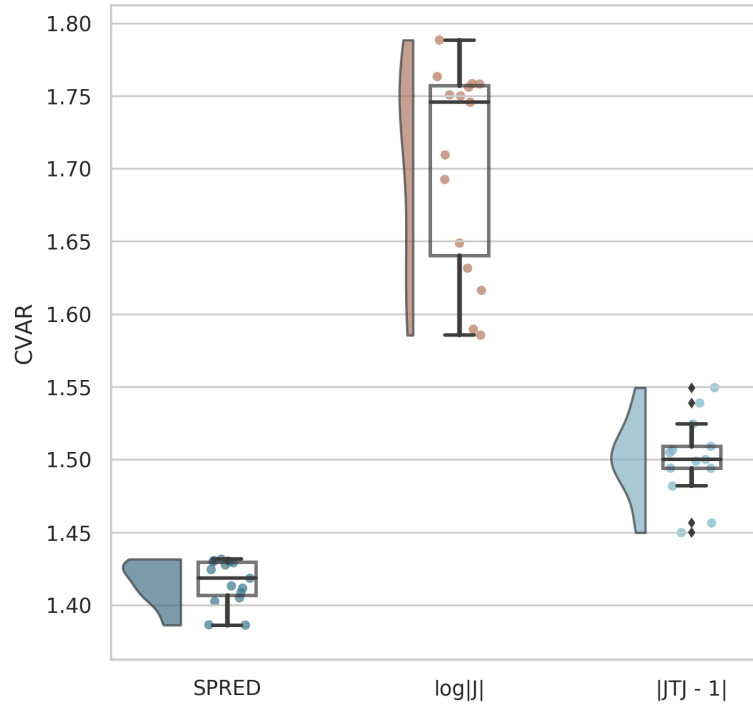


Figure 3.16: Mean CVAR distributions (regularisers) - Distribution of mean CVAR within the brain for warps from all subjects to NA01 of the NIREP dataset. Observe that SPRED clearly best preserves the original shape of the underlying anatomy during deformation. The local rigidity penalty, while not quite as good as SPRED, still consistently outperforms the log-Jacobian determinant penalty.

volumetric changes. What changes are visible appear to be largely confined to single tissue types - note for example the shrinkage of the right ventricle - representing gross differences in tissue type volumes between subjects.

The SPRED penalty map is more conventional, with the largest changes visible in the more variable regions such as the cortex as expected. In areas of low contrast, such as the white matter, the deformation is minor and varies smoothly.

The local rigidity map is perhaps the most striking. There are substantial areas (appearing dark on the map) where the Jacobian determinant is either very close to zero, or has gone negative. This is a feature not captured by the 5th- 95th percentile range, but which is clearly visible here. Such a result is not wholly unexpected, given that one of the key difference between this regularisation method and SPRED is that it does not guarantee diffeomorphism - however it is a wholly unrealistic deformation and represents a breakdown of anatomical plausibility.

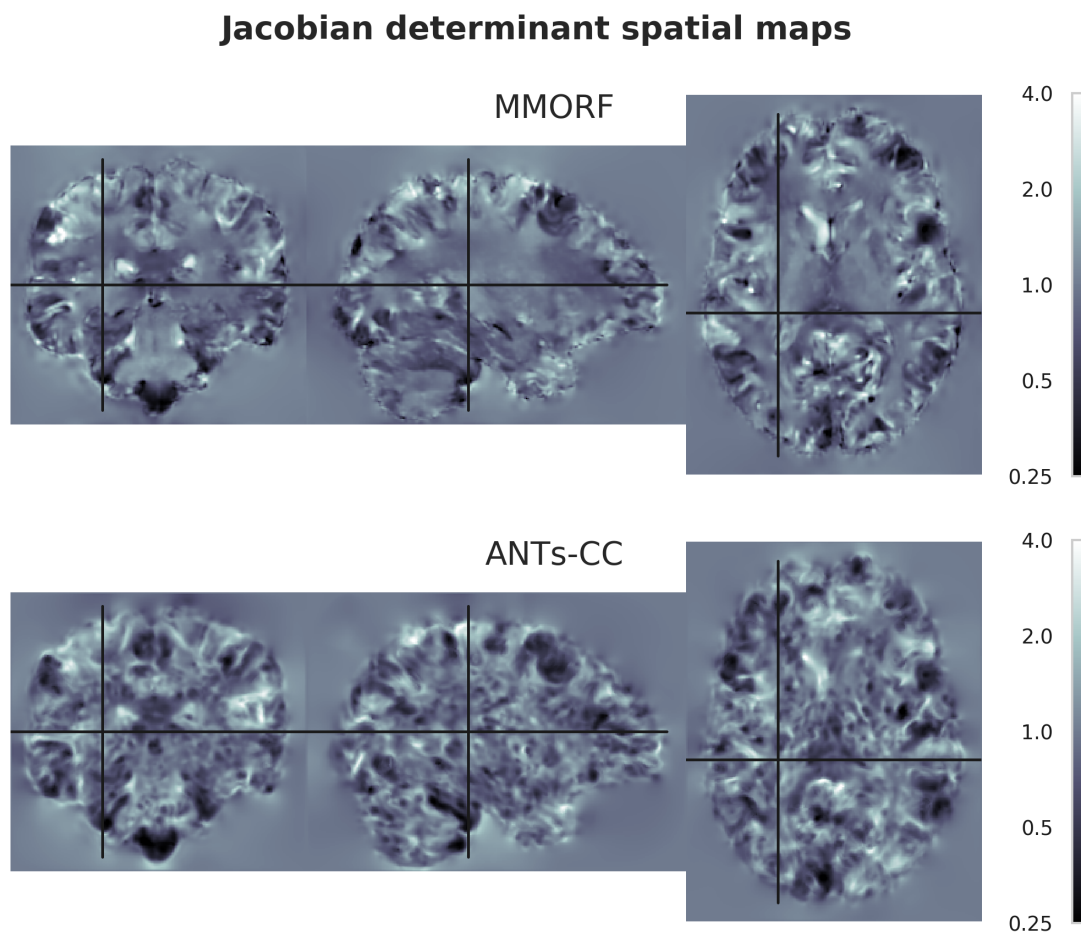


Figure 3.17: Jacobian determinant spatial maps (tools) - Spatial distribution of Jacobian determinants for MMORF and ANTs. Note the distinct visual differences within the white matter for ANTs-CC. The SPRED penalty has resulted in smooth changes within this region, whilst ANTs-CC has produced high frequency variations. As a T_1 image contains little information within the white matter, the anatomical plausibility of those rapid variations is not clear. By maintaining smoothness within the white matter, MMORF produces a result that is more anatomically believable given the information present within the images being registered. Importantly, the irregularity introduced by ANTs-CC is not necessary for achieving high overlap scores, as Figures 3.6 and 3.8 show. Finally, whilst this may not have any deleterious effect on overlap scores of grey matter regions, the effect of transforming a modality rich in white matter information through such a warp could be significant.

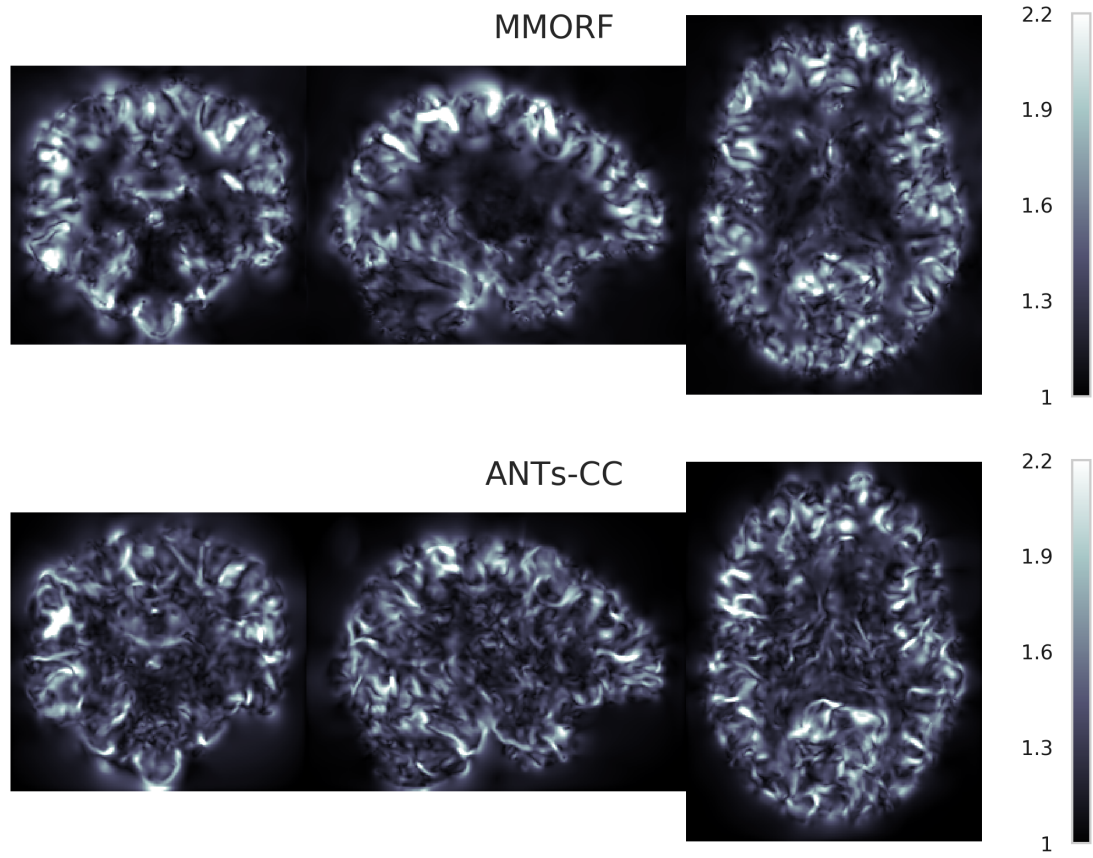


Figure 3.18: CVAR spatial maps (tools) - Spatial distribution showing the Cube-Volume Aspect Ratio (CVAR) for MMORF and ANTs-CC. CVAR is a measure of shape change which extends the concept of aspect ratio to arbitrary dimensions, and is described in detail in 3.2.5. Values above 1 correspond to increasing deviations from the original shape. It represents the maximum linear relative change along any direction of the deformed voxels. Note that the differences in appearance between MMORF and ANTs-CC are very similar to that in Figure 3.17.

Spatial maps of the local CVAR value are shown in Figure 3.20. The log-Jacobian determinant penalty CVAR map contrasts starkly with the flat Jacobian determinant map in Figure 3.14 - the widespread high values confirm that control of the Jacobian determinant has come at the expense of excessive shape deformation.

The local rigidity penalty has done an excellent job of preserving shape where little deformation was required, such as in the white matter. However, where larger deformations are required, such as in the cortex, the distortion increases rapidly - particularly in areas where diffeomorphism has broken down.

The SPRED penalty performs considerably better than the other methods.

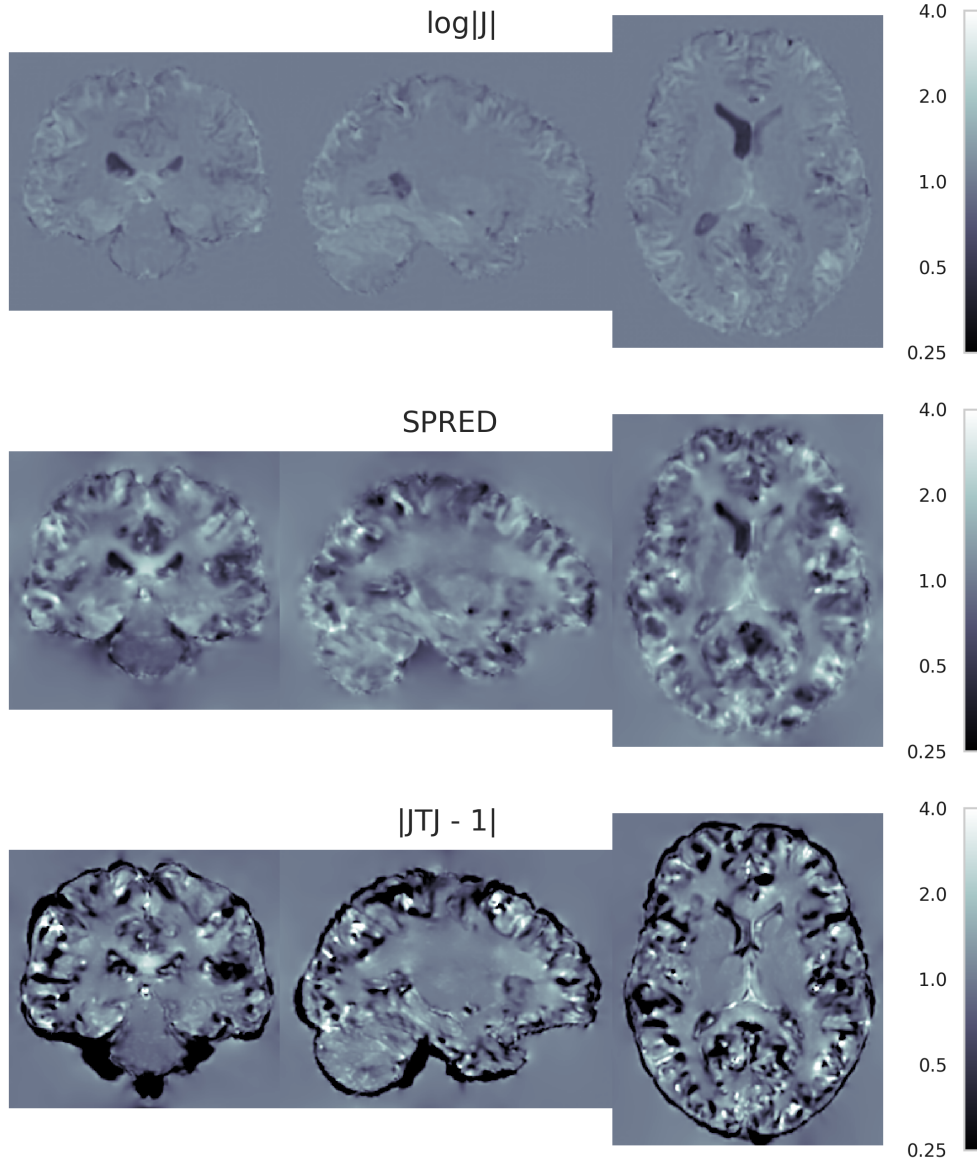


Figure 3.19: Jacobian determinant spatial maps (regularisers) - Spatial distribution of Jacobian determinants for SPRED, $\log |\mathbf{J}|$, and local rigidity regularisation. Penalising the volumetric changes directly clearly leads to the flattest Jacobian determinant maps. As the local rigidity penalisation does not guarantee diffeomorphism you can observe a number of areas where the Jacobian determinant is either very close to 0, or negative, particularly in and around the cortex. SPRED is somewhere between the other two methods, which agrees with what is observed in Figure 3.14.

CVAR is well controlled throughout the brain, varying smoothly with small areas of larger distortion confined to the cortex.

Finally, Figure 3.21 shows a grid deformed by each method. The undesirable nature of the log-Jacobian determinant penalty warps is immediately obvious - controlling the volumetric changes has come at the expense of an extremely convoluted warp with highly implausible shape distortions.

The similarities between the local rigidity and SPRED penalties are evident, as both methods have deformed the grid in a very similar way. The exception is those areas, particularly noticeable as blobs in the cortex, where topological folding has occurred.

These examples clearly show that simply penalising the Jacobian determinant is insufficient to ensure plausible deformations, and that the additional complexity of the SPRED penalty compared to a simpler local rigidity penalty is justified.

3.4 Discussion

I have implemented and investigated the properties of a regularisation penalty based on the assumption that the singular values of the local Jacobian field are drawn from a log-normal distribution. The regulariser has some very attractive properties, but is computationally very expensive. This meant that the implementation in the papers which first proposed such a penalty (Ashburner et al. 1999; Ashburner et al. 2000) used a voxel-by-voxel optimising scheme rather than a global optimiser, which can potentially cause order effects and convergence to a non-optimal solution. I implemented the regulariser utilising GPU parallelisation, using a B-spline basis to represent the warps, and a Gauss-Newton optimiser, resulting in run-times of under half an hour for a full multi-pyramid registration. This has allowed me to run a large number of registrations and compare the results to one of the best performing algorithms in popular use (Avants et al. (2008), A. Klein et al. (2009)). The main finding is that I have been able to show that it performs as well as ANTs in terms of overlap scores, and that it achieves that with significantly reduced volumetric and shape distortion.

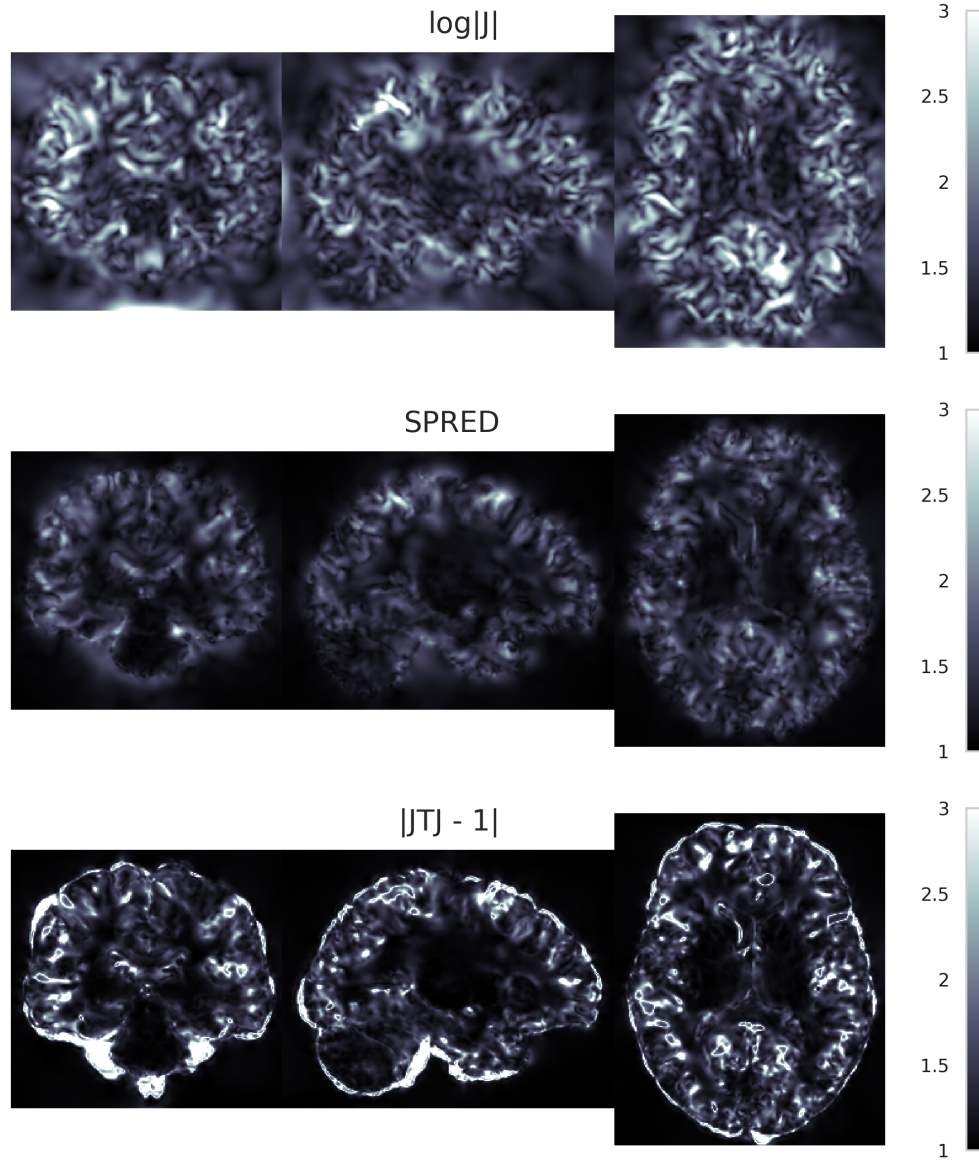


Figure 3.20: CVar spatial maps (regularisers) - Spatial distribution of CVar values for SPRED, $\log |\mathbf{J}|$, and local rigidity regularisation. It is immediately clear that the flat Jacobian determinant maps for the $\log |\mathbf{J}|$ penalty are accompanied by a marked increase in CVar, confirming that volumetric control has come at the cost of extreme shape changes. Local rigidity penalisation leads to excellent preservation of shape within the white matter, but shows extreme distortion within some areas the cortex. SPRED performs considerably better than either of the other methods, confirming my observations in Figure 3.16

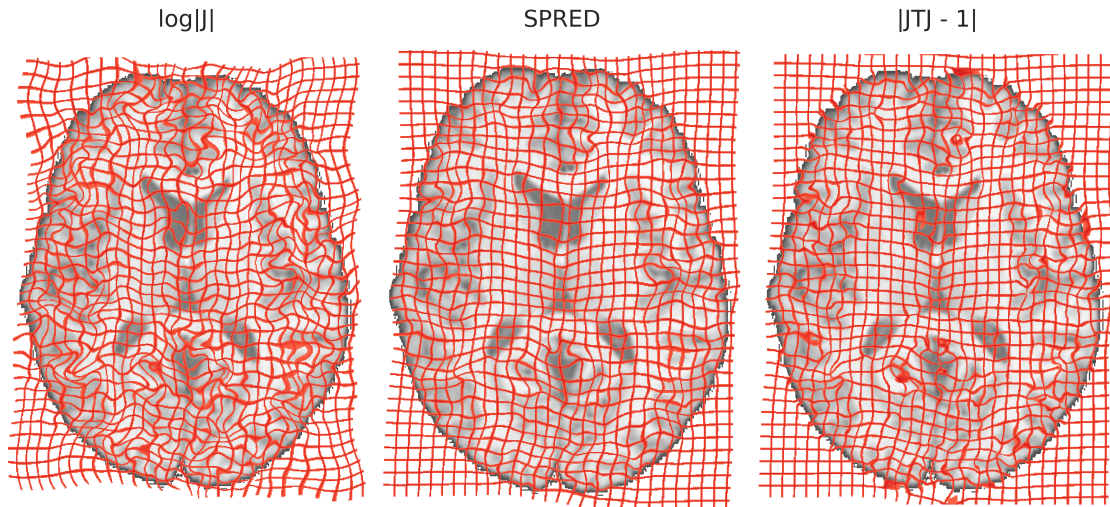


Figure 3.21: Comparison of deformed grids - Warped grid showing the effect of the different regularisation penalties. Penalising the Jacobian determinant alone has led to extremely convoluted warps, whereas penalising the Jacobian singular values results in smoothly varying warps which are far more plausible. Interestingly, penalising local rigidity has led to a very similar deformation to SPRED, with the exception of areas where topological folding or collapse has occurred. These areas are particularly evident within the cortex and at tissue boundaries.

3.4.1 Relationship to Earlier Work

A number of related penalties beginning with different suppositions or models have been suggested since the paper from Ashburner et al. (1999). The majority of them suffer from the same problems as the original work - that these types of penalties are highly computationally intensive to implement, and therefore do not find their way into useable registration tools. Pennec et al. (2005) suggested a similar penalty function based on the concept of a Riemannian manifold for the allowed (diffeomorphic) warps, but the characteristics of that penalty were never explored in detail as it was not incorporated into a full registration framework. Other regularisers (e.g., Burger et al. (2013)) with similarities to SPRED have been suggested which draw on models for hyperelastic materials. These are models that describe the stress-strain relationships of, for example, rubber - which are not well described by linear elastic models. Such models have also been used as regularisation for correcting susceptibility artefacts in diffusion-weighted MRI (Ruthotto et al. (2012)). Similar hyperelastic models have additionally been used to model cortical

growth in the developing brain (Knutsen et al. 2010) and for regularisation of surface based cortical registration (Robinson et al. 2018).

It should be noted that neither the present work, nor that in Ashburner et al. (1999) or Ashburner et al. (2000), is explicitly based on hyperelastic models. Nor do I believe that hyperelastic models necessarily provide meaningfully descriptive generative models for explaining anatomical differences between subjects. Realistic models of cortical growth patterns (see for example Van Essen (1997)) are likely too complex to be described by a few material constants. The reasoning behind the SPRED regularisation function is therefore purely statistical - resulting in a function with desirable deformation characteristics, capable of producing smooth and plausible warps in both regions with both high and low levels of anatomical information.

3.4.2 Beyond Enforcing Diffeomorphic Warps

It is widely agreed that invertibility is a desirable property in a nonlinear spatial transform, and much of the discussion in image registration research focusses on this element of the deformations. There are two principal ways in which diffeomorphism can be enforced

Warp Construction Any warp that is a composition (integration in the limit) of diffeomorphic warps is itself diffeomorphic. Hence, methods that estimate warps as compositions of many small updates will by construction ensure that the end result is invertible.

Warp Penalisation A highly nonlinear penalty function that goes to infinity as the local Jacobian determinant approaches zero will allow large deformations while maintaining invertibility. It should be noted that functions such as membrane energy or bending energy do not fall into this category.

For completeness both of these approaches will be discussed in more detail below.

Diffeomorphism should not be treated as the be all and end all of desirable warp characteristics. A warp can be invertible and yet highly unrealistic in that it causes large amounts of volumetric and/or shape distortions. For this reason most

methods that enforce invertibility using one of the techniques mentioned above will combine that with one or more “traditional” regularisers that enforce smoothness. Ashburner and Ridgway (2013) has shown very convincingly that even within the space of diffeomorphic warps one will obtain very different solutions depending on the exact details and weights of the additional regularisers.

Hence, what I aim to achieve with this choice of regulariser goes well beyond simply ensuring diffeomorphic warps. The aim is to find the maximally plausible, in terms of volume and shape distortions, of all possible warps within the space of diffeomorphisms. And to do this with a single penalty function (and a single weight) that achieves the joint objective of ensuring invertibility and simultaneously minimising volume and shape distortions.

I recognise that a judicious choice of forms and weights of additional penalty functions within inherently diffeomorphic frameworks, such as LDDMM and viscous fluid based methods, could potentially find an equally advantageous solution. But it is nevertheless the case that when comparing SPRED to a state-of-the-art diffeomorphic method I was able to obtain invertible warps with equally good overlap scores and significantly less volume and shape distortions. Furthermore, there is no intrinsic superiority of inherently diffeomorphic methods over any other method that also guarantees invertibility, but achieves that with equally good, or better, registration accuracy.

3.4.2.1 Diffeomorphism by Warp Construction

Methods such as ANTs fall into the category of inherently diffeomorphic warps. In particular, ANTs is an example of a greedy approximation of the general LDDMM method. LDDMM methods seek to find a time varying velocity field which minimises a metric based on total path length in the space of diffeomorphisms. Originally introduced by Beg et al. (2005), these methods guarantee that the total deformation remains diffeomorphic when the velocity field is integrated over sufficiently small timesteps. A downside to these methods is the large number of parameters which need to be estimated at each timepoint. Tools such as DARTEL (Ashburner 2007)

seek to overcome this by instead parametrising a stationary velocity field, thereby greatly reducing the parameter space, but at the expense of potentially larger path lengths. Subsequently, Geodesic Shooting (Ashburner and Friston 2011) based methods have been able to reformulate the original LDDMM problem such that instead of estimating the entire sequence of time-varying velocity fields, only the initial velocity need be estimated, thereby greatly reducing both the parameter space and convergence time of the optimisation.

Whilst all of these methods are capable of ensuring warps remain diffeomorphic, they differ from SPRED in that the prior on which they are based is that deformations follow the smallest energy path-lengths, rather than deformations conserving the shape of underlying anatomy. As such, diffeomorphism is an intrinsic rather than a controlled property of the transformation model. It is certainly not impossible to include explicit regularisation of shape changes in these models, however it is not the norm.

3.4.2.2 Diffeomorphism by Warp Penalisation

An alternative method of enforcing diffeomorphism is by using a penalty function that goes to infinity as the local Jacobian determinant approaches zero.

Most commonly, the Jacobian determinant itself is used as a hard constraint that bounds the allowed range (Haber and Modersitzki 2004; Sdika 2008; Sdika 2013; Haber et al. 2010), or fixes it to a particular value (Mansi et al. 2011), or a smoothly varying function of the Jacobian determinant is used as a soft constraint (Borzi et al. 2003; Yanovsky et al. 2007; Leow et al. 2007; Heyde et al. 2016; Mang et al. 2018). It should be noted that when modelling the warps as a velocity field the local divergence can be used as a proxy for the Jacobian determinant.

The crucial difference between SPRED and those penalties based on a function of the Jacobian determinant is that the latter *only* penalises volume distortions. Not only does that mean that shape changes are not penalised, in practice such a function will result in a method where very large shape changes are used to circumvent the volume change limitations. In 3.3 I show the results obtained with a

Jacobian determinant penalty function ($\sum_n v (1 + |\mathbf{J}_n|) \log^2 |\mathbf{J}_n|$) where the weight was calibrated so as to yield the same overlap score as the SPRED penalty. It can be seen that the Jacobian determinant penalty yields equally good overlap scores (Figures 3.9 and 3.10) and results in warps that are diffeomorphic, with a very narrow range of Jacobian determinants (Figures 3.14 and 3.19). However, this has been achieved by completely ignoring shape changes, and has resulted in much larger CVARs (Figures 3.16 and 3.20). Looking at a randomly selected warp (Figure 3.21) it is very clear that the Jacobian determinant penalty has resulted in lots of gratuitous distortions.

This is the reason that many groups have combined a hard or soft constraint on volume changes with an additional regulariser (see for example Haber et al. (2010), Yanovsky et al. (2007), Leow et al. (2007) or Mang et al. (2018)). While that may work well, it means that an additional weight factor needs to be empirically determined.

There is another group of algorithms (Loeckx et al. 2004; Ruan et al. 2006; Staring et al. 2007; Modersitzki 2008; Mang and Biros 2016) that use functions that penalise deviations from local rigidity (*i.e.* anything beyond local translation and rotation). In that respect they have similarities to SPRED, but they have mostly been used to enforce near-total rigidity in selected parts of images containing a mix of rigid tissue (*e.g.*, bone) and non-rigid tissue (*e.g.*, muscle).

Within this category of registration methods, Loeckx et al. (2004) and Mang and Biros (2016) are probably the most closely related to SPRED of which I am aware. Mang and Biros (2016) acknowledge the issue of excessive shape changes and seek to combat it by explicitly penalising the shear of the velocity field. In 3.2.1.2 I show the relationship between the SPRED penalty and the local rigidity penalty of Loeckx et al. (2004) ($\sum_n v \|\mathbf{J}_n^T \mathbf{J}_n - \mathbf{I}\|_F$). I present the results of using this penalty, with the weighting calibrated so as to yield the same overlap score as SPRED. It can be seen that the local rigidity penalty yields equally good overlap scores (Figures 3.9 and 3.10), but it is unable to preserve diffeomorphism. Whilst the CVAR values

are mostly well controlled (Figures 3.16 and 3.20), looking at a randomly selected warp (figure 3.21) demonstrates that topology is not being preserved in the cortex.

3.4.3 Other Parallelised Algorithms

It is increasingly common for registration tools to employ some form of parallelisation, and it is worth contextualising my approach through comparison with some of these methods.

Some of the Insight Segmentation and Registration Toolkit (Yoo et al. 2002) methods use CPU multithreading on a single device to accelerate the portions of their code which solve the linear system of equations necessary to compute update steps. This is a simple method to implement, but gains tend to be modest.

A recently described (Mang et al. 2018) parallelised version of Mang and Biros (2016) utilises distributed-memory parallelism on multiple CPU nodes. It uses a Gauss-Newton optimisation strategy, and solves for a stationary velocity field transformation. The primary focus of the parallelism is in the *inversion*, rather than the *calculation*, of the Hessian.

Possibly the most closely related method to MMORF is NiftyReg (Modat et al. 2010) which employs a B-spline transformation and GPU parallelisation. NiftyReg differs in using a first order optimisation strategy and therefore does not require calculation of the Hessian. Reported performance improvements are very similar to MMORF, indicating comparable levels of code optimisation. Their impressive sub-minute runtimes are aided by a computationally simpler bending energy regularisation metric. Therefore a direct comparison of runtimes to MMORF with SPRED is not meaningful.

As discussed in Eklund et al. (2013), most applications of GPU parallelisation to medical image registration focus on speeding up existing methods. However, an interesting alternative is the development of more advanced registration methods which might otherwise have been rejected on the basis of computational complexity - and it is within this grouping that the SPRED penalty is placed.

3.4.4 Registration Accuracy

3.4.4.1 Tools

As I have previously stated, absolute registration accuracy is not the primary focus of this chapter. Instead it stands as a reference point for my discussion of anatomical plausibility, in that only two methods which are comparable in terms of registration accuracy can be meaningfully differentiated based on anatomical plausibility.

Based on the overlap scores in Section 3.3.1, it can be seen that MMORF and ANTs-CC are clearly the best performing tools in terms of registration accuracy. Additionally there is very little to differentiate between their performance as a whole. Note that the results for ANTs-CC are slightly better than those which have been previously reported (Ou et al. 2014) for the same validation data set. I therefore believe that the way I have used ANTs-CC has yielded a close to optimal performance on this dataset.

3.4.4.2 Regularisers

For the within-MMORF regulariser comparison, all regularisation methods led to very similar overlap scores making a subsequent comparison of the warp characteristics meaningful. The fact that such similar overlap scores can be achieved using such different regularisation strategies highlights the under-determined nature of the image matching metric, and therefore the important role that regularisation plays in constraining the space of possible solutions.

3.4.5 Anatomical Plausibility

3.4.5.1 Tools

Based on the summary 5th to 95th percentile log-Jacobian determinant range metric, MMORF is significantly more anatomically plausible than ANTs-CC. Figure 3.13 supports this argument, by showing that this is true for a vast majority of the registrations. In other words, for a given registration accuracy one is almost guaranteed to have smaller volumetric changes when using MMORF over ANTs-CC.

Another way of framing this result is to say that larger volume distortions are not a necessary trade-off for achieving high registration accuracy.

A striking feature of the Jacobian-range (figure 3.17) and CVAR maps (figure 3.18) are the seemingly gratuitous warps in white matter. This could potentially be a particular problem if one intends to use the structural registration to transform diffusion data. I believe this to be a strong case for extending the notion of anatomical plausibility beyond that of simply maintaining diffeomorphism and for building that notion into the registration.

Why this behaviour is observed in ANTs-CC is an interesting question in and of itself. At first this might be thought to be a case of insufficient regularisation, however the deformations generated by ANTs-MSQ do not display this same behaviour. Therefore I suggest that this is due to the somewhat *scale-free* nature of the CC metric. In other words, the gradient of CC does not depend on the amount of signal contrast present, only on the local correlation of that signal. What this means in practice is that regularisation on the level of the gradient cannot alter this behaviour, rather regularisation of the metric itself would be required. How to achieve this is beyond the scope of this work, however it highlights the importance of taking a considered approach to evaluating the believability of a deformation.

3.4.5.2 Regularisers

Two key observations can be made regarding the anatomical plausibility of the three regularisation penalties that were tested.

The first is that, despite its widespread use, the Jacobian determinant *by itself* is insufficient to evaluate how plausible a deformation is. Penalising the Jacobian determinant directly did indeed lead to the least amount of volumetric change (as evidenced by Figures 3.14 and 3.19), but this was at the expense of excessive shape deformation (as shown in Figures 3.16 and 3.20).

The second is that, despite its insufficiency, the Jacobian determinant is still a *necessary* measure for evaluating the anatomical plausibility of a deformation. The simpler local rigidity penalty, which did not explicitly constrain the Jacobian

determinants to remain positive, did limit shape deformations in the majority of the images - however in areas where larger deformations were required, diffeomorphism broke down leading to zero or negative Jacobian determinants which are biologically impossible.

The SPRED penalty strikes a neat balance between these two methods. I acknowledge that by combining the log-Jacobian determinant and local rigidity penalties, in a similar manner to Staring et al. (2007), it may be possible to overcome the individual limitations of each penalty. However, this would introduce additional regularisation weights to tune. In this respect the SPRED penalty is far more elegant, as it effectively controls both volumetric and shape changes, preserves topology, and requires the use of only a single tuning parameter.

3.4.6 Performance Summary

The SPRED penalty allows MMORF to overcome the oft-quoted limitations of small deformation frameworks and achieve levels of registration accuracy comparable to state-of-the-art large deformation tools such as ANTs-CC. MMORF achieves this while simultaneously producing warps which are consistently less aggressive in terms of both volume and shape distortions. Finally, given the information present in a structural scan, the spatial distribution of the log-Jacobian determinants MMORF produces appear more plausible compared to ANTs-CC, which produces variations that are not obviously necessary.

Overall I conclude that SPRED as implemented within MMORF produces warps which can be used with a high level of confidence. MMORF's registration accuracy is on par with state-of-the-art volumetric methods, and therefore no sacrifice need be made in terms of maximising data consistency.

3.5 Conclusion and Outlook

I have demonstrated that the SPRED penalty is capable of matching the registration accuracy (as measured using overlap scores of manually segmented cortical regions) of the most well established large-deformation (ANTs-CC) framework available

today. Additionally, I find that the results from using the SPRED penalty are consistently more anatomically plausible both in terms of Jacobian determinants and CVARs. I have leveraged GPU parallelisation in order to make optimisation of the penalty computationally tractable, allowing this method to be of practical benefit as part of a usable registration framework.

I have shown that using a regularisation function such as SPRED can address the shortcomings of elastic small deformation frameworks, yielding the best of both worlds: large deformation, diffeomorphic warps with minimal distortions and high registration accuracy.

4

Diffusion Tensor Registration

Contents

4.1	Introduction	139
4.2	Motivation	139
4.3	Diffusion Imaging Modalities	139
4.4	Similarity Metric	140
4.5	Tensor Reorientation	142
4.6	Implementation Within MMORF	145
4.6.1	Cost and Transformation	145
4.6.2	Gradient	147
4.6.3	Hessian	149
4.7	Computational Complexity	150
4.7.1	Translation Complexity	151
4.7.2	Rotation Complexity	151
4.8	Existing Methods	152
4.8.1	DTI-TK	152
4.8.2	DT-REFinD	153
4.8.3	DR-TAMAS	154
4.9	Reorientation Testing	154
4.10	Single Subject Comparison	157
4.10.1	Methods	158
4.10.2	Results and Discussion	164
4.10.3	Conclusion	168

4.1 Introduction

In this chapter I deal with registration of diffusion MRI data, beginning with a motivation for why one might want to do this in the first place. I discuss why I have chosen the diffusion tensor as the type of data to drive the registration, and the why the L^2 similarity is a good choice metric for driving the registration. As the diffusion tensor encodes directional data, I deal with how rotational components of the warp field affect the orientation, and how this is accounted for during registration. I then present in more detail how diffusion tensor registration has been implemented within MMORF with consideration of computational complexity. A simple example of why it is necessary to consider rotation during optimisation is then given. Finally, a comparison to existing diffusion tensor registration tools is presented.

4.2 Motivation

Including some form of diffusion MRI data into MMORF is a logical and desirable thing to do. The white matter regions of the brain are highly complex, but T1 and T2 weighted images are insensitive to these complexities. Due to the regularity of packing in white matter bundles, diffusion of water molecules is more restricted in directions perpendicular to than parallel with the primary orientation of axon bundles (Chenevert et al. 1990). Directional sensitivity to these differences in diffusion is what provides contrast in diffusion weighted imaging. Correct anatomical registration therefore requires correctly aligning these white matter tracts both in terms of position and rotation. As the spatial organisation of the tracts drives the diffusion contrast, aligning the diffusion data as well as possible should correspond to aligning white matter anatomy.

4.3 Diffusion Imaging Modalities

T1 and T2 imaging generally produces only one image per subject, and either or both of these can be used to drive the registration. However, diffusion imaging typically produces tens to hundreds of images per subject, depending on how many directions

and shells are being acquired. While the diffusion weighted images themselves are the richest source of information, simultaneously registering that many images directly would be computationally challenging. There is also considerable redundancy in the information contained in images with similar diffusion weighting, and it would therefore be inefficient to register the diffusion weighted data directly.

One solution to this is to reduce the complexity by using a model which captures the latent information in the data, but contains fewer degrees of freedom. One such model is the diffusion tensor (Basser et al. 1994b; Basser et al. 1994a; Pierpaoli et al. 1996). The diffusion tensor models the underlying source of the diffusion signal as an ellipsoid. In 3 dimensions this corresponds to six degrees of freedom - the three principal (orthogonal) directions of the ellipsoid, and their magnitudes. These directions and magnitudes correspond to the eigenvalues (λ_i) and eigenvectors (V_i) of a 3×3 symmetric positive definite (SPD) tensor matrix. This model therefore encodes information about both the magnitude and the direction of diffusion within a voxel.

More complex models are also available, such as using spherical harmonics as a basis set for defining the diffusion profile (Tournier et al. 2004). While these more complex models are better able to predict the diffusion data, they do not necessarily inform the registration process any more than the tensor model.

With this in mind, I have chosen to use the diffusion tensor image (DTI) as the modality which drives the registration of diffusion data as it provides a good balance between simplicity and information content.

4.4 Similarity Metric

Having selected the diffusion tensor as the driving modality for the registration, there remains the question of how to measure the similarity of two tensors. I have chosen to implement both the Euclidean and Log-Euclidean (Arsigny et al. 2006b; Filiard et al. 2007) L^2 -norm. These distance measures have a number of desirable properties (D. C. Alexander et al. 1999; Peeters et al. 2009), namely that they are:

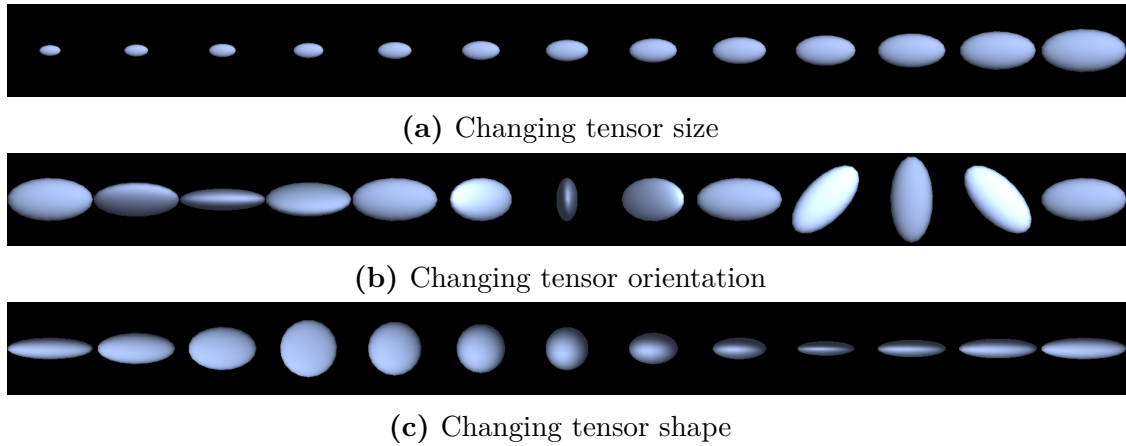


Figure 4.1: Changing tensor properties - (a) shows a tensor with changing size (MD), but with constant shape and orientation. (b) shows a tensor with changing orientation being rotated 180° about the first, then second, then third eigenvectors, with constant shape and size. (c) shows a tensor changing shape from linear, to planar, to spherical, and returning to linear, with constant primary eigenvector orientation and size (MD).

- Sensitive to changes in tensor size
- Sensitive to changes in tensor orientation
- Sensitive to changes in tensor shape
- Robust to noise (Euclidean only)
- Metric distances

Being sensitive to size, orientation and shape of the tensor means that these metrics are sensitive to the expected changes different brain tissues would have on the diffusion tensor. Grey matter and cerebrospinal fluid have very similar orientation and shape, and therefore need to be distinguished by size. Single fibre tracts in close proximity to each other may have similar shape and size, but could be distinguished by orientation. Crossing fibre tracts may have similar size and orientation to single fibre tracts or nearby grey matter, but can be distinguished by their characteristic planar shape. Figure 4.1 shows how the appearance of a tensor changes as size, orientation and shape are varied.

A commonly used alternative to the full tensor is the deviatoric tensor (D. C. Alexander and Gee 2000). This removes any size information, leaving only shape

and orientation. If diffusion analysis is to be constrained to white-matter only regions then this may improve alignment of the principal diffusion direction. As such, deviatoric tensor matching is often weighted by fractional anisotropy. In that case though, I see very little benefit of using the deviatoric tensor over simply the primary eigenvector, as that is by far the dominant driving factor of such a weighted metric. Additionally, the deviatoric tensor is no longer positive definite, and therefore log-Euclidean metrics cannot be used.

The motivation behind using log-Euclidean tensors (Arsigny et al. 2006b; Filiard et al. 2007) is that changes in diffusivity are more meaningfully described by ratios (i.e., double or half the diffusivity) than by absolute differences. As diffusion tensors are SPD, taking the matrix logarithm of the tensors is possible - and the L^2 norm of the difference in logarithmic tensors does indeed scale linearly with the ratio of their diffusivities rather than the difference. One difficulty introduced by the log transform is what to do with background voxels where the diffusion tensor is normally set to zero. A zero diffusion tensor is a reasonable distance from other tensors in the Euclidean sense, but infinitely far from all other tensors in the log-Euclidean sense. Masking out the background is one way of countering this, but it removes a *possibly* helpful edge to guide the registration. The approach I have chosen is to replace the background voxels with the average tensor across all of the non-zero brain voxels. This allows the edge of the brain to be retained, as well as producing a reasonable background intensity.

In practice I have found very little difference in the registration quality when using the log-Euclidean L^2 norm to *drive* the registration, and have therefore used the simpler Euclidean L^2 norm for the results presented here. Log-Euclidean interpolation is, however, always used when applying the *final* resampling of the images.

4.5 Tensor Reorientation

The orientation information in the diffusion tensor can inform the rotational effects of the registration warp in a way that neither scalar modalities nor regularisation can. Scalar modality similarity metrics predominantly ensure that tissue types are

well aligned. The SPRED regularisation used in MMORF ensures that the scalar shape changes which occur during the tissue alignment are reasonable. But, by virtue of being scalar valued, neither of these costs are affected by the rotation of an individual voxel. Sensitivity to orientation, particularly within the white matter, allows the penalisation of deformations which correctly align white matter *tissue*, but incorrectly align the *direction* of the white matter bundles.

In order to implement this, we require a method of determining what the rotation induced by the warp at each point in space is. I return again to the logic introduced for the SPRED penalty in Chapter 3, where the local Jacobian matrix of the warp field describes a linear approximation to the deformation at each voxel. In Section 3.2.1.2 I described how to estimate the lineal components of the deformation using the singular values of the local Jacobian. Here, I will instead describe how the rotational components of the deformation may be similarly estimated.

According to D. C. Alexander et al. (2001) there are three principal approaches to reorientation. The first, is No Reorientation (NR). This is as simple as it sounds, and resampling of a tensor field takes no reorientation into account. While this is clearly inadequate for tensor analysis, NR may not affect registration particularly negatively. For relatively small deformations the correct mapping may still be found despite ignoring reorientation, but it would mean that a potentially important source of information in the diffusion data is unused.

The second is Finite Strain (FS) reorientation. Alexander notes that any linear transformation $\mathbf{A}$ can be decomposed into a rigid rotation $\mathbf{R}$ and a deformation $\mathbf{U}$ such that $\mathbf{A} = \mathbf{U}\mathbf{R}$. The tensor can then be reoriented according to $\mathbf{D}_{rot} = \mathbf{R}\mathbf{D}\mathbf{R}^\top$. $\mathbf{R}$ is straightforward to calculate, and has a closed form solution:

$$\mathbf{R} = (\mathbf{A}\mathbf{A}^\top)^{-\frac{1}{2}} \mathbf{A}. \quad (4.1)$$

FS reorientation has the benefit of simplicity in that it does not depend on the current orientation of the object being transformed. However, this is also its weakness, as in reality the $\mathbf{U}$ matrix would have an effect that varies as a function of the orientation of the object being transformed.

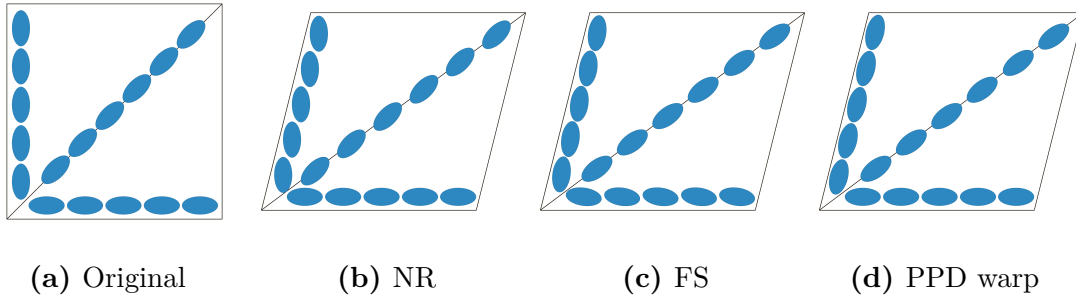


Figure 4.2: Reorientation strategies - Effect when applying a shear to tensors with varying primary diffusion directions. (a) shows the original, undistorted tensors - note how each is parallel to either the vertical, horizontal or diagonal edges of the square. (b) shows the NR approach - note that the tensors are displaced but not rotated. (c) shows the FS approach - note that all tensors are rotated the same amount, and therefore the vertical tensors are under-rotated, and the horizontal tensors are over-rotated. (d) shows the PPD approach - note that all tensors are correctly reorientated by varying amounts, depending on their original orientation, such that they remain parallel to the edges of the trapezoid.

The third approach is known as Preservation of Principal Direction (PPD). This approach seeks to ensure that the primary orientation of the object is correctly rotated. In the tensor, this would be the largest eigenvalue/eigenvector pair (L_1 , V_1). This direction is determined by applying $\mathbf{A}$ to V_1 and rotating the tensor to match this. V_2 is then rotated such that it is perpendicular V_1 , lying in the plane spanned by $\mathbf{A}V_1$ and $\mathbf{A}V_2$. The direction of V_3 is then the direction perpendicular to both V_1 and V_2 . In this way, the reorientation effect of $\mathbf{A}$ is captured in its entirety. Clearly this takes the form of an algorithm rather than a algebraic operation though, and requires an eigen-decomposition of $\mathbf{D}$. This makes PPD impractical for implementation within a registration framework, as the gradients required for optimisation are difficult, if not potentially impossible, to define in a computationally tractable manner. It is, however, the preferred reorientation method when performing the final tensor reorientation following registration.

I have therefore chosen to implement FS reorientation within MMORF as it accounts for a large portion of the rotational component of the warp, but remains tractable to implement.

4.6 Implementation Within MMORF

A key distinguishing feature of how tensor registration has been implemented within MMORF is the choice of which image (i.e., the reference or moving image) to reorient. In practice either option should produce the same result, but they have different practical implications. In MMORF, I have chosen to reorient the reference image as it simplifies calculation of the spatial derivatives of the moving image used during the optimisation.

I will use the following notation for the remainder of this chapter:

$\Omega \subseteq \mathbb{R}^3$ is the image domain

$\Psi_{dti} \subseteq \mathbb{R}^{3 \times 3}$ is the space of SPD 3×3 tensors

$\Psi_{rot} \subseteq \mathbb{R}^{3 \times 3}$ is the space of 3×3 rotation matrices

$\phi : \Omega \rightarrow \Omega$ is the registration transform

$\mathbf{R} : \phi \rightarrow \Psi_{rot}$ is the FS rotation of the transform

$\mathbf{F} : \Omega \rightarrow \Psi_{dti}$ is the reference tensor image

$\mathbf{G} : \Omega \rightarrow \Psi_{dti}$ is the moving tensor image

$\mathbf{F}' = \mathbf{R}\mathbf{F}\mathbf{R}^\top$ is the rotated reference tensor image

$\mathbf{G}' = \mathbf{G} \circ \phi$ is the resampled moving tensor image

4.6.1 Cost and Transformation

Here I will describe the diffusion cost function I have chosen, and how the transformation affects it. I chose the squared L^2 or Frobenius norm.

$$\begin{aligned} \mathcal{C}_{dti} &= \|\mathbf{F}'_{ij} - \mathbf{G}'_{ij}\|_F^2 \\ &= \sum_{i=1}^3 \sum_{j=1}^3 \left(\mathbf{F}'_{ij} - \mathbf{G}'_{ij} \right)^2 \end{aligned} \quad (4.2)$$

I've separated the rotational and translational effects of the warp such that only tensors in the reference image are rotated and only the moving image is translated. This does not affect the registration result, but keeping the two operation separate simplifies the implementation of the calculation of gradients and Hessians.

If one ignores the rotational component, Equation 4.2 is identical to the sum of squares over nine (of which six are unique) scalar images, and can be treated as such. Therefore, if using the NR approach this can easily be implemented as a special case of the scalar MSQ cost function already implemented in MMORF.

The FS rotation of the transform can be derived by noting that the Jacobian matrix $\mathbf{J}$ is the first order linear approximation of the warp. Using Equation 4.1, and substituting $\mathbf{A}$ for $\mathbf{J}$, you then have that:

$$\mathbf{R} = (\mathbf{J}\mathbf{J}^\top)^{-\frac{1}{2}} \mathbf{J}. \quad (4.3)$$

When including FS rotation, the simplest approach is to apply the reorientation after each step in the optimisation, and consider it fixed between steps. In this case, when calculating an update step $\mathbf{R}$ is not considered to be a function of the warp parameters, and therefore $\frac{\partial \mathbf{R}}{\partial \mathbf{w}} = 0$ (see Equation 4.4). While this will not optimise the rotational component of the warp directly, it will at least ensure that warps which would negatively affect the rotational similarity are not favoured.

The most complex alternative is to consider $\frac{\partial \mathbf{R}}{\partial \mathbf{w}}$ to be non-zero. In this case, rotations of the reference tensor which would improve the similarity metric can be optimised for explicitly during the optimisation. This is the approach which I favour, and which I have implemented. Despite the added complexity, the benefits of this method are exemplified in Section 4.9.

4.6.2 Gradient

The gradient of the L^2 cost (Euclidean or log-Euclidean) is derived as follows:

$$\begin{aligned}
\nabla \mathcal{C}_{dti} &= \frac{\partial \mathcal{C}_{dti}}{\partial \mathbf{w}} \\
&= \frac{\partial}{\partial \mathbf{w}} \sum_{i=1}^3 \sum_{j=1}^3 (\mathbf{F}'_{ij} - \mathbf{G}'_{ij})^2 \\
&= 2 \sum_{i=1}^3 \sum_{j=1}^3 (\mathbf{F}'_{ij} - \mathbf{G}'_{ij}) \frac{\partial}{\partial \mathbf{w}} (\mathbf{F}'_{ij} - \mathbf{G}'_{ij}) \\
&= 2 \sum_{i=1}^3 \sum_{j=1}^3 (\mathbf{F}'_{ij} - \mathbf{G}'_{ij}) \left(\sum_{k=1}^3 \sum_{l=1}^3 \frac{\partial \mathbf{F}'_{ij}}{\partial \mathbf{R}_{kl}} \frac{\partial \mathbf{R}_{kl}}{\partial \mathbf{w}} - \frac{\partial \mathbf{G}'_{ij}}{\partial \mathbf{w}} \right) \\
&= 2 \sum_{i=1}^3 \sum_{j=1}^3 (\mathbf{F}'_{ij} - \mathbf{G}'_{ij}) \left(\underbrace{\sum_{k=1}^3 \sum_{l=1}^3 \sum_{m=1}^3 \sum_{n=1}^3 \frac{\partial \mathbf{F}'_{ij}}{\partial \mathbf{R}_{kl}} \frac{\partial \mathbf{R}_{kl}}{\partial \mathbf{J}_{mn}} \frac{\partial \mathbf{J}_{mn}}{\partial \mathbf{w}}}_{\text{rotation}} - \underbrace{\frac{\partial \mathbf{G}'_{ij}}{\partial \mathbf{w}}}_{\text{translation}} \right) \quad (4.4)
\end{aligned}$$

This shows how choosing to apply the rotation to the reference image neatly separates the gradient due to displacement and the gradient due to rotation into separate terms.

I will deal with the $\frac{\partial \mathbf{G}'_{ij}}{\partial \mathbf{w}}$ term first, as it is the simplest. It can be dealt with in a very similar manner to the scalar MSQ cost function. First, one can separate out the term:

$$\begin{aligned}
\nabla \mathcal{C}_{dti-trans} &= -2 \sum_{i=1}^3 \sum_{j=1}^3 (\mathbf{F}'_{ij} - \mathbf{G}'_{ij}) \left(\frac{\partial \mathbf{G}'_{ij}}{\partial \mathbf{w}} \right) \\
&= -2 \sum_{i=1}^3 \sum_{j=1}^3 (\mathbf{F}'_{ij} - \mathbf{G}'_{ij}) \left(\frac{\partial \mathbf{G}'_{ij}}{\partial x/y/z} \frac{\partial x/y/z}{\partial \mathbf{w}} \right) \\
&= -2 \left(\sum_{i=1}^3 \sum_{j=1}^3 (\mathbf{F}'_{ij} - \mathbf{G}'_{ij}) \left(\frac{\partial \mathbf{G}'_{ij}}{\partial x/y/z} \right) \right) \frac{\partial x/y/z}{\partial \mathbf{w}} \quad (4.5)
\end{aligned}$$

Note that $\frac{\partial x/y/z}{\partial \mathbf{w}}$ is independent of i and j , and we can therefore perform the summing over elements of the tensor before multiplying by $\frac{\partial x/y/z}{\partial \mathbf{w}}$.

Now, considering the rotational component of the gradient:

$$\begin{aligned}
\nabla \mathcal{C}_{dti-rot} &= 2 \sum_{i=1}^3 \sum_{j=1}^3 (\mathbf{F}'_{ij} - \mathbf{G}'_{ij}) \left(\sum_{k=1}^3 \sum_{l=1}^3 \sum_{m=1}^3 \sum_{n=1}^3 \frac{\partial \mathbf{F}'_{ij}}{\partial \mathbf{R}_{kl}} \frac{\partial \mathbf{R}_{kl}}{\partial \mathbf{J}_{mn}} \frac{\partial \mathbf{J}_{mn}}{\partial \mathbf{w}} \right) \\
&= 2 \underbrace{\left(\sum_{m=1}^3 \sum_{n=1}^3 \left(\underbrace{\sum_{k=1}^3 \sum_{l=1}^3 \left(\underbrace{\sum_{i=1}^3 \sum_{j=1}^3 (\mathbf{F}'_{ij} - \mathbf{G}'_{ij}) \frac{\partial \mathbf{F}'_{ij}}{\partial \mathbf{R}_{kl}}}_{\frac{\partial \mathcal{C}_{dti-rot}}{\partial \mathbf{R}}} \right) \frac{\partial \mathbf{R}_{kl}}{\partial \mathbf{J}_{mn}}}_{\frac{\partial \mathcal{C}_{dti-rot}}{\partial \mathbf{J}}} \right) \frac{\partial \mathbf{J}_{mn}}{\partial \mathbf{w}}}_{\frac{\partial \mathcal{C}_{dti-rot}}{\partial \mathbf{w}}} \quad (4.6)
\end{aligned}$$

Performing the calculations from the inner brackets outwards provides an algorithm of sorts for how to calculate this gradient. I will now discuss each of the partial derivatives and their calculation separately.

$\frac{\partial \mathbf{F}'_{ij}}{\partial \mathbf{R}_{kl}}$ is fairly straightforward to compute. Recalling that

$$\mathbf{F}' = \mathbf{R} \mathbf{F} \mathbf{R}^\top,$$

therefore:

$$\begin{aligned}
\mathbf{F}'_{ij} &= \mathbf{R}_{j1}(\mathbf{F}_{11}\mathbf{R}_{i1} + \mathbf{F}_{21}\mathbf{R}_{i2} + \mathbf{F}_{31}\mathbf{R}_{i3}) \\
&\quad + \mathbf{R}_{j2}(\mathbf{F}_{12}\mathbf{R}_{i1} + \mathbf{F}_{22}\mathbf{R}_{i2} + \mathbf{F}_{32}\mathbf{R}_{i3}) \\
&\quad + \mathbf{R}_{j3}(\mathbf{F}_{13}\mathbf{R}_{i1} + \mathbf{F}_{23}\mathbf{R}_{i2} + \mathbf{F}_{33}\mathbf{R}_{i3}) \quad (4.7)
\end{aligned}$$

From this $\frac{\partial \mathbf{F}'_{ij}}{\partial \mathbf{R}_{kl}}$ can easily be calculated. The complexity of this calculation can be reduced by noting that:

$$\frac{\partial \mathbf{F}'_{ij}}{\partial \mathbf{R}_{kl}} = 0 \forall k \neq i, j$$

It is straightforward then to calculate the total $\frac{\partial \mathcal{C}_{dti-rot}}{\partial \mathbf{R}}$ term.

Next, I will address $\frac{\partial \mathcal{C}_{dti-rot}}{\partial \mathbf{J}}$, which contains the term $\frac{\partial \mathbf{R}}{\partial \mathbf{J}}$. When considering that:

$$\mathbf{R} = (\mathbf{J} \mathbf{J}^\top)^{-\frac{1}{2}} \mathbf{J},$$

it is not clear how to calculate $\frac{\partial \mathbf{R}}{\partial \mathbf{J}}$, and at first sight it is not obvious whether it has an analytical solution or not. Fortunately, Yeo et al. (2009) were able to derive the following closed form solution:

$$d\mathbf{R} = -\mathbf{R} \left[\mathbf{R}^\top (\text{tr}(\mathbf{S})\mathbf{I} - \mathbf{S})^{-1} \mathbf{R} \sum_j (\mathbf{R}^\top)_j \times (d\mathbf{J}^\top)_j \right]^\oplus \quad (4.8)$$

Where:

$$\mathbf{S} = (\mathbf{J}\mathbf{J}^\top)^{\frac{1}{2}}$$

$\times$ = 3D vector cross-product

$(\cdot)_j$ = select the j^{th} column

$$m^\oplus = \begin{pmatrix} m_1 \\ m_2 \\ m_3 \end{pmatrix}^\oplus \triangleq \begin{pmatrix} 0 & -m_3 & m_2 \\ m_3 & 0 & -m_1 \\ -m_2 & m_1 & 0 \end{pmatrix}$$

This admittedly complex formula is adapted from the field of pose estimation, and relates an infinitesimal change in $\mathbf{R}$ to an infinitesimal change in $\mathbf{J}$. $\frac{\partial \mathbf{R}}{\partial \mathbf{J}_{mn}}$ (i.e., the partial derivative of the *entire* matrix $\mathbf{R}$ with respect to a change of 1 “unit” of the *single element* $\mathbf{J}_{mn}$) can then be calculated by setting $d\mathbf{J} = 0$ apart from element $d\mathbf{J}_{mn} = 1$. Repeating this for all the elements in $\mathbf{J}_{mn}$, we may then calculate the total $\frac{\partial \mathbf{R}}{\partial \mathbf{J}}$, and hence $\frac{\partial \mathcal{C}_{dti-rot}}{\partial \mathbf{J}}$.

The final term to consider is $\frac{\partial \mathbf{J}_{mn}}{\partial \mathbf{w}}$. This is identical to the term in the SPRED penalty (see Equation 3.38), and again is constant with respect to a the chosen B-spline parametrisation. One is therefore in a position to calculate the total $\frac{\partial \mathcal{C}_{dti-rot}}{\partial \mathbf{w}}$. Finally then:

$$\nabla \mathcal{C}_{dti} = \nabla \mathcal{C}_{dti-trans} + \nabla \mathcal{C}_{dti-rot}. \quad (4.9)$$

4.6.3 Hessian

The next step needed to incorporate DTI registration into MMORF is a calculation of the Hessian. From:

$$\begin{aligned} \nabla \mathcal{C}_{dti} &= 2 \sum_{i=1}^3 \sum_{j=1}^3 (\mathbf{F}'_{ij} - \mathbf{G}'_{ij}) \frac{\partial}{\partial \mathbf{w}} (\mathbf{F}'_{ij} - \mathbf{G}'_{ij}) \\ &= 2 \sum_{i=1}^3 \sum_{j=1}^3 \left(\mathbf{F}'_{ij} \frac{\partial \mathbf{F}'_{ij}}{\partial \mathbf{w}} - \mathbf{F}'_{ij} \frac{\partial \mathbf{G}'_{ij}}{\partial \mathbf{w}} - \mathbf{G}'_{ij} \frac{\partial \mathbf{F}'_{ij}}{\partial \mathbf{w}} + \mathbf{G}'_{ij} \frac{\partial \mathbf{G}'_{ij}}{\partial \mathbf{w}} \right) \end{aligned} \quad (4.10)$$

We have:

$$\begin{aligned}
\nabla^2 \mathcal{C}_{dti} &= 2 \sum_{i=1}^3 \sum_{j=1}^3 \left(\frac{\partial \mathbf{F}'_{ij}}{\partial \mathbf{w}} \frac{\partial \mathbf{F}'_{ij}}{\partial \mathbf{w}} + \mathbf{F}'_{ij} \frac{\partial^2 \mathbf{F}'_{ij}}{\partial \mathbf{w}^2} - \frac{\partial \mathbf{F}'_{ij}}{\partial \mathbf{w}} \frac{\partial \mathbf{G}'_{ij}}{\partial \mathbf{w}} - \mathbf{F}'_{ij} \frac{\partial^2 \mathbf{G}'_{ij}}{\partial \mathbf{w}^2} \right. \\
&\quad \left. - \frac{\partial \mathbf{G}'_{ij}}{\partial \mathbf{w}} \frac{\partial \mathbf{F}'_{ij}}{\partial \mathbf{w}} - \mathbf{G}'_{ij} \frac{\partial^2 \mathbf{F}'_{ij}}{\partial \mathbf{w}^2} + \frac{\partial \mathbf{G}'_{ij}}{\partial \mathbf{w}} \frac{\partial \mathbf{G}'_{ij}}{\partial \mathbf{w}} + \mathbf{G}'_{ij} \frac{\partial^2 \mathbf{G}'_{ij}}{\partial \mathbf{w}^2} \right) \\
&= 2 \sum_{i=1}^3 \sum_{j=1}^3 \left(\frac{\partial \mathbf{F}'_{ij}}{\partial \mathbf{w}} \frac{\partial \mathbf{F}'_{ij}}{\partial \mathbf{w}} - 2 \frac{\partial \mathbf{F}'_{ij}}{\partial \mathbf{w}} \frac{\partial \mathbf{G}'_{ij}}{\partial \mathbf{w}} + \frac{\partial \mathbf{G}'_{ij}}{\partial \mathbf{w}} \frac{\partial \mathbf{G}'_{ij}}{\partial \mathbf{w}} \right. \\
&\quad \left. + (\mathbf{F}'_{ij} - \mathbf{G}'_{ij}) \frac{\partial^2 \mathbf{F}'_{ij}}{\partial \mathbf{w}^2} - (\mathbf{F}'_{ij} - \mathbf{G}'_{ij}) \frac{\partial^2 \mathbf{G}'_{ij}}{\partial \mathbf{w}^2} \right) \tag{4.11}
\end{aligned}$$

The Gauss-Newton approximation to $\nabla^2 \mathcal{C}_{dti}$ drops all of the ∂^2 terms, which leaves:

$$\nabla^2 \mathcal{C}_{dti} \approx 2 \sum_{i=1}^3 \sum_{j=1}^3 \left(\frac{\partial \mathbf{F}'_{ij}}{\partial \mathbf{w}} \frac{\partial \mathbf{F}'_{ij}}{\partial \mathbf{w}} - 2 \frac{\partial \mathbf{F}'_{ij}}{\partial \mathbf{w}} \frac{\partial \mathbf{G}'_{ij}}{\partial \mathbf{w}} + \frac{\partial \mathbf{G}'_{ij}}{\partial \mathbf{w}} \frac{\partial \mathbf{G}'_{ij}}{\partial \mathbf{w}} \right) \tag{4.12}$$

This is often justified by noting that at the optimum $\mathbf{F}'_{ij} - \mathbf{G}'_{ij} = 0$, making the approximation exact. I have gone one step further and omitted the $\frac{\partial \mathbf{F}'_{ij}}{\partial \mathbf{w}} \frac{\partial \mathbf{G}'_{ij}}{\partial \mathbf{w}}$ term, leading to the final Hessian:

$$\mathbf{H}_{dti} = 2 \sum_{i=1}^3 \sum_{j=1}^3 \left(\frac{\partial \mathbf{F}'_{ij}}{\partial \mathbf{w}} \frac{\partial \mathbf{F}'_{ij}}{\partial \mathbf{w}} + \frac{\partial \mathbf{G}'_{ij}}{\partial \mathbf{w}} \frac{\partial \mathbf{G}'_{ij}}{\partial \mathbf{w}} \right) \tag{4.13}$$

The justification for this is threefold. Firstly, it can be thought of as a greedy approximation to the Hessian which assumes that the the moving image is not translated during rotation, and that the fixed image is not rotated during translation. Secondly, it decreases the complexity of the calculation considerably. Thirdly, it allows for a simple mechanism for disabling either the rotational or translational component of the cost function if desired.

4.7 Computational Complexity

I will consider the computational complexity of the DTI cost function in comparison to the scalar L^2 cost function and SPRED regularisation. The translational component is best compared to the scalar cost function, as it is *not* a function of the local Jacobian field. The rotational component is best compared to the SPRED regularisation, because it *is* a function of the local Jacobian field.

4.7.1 Translation Complexity

In essence, $\mathcal{C}_{dti}$ is simply the sum of 9 scalar cost functions - each representing an element of the 3×3 diffusion tensor. However, only 6 of the elements are unique, reducing the complexity by a third. Therefore, the translational component of the gradient and Hessian should have an upper limit of $6 \times$ the complexity of an equivalent scalar cost function.

When calculating the gradient sub-component $\nabla \mathcal{C}_{dti-trans}$, note that the sum over the elements of the tensors is performed *before* multiplying by $\frac{\partial x/y/z}{\partial \mathbf{w}}$. Therefore, there is only a $6 \times$ increase in the number of additions as compared to the scalar cost function, and the number of multiplications is unchanged. A similar logic holds for the calculation of the Hessian.

4.7.2 Rotation Complexity

The primary difference between the rotational and translational components of the DTI cost function is that the rotation depends on the local Jacobian field and therefore calculating the gradient and Hessian is more computationally complex.

In this sense, $\mathcal{C}_{dti-rot}$ is actually very similar the SPRED penalty, as they are both functions of the elements of the local Jacobian matrix. From this it can be deduced that once $\frac{\partial \mathcal{C}}{\partial \mathbf{J}_{mn}}$ has been calculated for either $\mathcal{C}_{SPRED}$ or $\mathcal{C}_{dti-rot}$, then $\frac{\partial \mathcal{C}}{\partial \mathbf{J}_{mn}} \frac{\partial \mathbf{J}_{mn}}{\partial \mathbf{w}}$, has the same amount of computational complexity.

In the case of the approximate SPRED penalty, $\frac{\partial \mathcal{C}}{\partial \mathbf{J}_{mn}}$ is only a single scalar value. In contrast, as the rotational component of the DTI penalty is a function of the elements of the rotation matrix, $\frac{\partial \mathcal{C}}{\partial \mathbf{J}_{mn}}$ is then $9 \times$ more complex to calculate.

A final factor impacting the complexity is the $\frac{\partial \mathbf{R}_{kl}}{\partial \mathbf{J}_{mn}}$ term. Most of Equation 4.8 can be calculated through simple matrix algebra. The exception is $\mathbf{S} = (\mathbf{J}\mathbf{J}^\top)^{\frac{1}{2}}$, where the matrix square root operation is non-trivial to implement in closed form. This is the dominant source of extra complexity in the DTI cost function compared to the SPRED penalty, as this operation is calculated on the CPU using an iterative solver from the Armadillo matrix library (Sanderson and Curtin 2016). This is

an area which could be improved in the future, although the effect on the overall performance of MMORF would be modest.

4.8 Existing Methods

I will now discuss some of the key diffusion-based registration tools currently in use.

4.8.1 DTI-TK

DTI-TK (H. Zhang et al. 2006) was one of the first diffusion registration tools to explicitly consider tensor reorientation during registration. Zhang recognised that the FS reorientation strategy could be optimised directly by parametrising the transformation as:

$$\phi_{DTI-TK}(\mathbf{p}, \mathbf{x}) = \mathbf{R}\mathbf{S}\mathbf{x} + \mathbf{T} \quad (4.14)$$

Where:

$$\mathbf{p} = (\mathbf{r}, \mathbf{s}, \mathbf{t})$$

$\mathbf{R} = \mathbf{R}(\mathbf{r}, \mathbf{x})$ is a rotation matrix with 3 Euler angles $\mathbf{r}$

$\mathbf{S} = \mathbf{S}(\mathbf{s}, \mathbf{x})$ is a deformation matrix with 6 independent components $\mathbf{s}$

$\mathbf{T} = \mathbf{T}(\mathbf{t}, \mathbf{x})$ is a translation with 3 components $\mathbf{t}$

In other words, ϕ_{DTI-TK} is a spatially varying affine transform. Specifically, Zhang et al define ϕ_{DTI-TK} to be piecewise constant within each of i equal-sized regions Ω_i covering the domain of the reference image. A natural consequence of this parametrisation is that the affine transforms are discontinuous. Assuming Ω_i are cube-shaped, then each region has 6 discontinuities where it interfaces with neighbouring regions. In order to promote a smoother transform, a term penalising the magnitude of this discontinuity is included in the cost function. Optimisation follows a hierarchical approach. The template is initially subdivided into $n = 4$ regions in each dimension (i.e., $i = n^3 = 64$), with n being doubled at each iteration until $n = 32$. As each region is parametrised by 12 affine components, the total dimension of the parameter space at the highest resolution is 393,216

After convergence, the final warp field is generated by trilinear interpolation of the affine parameters. In order to achieve large deformations, DTI-TK can be run iteratively, with the final deformation being the composition of the warp generated at each iteration.

DTI-TK uses the Euclidean distance between the *deviatoric* tensors of $\mathbf{F}$ and $\mathbf{G}$ (D. C. Alexander et al. 1999), defined as:

$$\mathbf{D}_{deviatoric} = \mathbf{D} - \frac{1}{3} \text{tr}(\mathbf{D})\mathbf{I} \quad (4.15)$$

What this does is subtract the average eigenvalue from each eigenvalue, thereby removing all size information from the tensor. That is, the trace of the deviatoric tensor is always equal to 0. The similarity of deviatoric tensors then is only affected by tensor shape and orientation. This is beneficial when alignment of the white matter is the main priority of the registration, as any uninteresting (or artefactual) differences in tensor size will not affect the result. It will, however, compromise the registration in areas where grey matter and CSF lie adjacent to each other, such as around the ventricles and between gyri, as both tissues have predominantly isotropic diffusion tensors, and differ only in terms of tensor size.

4.8.2 DT-REFinD

DT-REFinD (Yeo et al. 2009) was the first method to utilise Yeo’s derivation of the exact FS gradient. This method extends the diffeomorphic Demons registration algorithm to include tensor alignment and reorientation. The transformation used by the diffeomorphic Demons algorithm is a composition of multiple small deformations, each parametrised by a stationary velocity field. The cost function can either be the Euclidean or log-Euclidean full tensor distance (i.e., not the deviatoric), both of which were shown to perform similarly well. The critical contribution of this work was showing how the rotational effects of the warp can be included into the gradient calculations of the optimisation without having to parametrise them directly, as in DTI-TK.

4.8.3 DR-TAMAS

DR-TAMAS (Irfanoglu et al. 2016) is based on the ANTs registration method, and adapts it to include tensor alignment. The transformation model is therefore the greedy variant of the SyN method developed by Avants et al. (2008), which uses a composition of multiple small displacements of the moving and reference images to a mid-space. It utilises the same FS gradient as DT-REFinD, but employs a weighted combination of the trace and the deviatoric tensor in its similarity metric. Additional scalar images may also be included in the cost function, making this method truly multimodal. By default, the deviatoric tensor is upweighted in the white matter using a smoothed FA mask, with the remainder of the modalities being weighted equally. DR-TAMAS also provides the option to weight all modalities equally, upweight the deviatoric similarity significantly, or down-weight the deviatoric similarity significantly.

4.9 Reorientation Testing

As the inclusion of reorientation in the gradient and Hessian calculations increases the complexity of the optimisation significantly, it is worth providing a concrete demonstration of the benefits of its inclusion, as well as demonstrate that it is performing correctly. I have therefore devised a simple illustrative example for this purpose.

Consider a 3D tensor volume $\mathbf{F}$ where the elements of the tensor are constant across the volume, with values:

$$\mathbf{F} = \begin{bmatrix} \frac{F}{2} & 0 & 0 \\ 0 & F & 0 \\ 0 & 0 & \frac{F}{4} \end{bmatrix}$$

Now consider a second volume $\mathbf{G}_{x/y/z}$ which is identical to $\mathbf{F}$ except that the tensors have been rotated by an angle $\theta_{x/y/z}$ around either the x , y or z axis, such that

$$\mathbf{G}_{x/y/z} = \mathbf{R}_{\theta_{x/y/z}} \mathbf{F} \mathbf{R}_{\theta_{x/y/z}}^{\top},$$

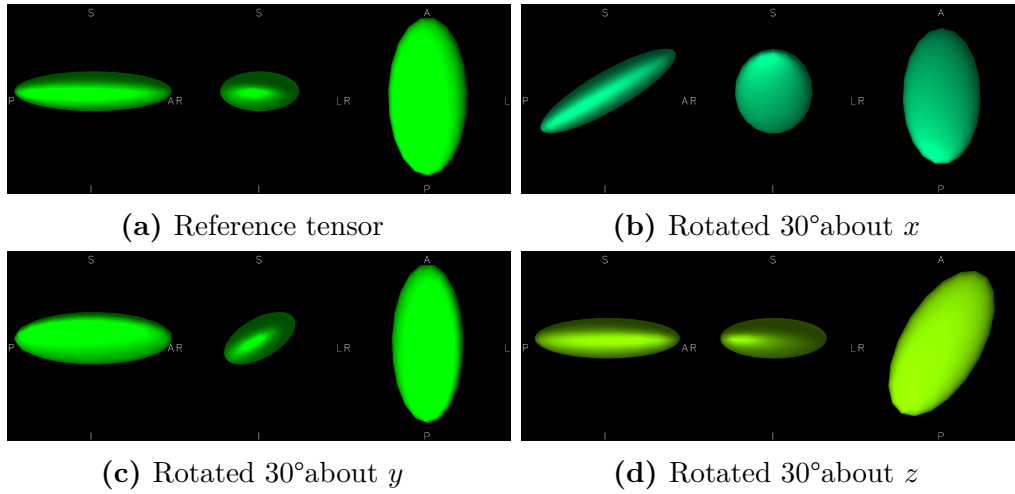


Figure 4.3: Tensors for rotation test - Example tensors for the reference field (a), and moving fields with rotations of 30° about the x (b), y (c) and z (d) axes. Note that each image shows three views (yz , xz and xy planes) of *the same* tensor.

where $\mathbf{R}_{\theta_{x/y/z}}$ is a rotation matrix. Each element of $\mathbf{G}_{x/y/z}$ is then also constant across the volume, and so $\frac{\partial \mathbf{G}_{x/y/z}}{\partial x/y/z} \frac{\partial x/y/z}{\partial \mathbf{w}} = \mathbf{0}$. In other words, the gradient with respect to displacement is zero. Therefore, if tensor reorientation is not considered as part of the optimisation, there would be no gradient to drive the registration, resulting in zero displacement. If tensor reorientation is considered, then $\frac{\partial \mathbf{F}}{\partial \mathbf{R}} \frac{\partial \mathbf{R}}{\partial \mathbf{w}} \neq \mathbf{0}$ and the optimisation will be able to proceed.

Figure 4.3a shows a single tensor belonging to $\mathbf{F}$. Figures 4.3b to 4.3d show tensors from $\mathbf{G}_{x/y/z}$ where $\mathbf{R}_{\theta_{x/y/z}}$ is a 30° rotation about the x , y and z axis respectively. Figure 4.4a shows yz , xz and xy slices through an $11 \times 11 \times 11$ tensor field $\mathbf{F}$ corresponding to Figure 4.3b. Figures 4.4b to 4.4d show the same slices for the tensor fields $\mathbf{G}_x$, $\mathbf{G}_y$ and $\mathbf{G}_z$ respectively.

Each of $\mathbf{G}_{x/y/z}$ was registered to $\mathbf{F}$ using MMORF, and the resulting warps are shown in Figure 4.5. Note how at the centre of the warp the field is an almost perfect rotation. Towards the edges the rotation effect gets slightly “squared-off”. This is a consequence of the boundary conditions forcing the corners of the reference and moving images to remain aligned.

As desired, the warps are able to rotate the tensors back into alignment with the reference warp field. This example neatly illustrates the type of scenario

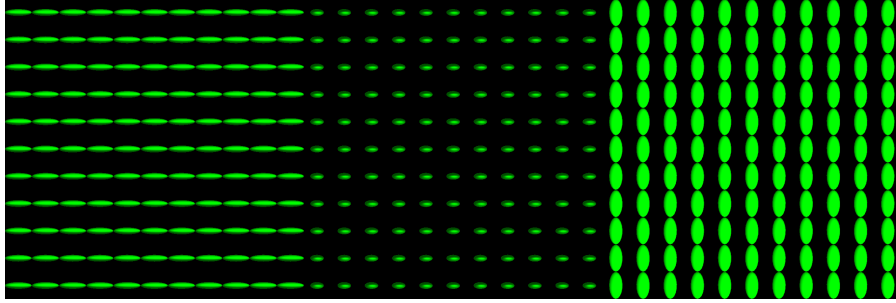
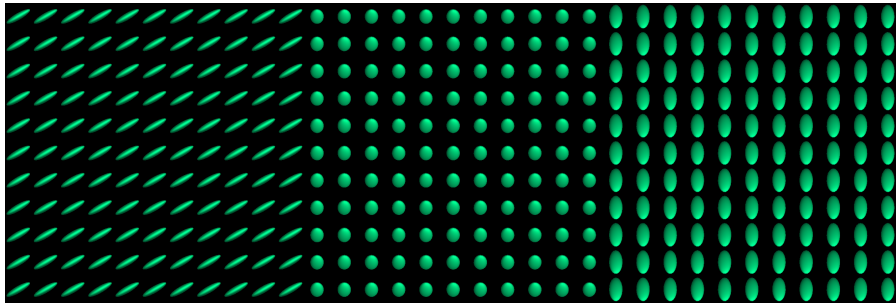
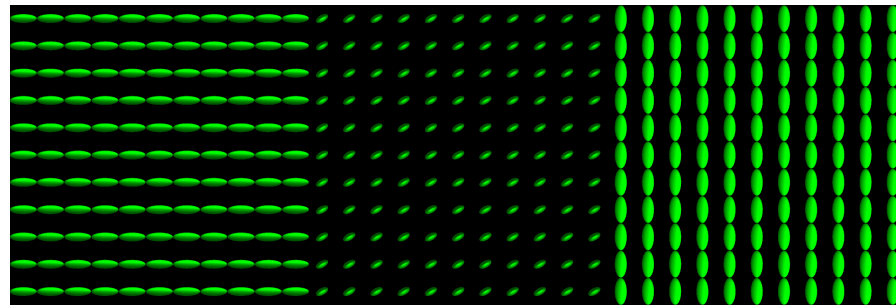
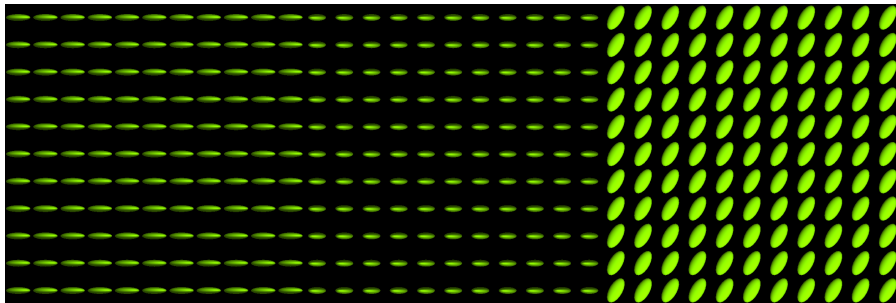
(a) Reference tensor field $\mathbf{F}$ (b) Moving tensor field $\mathbf{G}_x$ (c) Moving tensor field $\mathbf{G}_y$ (d) Moving tensor field $\mathbf{G}_z$

Figure 4.4: Reference and moving tensor fields - (a) shows yz , xz and xy slices through the reference tensor field $\mathbf{F}$, which is constant and has all eigenvectors aligned with the principal axes of the coordinate system. (b) to (d) show slices through moving images $\mathbf{G}_{x/y/z}$, which are versions of $\mathbf{F}$ rotated about the x , y and z axes respectively.

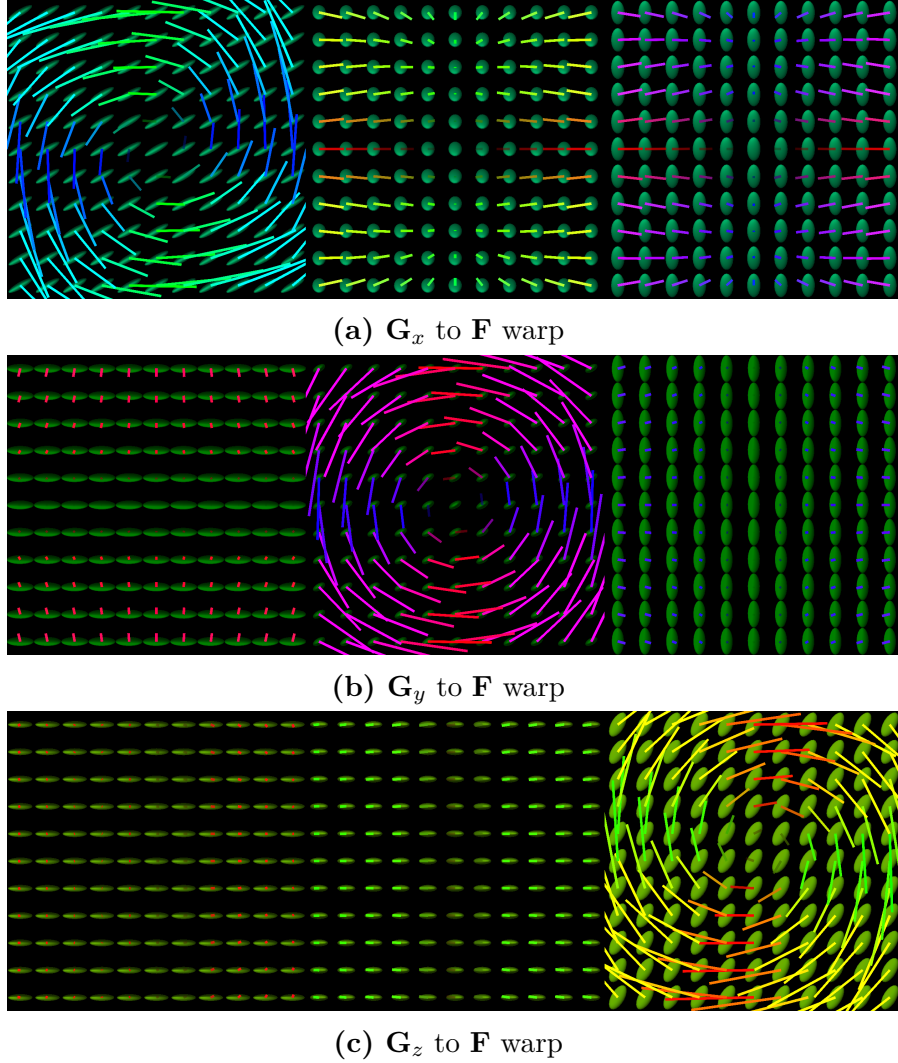


Figure 4.5: Warp fields - (a) to (c) show yz , xz and xy slices through the warp fields generated when registering $\mathbf{G}_x$, $\mathbf{G}_y$ and $\mathbf{G}_z$ to $\mathbf{F}$ respectively. The vectors representing the warps are colour coded. Red corresponds to the x direction, green to the y direction and blue to the z direction. The rotational effect of each warp field about a single axis is clearly evident in each case.

which can only be solved by including the rotational effect of the warp in the gradient calculation.

4.10 Single Subject Comparison

In this section I present a comparison between MMORF, DTI-TK and DR-TAMAS when performing inter-subject diffusion tensor registration. This comparison therefore includes:

- DTI-TK: The original, and widely used, diffusion tensor registration method incorporating tensor reorientation - based on the composition of a number of piecewise affine transformations. Parametrises rotations directly.
- DR-TAMAS: A relatively new diffeomorphic diffusion registration tool - based on the ANTs symmetric normalisation transform. Uses Yeo’s analytic finite strain derivative during gradient decent optimisation.
- MMORF: My proposed unified multi-modal framework - based on a B-Spline parametrised free-form deformation with the SPRED penalty ensuring diffeomorphism. Also uses Yeo’s analytic finite strain derivative, but in a Gauss-Newton optimisation framework.

The comparison presented here is less expansive than that of the SPRED penalty in Chapter 3, as I am not attempting to characterise a previously unexplored metric. In Chapter 6 I perform a far more interesting and detailed validation of the value added by adding DTI images as an additional modality within a multimodal registration optimisation. I am also not making any firm claims about where MMORF “ranks” among diffusion registration tools. Rather, I wish to show that the design decisions I have taken when developing the DTI registration component of MMORF lead to results which are of broadly comparable quality to those of more established methods. In essence, this comparison acts as a sanity check for ensuring that Chapter 6’s unimodal versus multimodal MMORF comparisons can be meaningfully analysed and interpreted.

4.10.1 Methods

Each method was used to register the DTI volumes of a randomly selected pair of subjects from the Human Connectome Project (Van Essen et al. 2012; Van Essen et al. 2013; Glasser et al. 2013). The volumes were first affine registered, followed by nonlinear registration.

4.10.1.1 Data Preparation

The original DTI volumes are in the format used by FSL, and have a resolution of 1.25 mm isotropic. This format stores only the upper triangular components of the tensor in a 4D NIfTI volume, in the order $[xx, xy, xz, yy, yz, zz]$. Additionally, the eigen-decomposition of the tensor is stored as three 3D NIfTI eigenvalue volumes (L1, L2 and L3) and three 4D NIfTI eigenvector volumes (V1, V2 and V3). DTI-TK and DR-TAMAS each use a different format to FSL and therefore the original DTI volumes needed to be converted into the correct format for each tool. DTI-TK provides a `fsl_to_dtitk` utility to convert from the FSL eigensystem representation to DTI-TK's tensor format, as well as the `TVEigenSystem` utility for converting back. DR-TAMAS provides two utilities for converting to and from the DTI-TK tensor format, namely `ConvertDTITIKTensorToTORTOISETensor` and `ConvertTORTOISETensorToDTITIKTensor`. These utilities were combined such that the registration could be carried out using the native DTI format for each tool, and the analysis of the results could be carried out in the FSL eigensystem format.

DTI-TK requires that the voxel dimensions of the reference volume be an integer power of 2, which necessitates an additional preprocessing step. One option for achieving this is to simply zero-pad the volume in all dimensions. However, as the original reference volume had voxel dimensions of $145 \times 174 \times 145$, this would require padding out to dimensions of $256 \times 256 \times 256$ which becomes excessive. Furthermore, as DTI-TK basis its resolution pyramid sizes on proportions of the original volume (i.e. $1/32$ of the original dimensions, then $1/16$ of the original dimensions, etc.) this would detrimentally reduce the warp resolution. Instead I used the DTI-TK tool `TVResample` to resample the reference volume to voxel dimensions of $256 \times 256 \times 128$ at a voxel resolution of $0.708 \text{ mm} \times 0.8496 \text{ mm} \times 1.25 \text{ mm}$. This preserves the size of the volume (measured in mm) in the x and y dimensions, and crops the background of the z dimension down slightly without removing any of the brain.

4.10.1.2 Affine Registration

Affine registration was performed separately for each tool being tested.

The affine stage of DTI-TK is based on matching the tensor images themselves, including reorientation. Following the recommended DTI-TK workflow, rigid body registration was performed first using the `dti_rigid_reg` script. Affine registration was then performed using the `dti_affine_reg` script, with the output of the rigid step acting as an initialisation. The resampled moving image was then used as the input to the nonlinear step, as DTI-TK does not allow an affine transform to be provided for initialising the optimisation. However, note that the final resampling was done in a single step from the original moving image rather than resampling the affine transformed image.

DR-TAMAS has the option to simply include affine registration as part of the nonlinear registration call, and this is the approach I have utilised. This is therefore also based on the tensor images themselves. The nonlinear step then has the option to save out the combined affine and nonlinear deformation as a single warp volume for resampling purposes.

MMORF does not include its own affine registration option, as it is designed to fit into the FSL suite of tools which already includes linear registration tool FLIRT. As FLIRT does not support DTI images, a scalar proxy image is required for this step. Affine registration was therefore performed between the $b = 0$ volumes, and the resulting affine matrix was used to initialise the nonlinear step. The combined deformation was calculated using the FSL `convert_xfm` utility.

4.10.1.3 Nonlinear Registration Parameters

The registration parameters for DTI-TK, DR-TAMAS and MMORF were selected as described below. For the most part these values were kept at the recommended defaults, except where changes improved results.

For DTI-TK, the number of iterations to use was set to the recommended maximum of 6. The images are subdivided into 4^3 , 8^3 , 8^3 , 16^3 , 16^3 and 32^3 piecewise affine subregions per iteration, with the highest resolution having an approximate

spacing of $5.7\text{ mm} \times 6.8\text{ mm} \times 5.0\text{ mm}$. Note that because the warps generated at each iteration are composed with the previous warp, the accuracy of the final warp can be higher than this value. Regularisation followed a decreasing pattern with each iteration, and was set to DTI-TK's slightly weaker setting (optimised for subject-to-template registration) in order to more closely match the regularisation strength of MMORF and DR-TAMAS. The default Jacobian determinant limit of 0.02 to 50 was left unchanged, as well as the relative tolerance for convergence of 0.002. DTI-TK solves for the rotational components of the warp field directly, which is equivalent to using FS reorientation during optimisation.

DR-TAMAS was run using parameters based on the supplied `dtireg_batch_register_to_target` script at detail level 6, with a few modifications. Both the tensor trace and deviatoric were used as metrics, and combined according to the FA weighting scheme rather than weighting them equally - the deviatoric being favoured in areas of high FA, and the trace in areas with low FA. The ANTs SyN transform was used, with the default gradient step of 0.3, update field smoothing of 2 and final field smoothing of 0.1 - which is a moderately aggressive deformation parametrisation. Four levels of subsampling were used, with shrink factors of 6, 4, 2 and 1 corresponding to a warp resolution of 1.25 mm at the highest resolution. The images were smoothed with Gaussian kernels of $\sigma = 1.0, 1.0, 0.5$ and 0.05 voxels isotropic at each subsampling level respectively. FS reorientation was used during optimisation.

MMORF was run with parameters which fall between DTI-TK and DR-TAMAS in terms of warp resolution. A warp resolution pyramid of four steps was used, with knot spacings of 16 mm, 8 mm, 4 mm and 2 mm isotropic. Smoothing at each iteration was performed with a Gaussian kernel of FWHM 4.0, 2.0, 1.0 and 0.5 mm isotropic, which correspond to σ s of approximately 1.7, 0.8, 0.4 and 0.2 mm respectively. Regularisation was decreased at each resolution, and set to 0.37, 0.31, 0.26 and 0.22 (arbitrary units), which correspond to values optimised for T1 registration at similar warp resolutions. MMORF uses FS reorientation during optimisation.

4.10.1.4 Resampling

Each method was resampled using its own implementation. In all cases the original moving image was resampled only once to include both affine and nonlinear deformations. The original reference image (i.e., not the version resampled for DTI-TK) was chosen as the destination space, in order for the methods to be comparable. Despite all methods using FS reorientation during registration, final resampling was performed using PPD for all methods as it more accurately represents the true effect of the deformation on the tensors.

4.10.1.5 Evaluation Strategy

Both qualitative and quantitative aspects of the registration were evaluated for each method. The resampled moving images were compared visually using RGB directionally encoded colour (DEC) maps of the primary eigenvector (V1), where the intensity is weighted by FA. This representation is useful for visually assessing both FA alignment, and directional consistency. Red is used to encode the x direction, green for the y direction and blue for the z direction. Jacobian determinant maps were calculated for each method, and visually compared to assess the nature of the deformations for each method, as well as comparing the 5th to 95th percentile ranges within two regions of interest - the whole brain, and areas where $FA \geq 0.2$.

Three quantitative measures of tensor alignment are used. The first measure is the scalar product of the primary eigenvectors (V1), weighted by the linear shape coefficients (CL) (A. L. Alexander et al. 2000), which I will refer to as CLV1. CL is defined as:

$$CL = \frac{L1 - L2}{L1 + L2 + L3} \quad (4.16)$$

and represents how linear the tensor shape is. For linear tensors, the most informative eigenvector is V1. CLV1 is then defined as:

$$CLV1 = \frac{2CL_r CL_m V1_r^T V1_m}{CL_r + CL_m} \quad (4.17)$$

where the subscripts r and m represent the reference and resampled moving image respectively. CLV1 is therefore a neat representation of how well V1 is aligned in areas where V1 is a meaningful measure.

The second measure is the scalar product of the tertiary eigenvector (V3), weighted by the planar shape coefficient (CP) (A. L. Alexander et al. 2000), which I will refer to as CPV3. CP is defined as:

$$\text{CP} = \frac{2(\text{L2} - \text{L3})}{\text{L1} + \text{L2} + \text{L3}} \quad (4.18)$$

and represents how planar the tensor shape is. For planar tensors, the most informative eigenvector is V3. CPV3 is then defined as:

$$\text{CPV3} = \frac{2\text{CP}_r \text{CP}_m \text{V3}_r^\top \text{V3}_m}{\text{CP}_r + \text{CP}_m} \quad (4.19)$$

where the subscripts r and m represent the reference and resampled moving image respectively. In a similar manner to CLV1, CPV3 then provides a neat measure of how well V3 is aligned in areas where V3 is a meaningful measure.

The third measure is the overlap of eigenvalue-eigenvector pairs (OVL), which is a measure of overall tensor similarity. OVL is defined as:

$$\text{OVL} = \frac{\sum_{i=1}^3 \text{Li}_r \text{Li}_m (\text{Vi}_r^\top \text{Vi}_m)^2}{\sum_{i=1}^3 \text{Li}_r \text{Li}_m} \quad (4.20)$$

where the subscripts r and m represent the reference and resampled moving image respectively.

For each registration tool, CLV1, CPV3 and OVL were calculated over two regions of interest (ROIs). The first is simply a mask of the whole brain. This mask was generated using the mean diffusivity map (MD) thresholded to zero below the robust lower range (as calculate with `fs1stats`) and then eroded by one voxel. This ensured that noisy voxels outside of the brain were not included in the analysis. The second was the intersection of a mask of regions where $\text{FA} \geq 0.2$ and the MD mask. This is a common method of producing a basic segmentation of white matter, and aims to restrict the analysis to those regions where directional measures are more meaningful. Both masks are defined based on the reference volume.

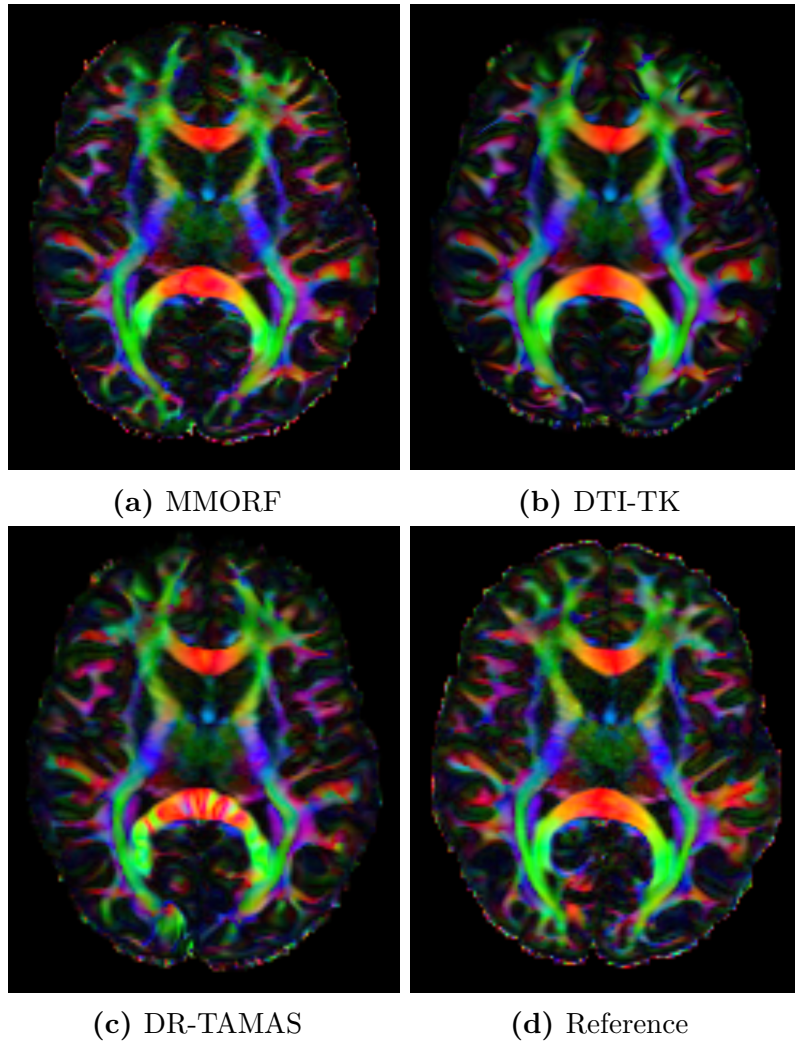


Figure 4.6: DEC images coloured according to the primary eigenvector, with intensity weighted by FA. (a) to (c) show the resampled moving images aligned using MMORF, DTI-TK and DR-TAMAS respectively. (d) shows the reference image. MMORF and DTI-TK have produced the best results, which are very comparable with each other. DR-TAMAS has aligned the FA well, but has caused incorrect tensor orientations in the splenium of the corpus callosum.

4.10.2 Results and Discussion

Figure 4.6 shows an axial slice of the DEC image for each method's transformed moving image, and the corresponding slice from the reference image. All three methods have produced results with a high degree of spatial similarity compared to the reference.

The results for MMORF and DTI-TK are remarkably similar, except for the posterior of the brain where DTI-TK has not quite matched the border of the brain

correctly. However, due to the highly variable folding patterns in this region, this does not mean that the alignment of the high FA values are particularly worse than MMORF. It does mean that if the warps were used to analyse regions of low FA, such as the grey matter, then DTI-TK would be at a disadvantage. In high FA regions the direction of V1 and the FA intensity match the reference image very well, indicating that the registration has been successful.

DR-TAMAS has produced results which are very similar to both other methods in most areas, but with one notable difference. The splenium of the corpus callosum shows an undesirable amount of fluctuation in the directionality of V1. In this region the white matter tissue is correctly matched to the template, but its orientation is inconsistent. The fact that the rest of the DEC map does not suffer from this issue indicates that this is a true feature of the warp in that area. Part of the issue may come from the combination of the more aggressive SyN transformation leading to a larger difference between PPD and FS reorientation results. Figure 4.7 shows the DR-TAMAS result using PPD and FS reorientation. It can be seen that using FS for the final reorientation, as was done during the optimisation, reduces this effect, but does not alleviate it entirely. Despite having a warp resolution that is almost as high as DR-TAMAS, MMORF does not suffer this same effect. A potential reason for this is that the SPRED regularisation employed by MMORF penalises warps which deviate too far from being locally rigid - thereby minimising precisely the characteristic most likely to cause difference between PPD and FS results.

The Jacobian determinant maps of the same axial slice as in Figure 4.6 are shown in Figure 4.8 for each of the three methods. All methods display broadly the same pattern of expansion and contraction as you might expect. MMORF appears to yield less extreme values than DTI-TK, but with more spatial variability. DR-TAMAS has considerably more areas of large expansion and contraction than the other two methods. DTI-TK has the smoothest appearance - most likely due to the lower resolution of the parametrisation. Looking at the splenium of the corpus callosum, MMORF and DTI-TK have both contracted this region in a largely uniform manner - indicating that this structure was larger in the reference image than in the moving

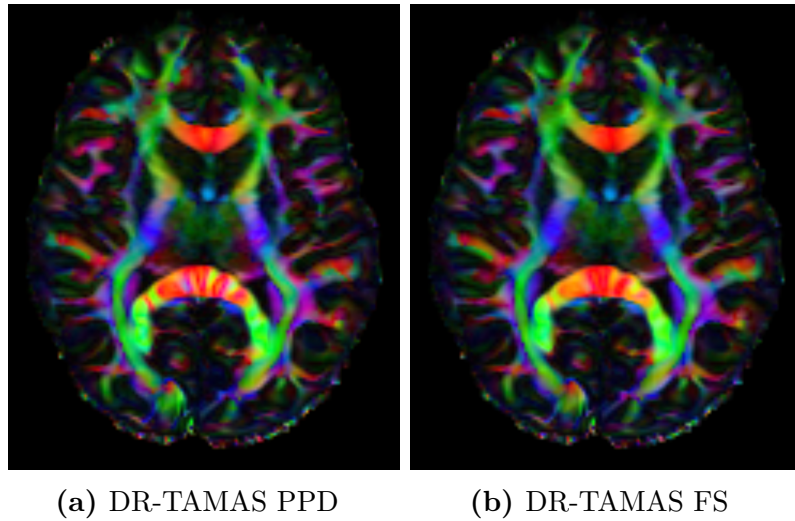


Figure 4.7: Difference between PPD and FS reorientation when using DR-TAMAS. The incorrect orientations in the splenium of the corpus callosum are reduced when using FS reorientation - the method used by DR-TAMAS during optimisation - suggesting that the difference between the reorientation methods is at least a partial cause of the error. The remaining inconsistencies in the orientations appear most prominent at the anterior border of the splenium, visible as “bumps”. As this is a point where there is a sharp transition in the FA cost function weighting used by DR-TAMAS, perhaps that too has contributed to this phenomenon.

image. DR-TAMAS has deformed the splenium much less uniformly, which is an obvious cause of the orientation inconsistencies observed in the DEC maps.

The distribution of log-Jacobian determinants within the brain, and within a mask of regions with $FA \geq 0.2$ are shown in Figure 4.9. In both ROIs, DR-TAMAS is deforming the volume considerably more than either MMORF or DTI-TK. In the whole brain distribution (Figure 4.9a), DTI-TK has a stronger central tendency than MMORF, but with slightly heavier tails. This corresponds to the observations seen in the log-Jacobian determinant maps, where DTI-TK appeared less variable but with more extreme values than MMORF. DTI-TK would be expected to deform the image most strongly within the FA mask, as the deviatoric tensor metric is far more sensitive in this region. Figure 4.9b supports this, as within this ROI the Jacobian determinants of MMORF and DTI-TK appear very similar. This suggests that the difference between their whole brain distributions is the fact that MMORF is also aligning grey matter regions due to using the full tensor rather than the deviatoric.

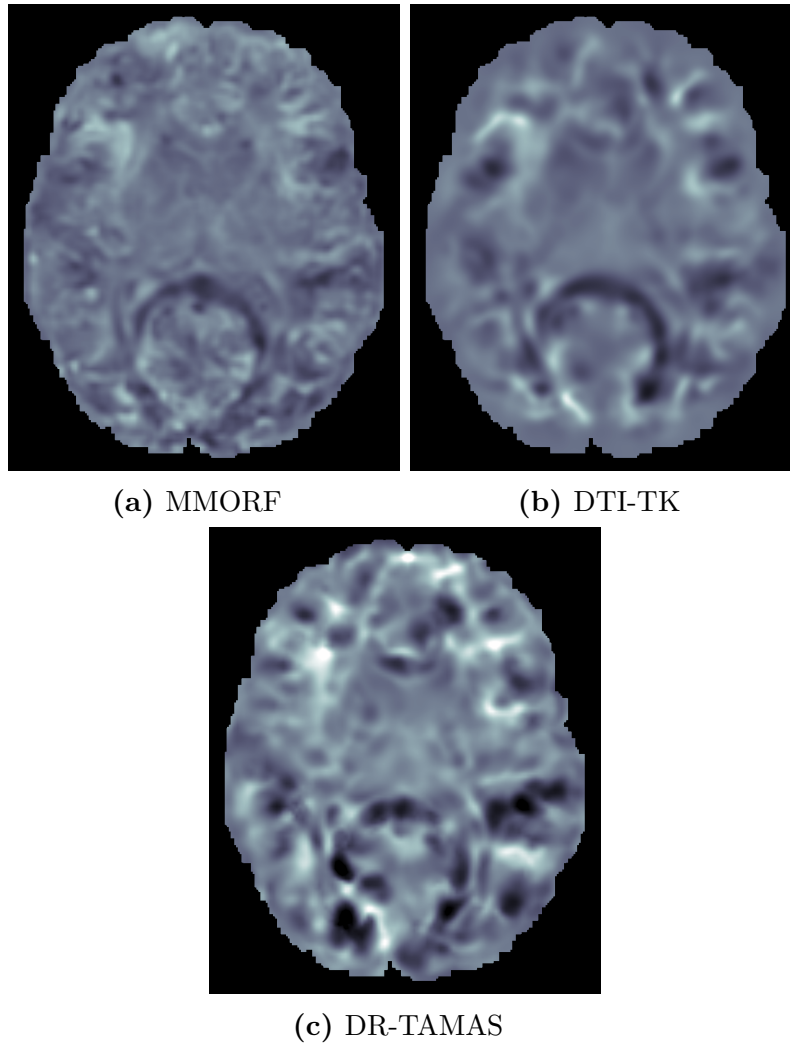


Figure 4.8: log-Jacobian determinant maps for MMORF, DTI-TK and DR-TAMAS. All images are scaled to the same intensity range. DR-TAMAS displays more aggressive expansion and contraction than the other methods. DTI-TK appears smoother than MMORF, but has larger amounts of peak expansion and contraction.

Table 4.1 provides a quantitative evaluation of the qualitative characteristics observed thus far. MMORF achieves the highest CLV1 and CPV3 values within both the whole brain and the FA mask, indicating that it is capable of aligning those areas where directionality of the diffusion tensor is most meaningful. DR-TAMAS and DTI-TK perform very similarly in their ability to align the primary eigenvector, as shown by CLV1, with DTI-TK outperforming DR-TAMAS in meaningfully aligning the tertiary eigenvector, as shown by CPV3. DTI-TK performs best in the overall tensor shape matching OVL metric within both ROIs, followed by

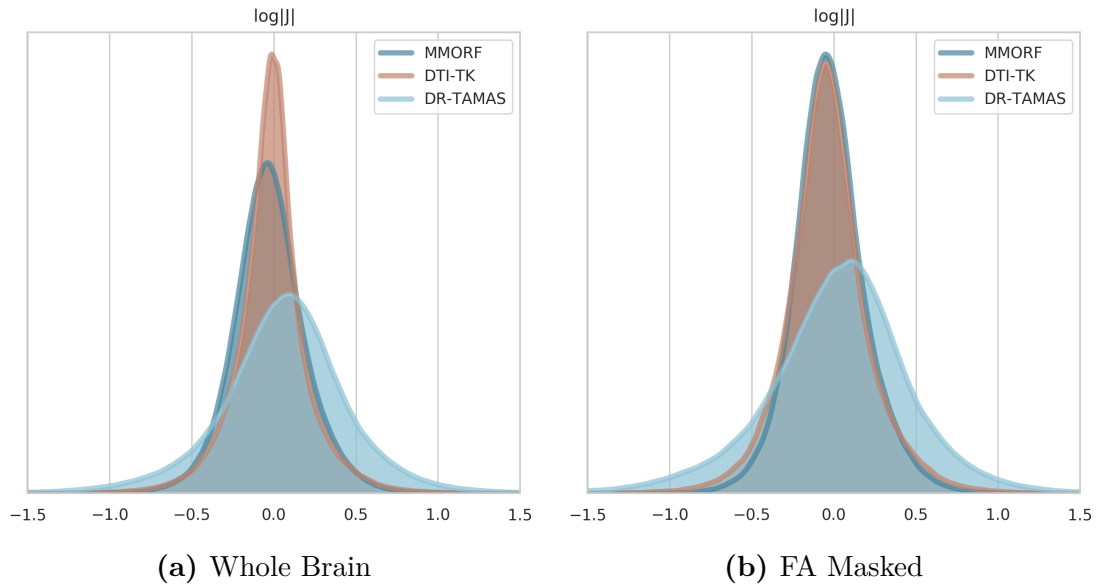


Figure 4.9: log-Jacobian determinant distribution within two ROIs - the whole brain and areas where $FA \geq 0.2$. Across the whole brain DTI-TK has a greater central tendency than MMORF, but has slightly heavier tails. Within the FA mask, their distributions are very similar, with DTI-TK again having slightly heavier tails. In both instances DR-TAMAS displays a greater degree of volumetric deformation, as evidenced by a much wider distribution.

MMORF and then DR-TAMAS. This agrees with the observed level of image similarity in the DEC maps.

In terms of the log-Jacobian determinant percentiles, MMORF and DTI-TK have almost identical 5th to 95th ranges across the entire brain, with MMORF having a narrower range within the FA mask. DR-TAMAS has a considerably wider range in both cases. Again, this agrees entirely with the earlier observations.

Taken together, these results show that MMORF is capable of aligning DTI volumes with high levels of accuracy, comparable to that of well established tools. This is achieved while keeping volumetric distortion well under control.

4.10.3 Conclusion

In this chapter I have presented my implementation of diffusion tensor image registration within the MMORF framework. I chose to implement both the Euclidean and Log-Euclidean L^2 tensor distance as similarity metrics due to their simultaneous sensitivity to tensor size, shape, and orientation - and their relative insensitivity

ROI	Whole Brain			FA Masked		
Method	MMORF	DTI-TK	DR-TAMAS	MMORF	DTI-TK	DR-TAMAS
CLV1	0.0914	0.0895	0.0895	0.1573	0.1553	0.1556
CPV3	0.0904	0.0876	0.0863	0.1440	0.1405	0.1370
OVL	0.5972	0.5976	0.5868	0.7284	0.7334	0.7132
5 th -95 th log J	0.736	0.735	1.293	0.738	0.856	1.449

Table 4.1: Average tensor similarity measures after inter-subject registration and 5th to 95th percentile range of the log-Jacobian determinant of the deformation. Three similarity measures are compared here. 1) The primary Eigenvector dot product, weighted by the linear tensor coefficient (CLV1). 2) The tertiary Eigenvector dot product, weighted by the planar tensor coefficient (CLV3). 3) The overlap of eigenvalue-eigenvector pairs (OVL). They are calculated across the entire brain, as well as within a mask of those regions where $FA \geq 0.2$. In the first two measures MMORF performs best, followed by DTI-TK and then DR-TAMAS. In the third measure DTI-TK performs best, followed by MMORF and then DR-TAMAS. For the log-Jacobian percentile range, a smaller value is usually considered better provided that the images are equally well registered. Across the whole brain DTI-TK appears to perform best followed very closely by MMORF - however the fact that MMORF is more sensitive to grey matter misalignment and therefore is more likely to generate larger deformations outside of the white matter may account for this. Within the FA mask MMORF performs best, followed by DTI-TK. Note that the MMORF results are very consistent between the two ROIs. Within both ROIs, DR-TAMAS produces by far the widest range of Jacobian determinants.

to noise. Tensor reorientation has been included in the optimisation using the finite strain approximation of the rotational effect of the warp field - with analytic forms of both the gradient and Hessian due to reorientation implemented using the method proposed by Yeo. The advantage of including reorientation in the optimisation gradient itself is demonstrated using an example which cannot be aligned without it - even if reorientation is applied between optimisation update steps. Finally I have compared MMORF to DTI-TK and DR-TAMAS in inter-subject registration, and found it matched or bettered both of them in terms of registration accuracy and volumetric distortion.

From this I conclude that DTI registration as implemented in MMORF can be considered to be a fully working method. In combination with the scalar registration capability and desirable characteristics of the SPRED regularisation, as demonstrated in Chapter 3, MMORF can therefore be used to characterise how multimodal registration results differ from either unimodal scalar or tensor

registration. This will form the basis of the chapters which follow.

5

Multimodal Template Creation

Contents

5.1	Introduction	172
5.2	Motivation	172
5.3	An Unbiased Average	172
5.4	Existing Multimodal Templates	173
5.4.1	ICBM (Human)	174
5.4.2	SRI24 (Human)	175
5.4.3	VALiDATe29 (Squirrel Monkey)	176
5.4.4	TORTOISE HCP (Human)	177
5.5	Methodology	178
5.5.1	Initial Affine Template	179
5.5.2	Nonlinear Template Iterations	180
5.6	Lemur and Macaque Template	182
5.6.1	Ethical Approval	182
5.6.2	Methods	183
5.6.3	Results and Discussion	185
5.6.4	Conclusion	190
5.7	UK Biobank Template	190
5.7.1	Data Acquisition and Preprocessing	191
5.7.2	Methods	192
5.7.3	Results and Discussion	194
5.7.4	Conclusion	194

5.1 Introduction

In previous chapters I have described how MMORF was developed with the aim of aligning multiple modalities simultaneously. However, it is rarely the case that registration is performed pairwise between individual subjects - with registration to a template or atlas being by far the more common procedure followed. Of course, a prerequisite for multimodal registration to a template is the *existence* of a multimodal template. In this chapter I will describe the method I have used for creating such multimodal templates, as well as examples of templates that I have already created for the ring-tailed lemur, rhesus macaque and human.

5.2 Motivation

It is possible to create a multimodal template without a multimodal registration tool by choosing a single modality to use in the registration, and transforming all other modalities through this modality's warp. A comprehensive example of such a procedure is presented in Gupta et al. (2016). This would generally be done using a T1 image, as this is the workhorse modality for anatomical alignment, and most studies would include this. If a sensible templating approach is followed then this should result in a template with good between tissue-type sharpness - as this is the main type of contrast visible in a T1 image. However, any modalities with contrast other than between different tissue-types are not guaranteed to be accurately represented in the resulting template, as that information was not leveraged to drive the template construction. Instead, the within tissue-type deformations are determined only by nearby tissue boundaries and regularisation.

5.3 An Unbiased Average

One tends to think of a template as being a form of average representation of what we are studying - thereby providing a common space in which to analyse and/or report results. But a common space does not necessarily need to be an average. In the neuroimaging context it could be a single subject's scan which is chosen as the

reference, and all other subjects registered to it. If the true one-to-one mapping between subjects was known, the choice of template space would be arbitrary. However, image registration is an ill-posed problem with no unequivocally defined ground truth. Therefore, in real life scenarios, such a mapping is unknowable. This is evidenced by the need for regularisation, which acts to constrain mappings to those which are deemed most plausible.

One would like the most probable template to subject deformation to be the identity - meaning that on average any point in the template does not move when being registered to the population. This means that the total regularisation penalty when registering the template to the entire population would be minimised. I will refer a template which has zero average deformation as the *shape average* of the population. Therefore, a single subject which is deformed such that the average deformation from that point to any other subject is zero would be a shape average template.

As MMORF uses image intensities to drive the registration, a template which has the correct shape average, but the *appearance* of a single subject (i.e., a level of contrast that is not identical to that of the population of scans) may not on average be deformed correctly to the population. It is therefore desirable to have the voxel intensity at each point in a template be set such that the average of the chosen similarity penalty between the template and the same anatomical location in all subjects is minimised. I will refer to a template which has the minimum average similarity penalty as the *appearance average*.

Hence, an unbiased template is one which represents the average of the population both in terms of shape and appearance. This concept was introduced by Joshi et al. (2004) in the context of template construction using velocity based diffeomorphic warps.

5.4 Existing Multimodal Templates

In this section I will discuss some of the multimodal atlas/template options currently available. I will only consider those templates which contain both structural (T1,

T2 etc.) and diffusion (FA, MD, DTI, etc) modalities. Where templates contain modalities with directional information (e.g., V1, DTI) I will display them using RGB directionally encoded colour (DEC) maps of the primary eigenvector (V1), where the intensity is weighted by FA. This representation is useful for visually assessing both FA sharpness, and directionality. Red is used to encode the x direction, green for the y direction and blue for the z direction.

5.4.1 ICBM (Human)

The International Consortium of Brain Mapping (ICBM) family of templates (Figure 5.1) were developed at the Montreal Neuroimaging Institute, and so the space in which they exist is known as MNI-space.

The most well-known, and utilised ICBM templates are the 152 subject ICBM-152 T1, T2 and PD atlas (Fonov et al. 2011) and the 81 subject ICBM-81 white matter atlas (Oishi et al. 2008; Mori et al. 2008). Although the ICBM-152 and ICBM-81 templates are both in MNI-space, they were not derived from the same population of subjects. Structures within each template are therefore not exactly spatially aligned. This means that these templates may be used individually, but joint statistics on both scalar and tensor modalities cannot be performed. Additionally, any potential improvements in registration accuracy from using multimodal registration cannot be taken advantage of. This is the simplest example of a multimodal atlas.

The ICBM-152 template used an iterative, nonlinear registration algorithm similar to the methodology I present in Section 5.5, including spatial unbiasing. However, as no unbiasing was applied at the affine initialisation stage, the ICBM-152 template has a characteristically stretched appearance in the inferior-superior direction. This is a side-effect of choosing to maintain a level of compatibility with the older MNI-152 linear template, which has the same overall shape. In this template, only the T1 modality was used to derive the warps, which were then applied to the T2 and PD images.

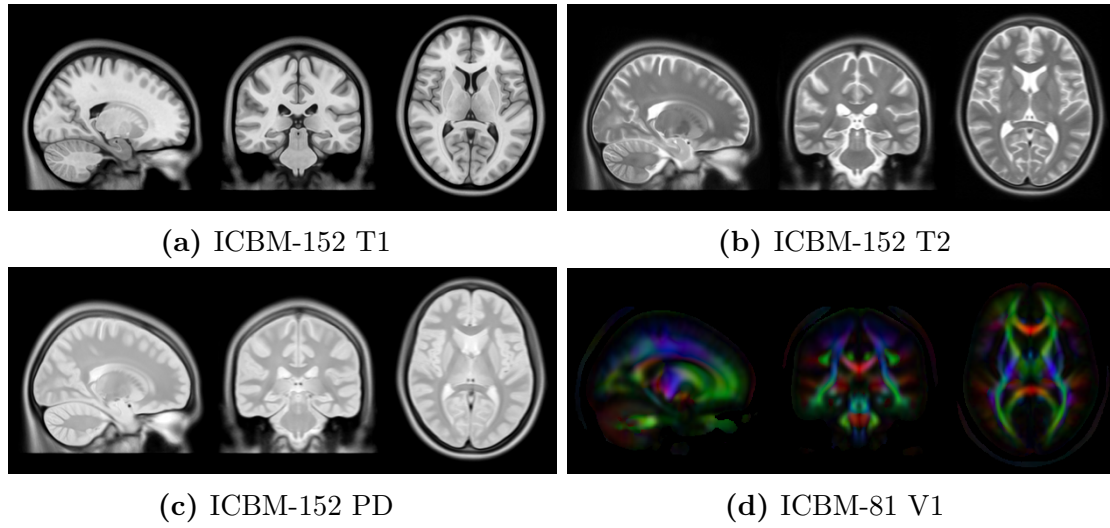


Figure 5.1: ICBM templates - Note the characteristically stretched appearance in the inferior-superior direction. Although the templates contain whole head images, a brain mask was used to constrain the optimisation to only consider within-brain regions, and therefore those regions outside of the brain appear very blurred. The DEC V1 image is considerably less sharp than the other modalities due to the use of affine-only registration.

The ICBM-81 template used only affine alignment to the same space as the ICBM-152 template, and sharpness suffers as a consequence.

5.4.2 SRI24 (Human)

The 24 subject SRI24 template (Rohlfing et al. 2010) comprises T1, FA, MD and axial diffusivity (L1) modalities (Figure 5.2). This is an example of a truly multimodal template, as all modalities were derived from the same subjects and therefore can provide both a target for simultaneous multimodal registration, and a space in which to perform a joint statistical analysis across subjects. However, only the scalar derived tensor measures, rather than the full diffusion tensor, are provided. This simplified the authors' template creation process as no tensor reorientation and resampling was required, but therefore does not allow directional information to be used when registering new subjects to this template. Additionally, only the T1 images were used to generate the template, and therefore while it provides diffusion derived modalities in template space, they did not inform the template creation directly.

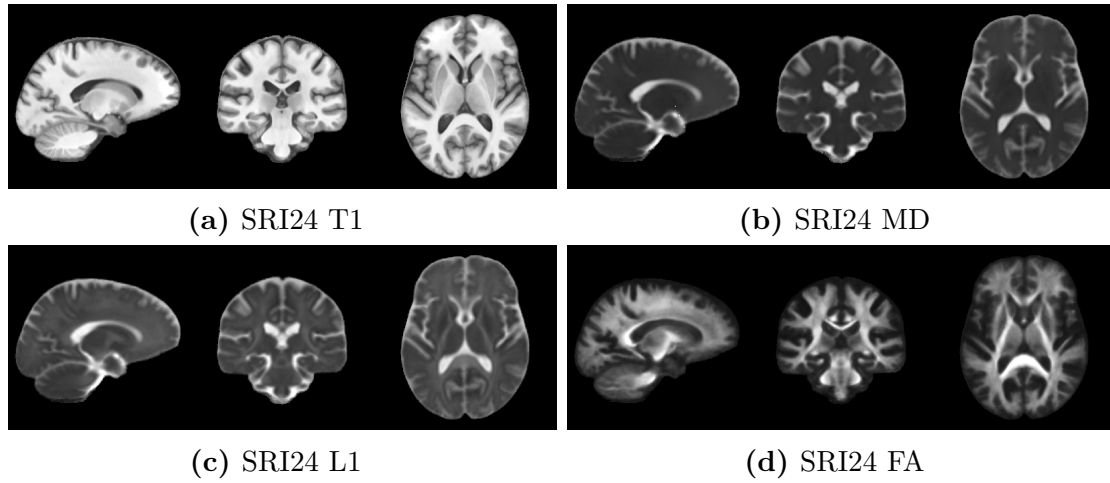


Figure 5.2: SRI24 template - Note that only brain extracted images are available for the SRI24 template. As the same subjects were used for both structural and diffusion modalities, they are all automatically in the same space.

The SRI24 template was generated using an interesting “template-free” methodology (Studholme and Cardenas 2004), which is an information theoretic approach that extends the mutual information similarity metric to an arbitrary number of images. This method seeks to minimise the voxelwise entropy across subjects, and is “non-iterative” in that no intermediate templates are generated during the process. It does, however, require all images to be loaded simultaneously and the process is therefore highly resource intensive both in terms of running time and memory usage. A gradient descent algorithm is used, and spatial unbiasing is achieved by ensuring that the average gradient across subjects, corresponding to the average displacement across subjects, is zero at each iteration. The authors describe this as a “gradient projection” approach.

5.4.3 VALiDATE29 (Squirrel Monkey)

The VALiDATE29 squirrel monkey template (Schilling et al. 2017) comprises T1, T2* and PD structural modalities, as well as both in vivo and ex vivo DTI derived FA and MD modalities (Figure 5.3). This is a somewhat unusual template in that, while all modalities are in a common space, not all subjects contributed to all modalities. Initially, a PD template was created using 18 subjects. 10 of those subjects also had T1 images, and those were transformed into template space

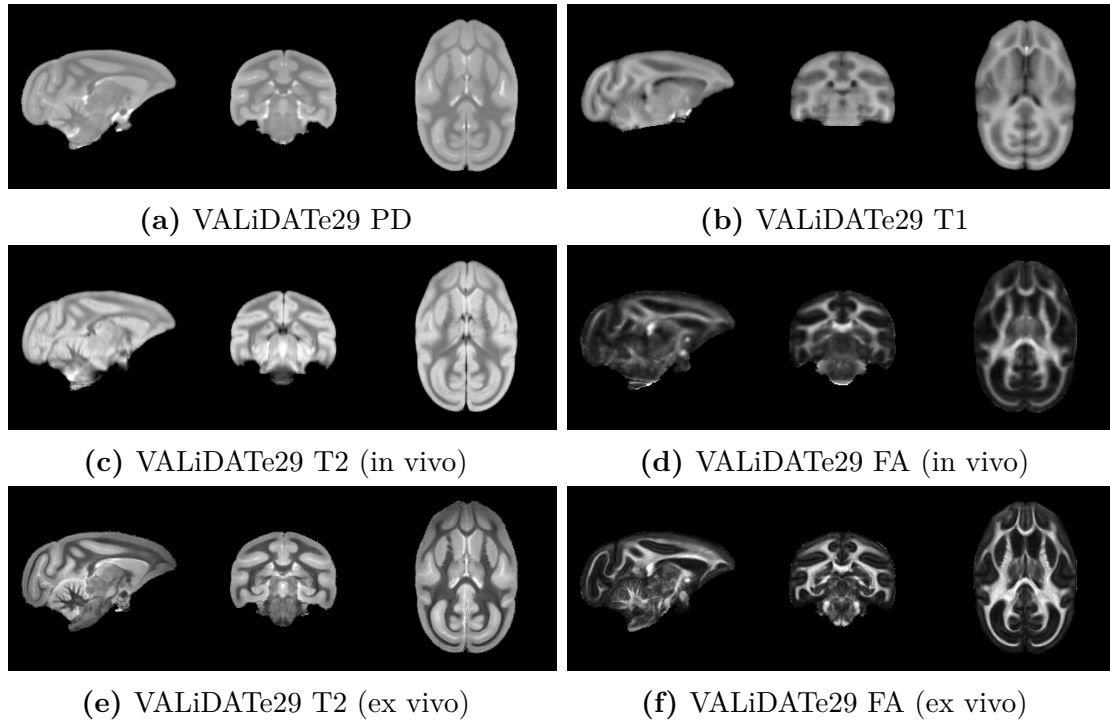


Figure 5.3: VALiDATE29 templates - Note the differences in field of view, sharpness and noise. This is in part due to inherent differences in acquisition (particularly between in vivo and ex vivo images), and in part due to the varying number of subjects used for each modality.

via the PD derived warps. A further 2 subjects with T1 images but without PD images were then separately nonlinearly registered to the temporary 10 subject T1 template. A similar process was followed to create the 22 subject T2* template. The diffusion derived modalities were initially registered to the T2* template using the T2 weighted $b=0$ images, before being refined by subsequent nonlinear iterations.

The general templating process followed the methodology of the ICBM-152 template, but as not all subjects were used to drive the registration, unbiasing of the template would not have been possible.

5.4.4 TORTOISE HCP (Human)

The TORTOISE group at NIH have recently made available a T1, T2 and DTI (and associated scalar derived quantities) atlas based on 971 subjects from the young adult HCP dataset (National Institute of Health 2020) (Figure 5.4). Very little information is provided about this atlas, but since it was developed by the

TORTOISE group it can be safely assumed that DR-TAMAS (Irfanoglu et al. 2016) was the nonlinear registration tool used to generate it. As, like MMORF, DR-TAMAS is able to simultaneously align scalar and tensor modalities, this is most likely the only multimodal template currently released where all modalities present in the atlas were used in generating it. In my opinion, this makes the TORTOISE atlas the “truest” multimodal atlas currently available.

In Irfanoglu et al. (2016), the authors state that they use the principles proposed by Joshi et al. (2004) for creating DTI templates, and the principles proposed by Avants et al. (2010) for scalar templates when using DR-TAMAS. These methods differ in how the unbiased appearance is calculated. As scalar images use the ANTs LNCC similarity metric (Avants et al. 2008), the simple voxelwise intensity average does not minimise the total dissimilarity between the template and the individual subjects. Instead, the appearance of the template must be explicitly optimised at each iteration. This normally leads to a characteristic edge-enhanced appearance in templates generated using the ANTs method, as described in Avants et al. (2010) and visible in Yang et al. (2020). This does not, however, appear to be visible in the tortoise template, which makes it unclear if the methodology described in Irfanoglu et al. (2016) was actually used or not. Furthermore, the scalar modalities do not appear to be as sharp as other templates based on this same dataset (see, for example Yang et al. (2020)), which suggests that either the warp resolution used to construct the template was fairly coarse, or that relatively few template refining iterations were performed. Given that 971 subjects were used to construct the template, this may have been a necessary trade-off to ensure that the templating process was computationally tractable. However, as no information about the template construction is provided, this is purely speculative.

5.5 Methodology

In this section I will present the methodology I have adopted for multimodal template creation. It is based on an iterative refinement technique similar to that used by Fonov et al. (2011) for the ICBM-152 template.

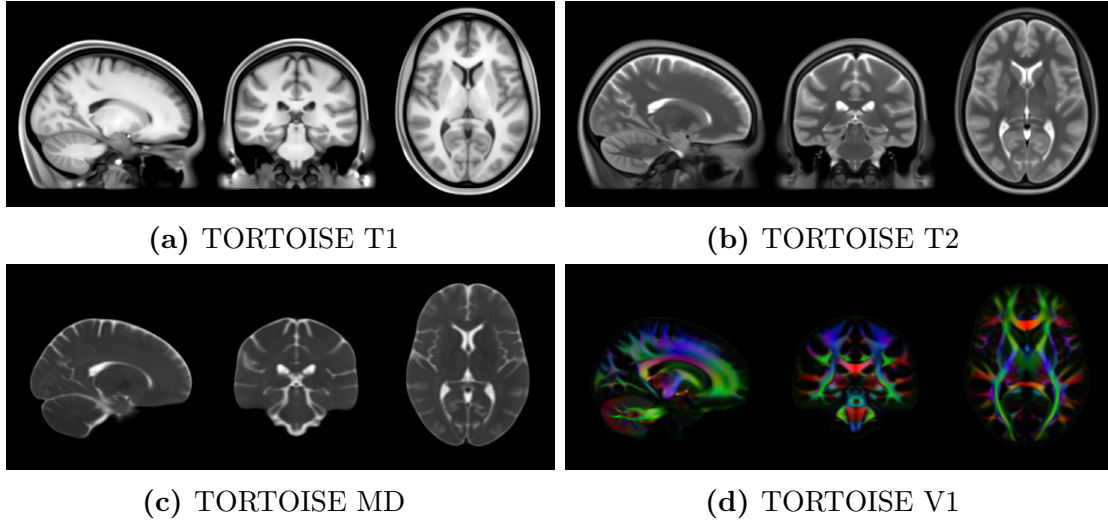


Figure 5.4: TORTOISE templates

Let a template comprising J distinct modalities be $T = \{T_1, T_2, \dots, T_J\}$. Next, define the i^{th} iteration of that template as T^i . I will now present the procedure for creating an I -iteration, J -modality, K -subject template. Note that this procedure assumes that the inter-modality, intra-subject images are already rigidly aligned.

5.5.1 Initial Affine Template

Assume there exist J distinct modalities $M = \{M_1, M_2, \dots, M_J\}$ for each subject. Begin by creating T^0 based on a 12 degrees of freedom affine transform only. A single subject is selected as an initial reference, and all other subjects are unimodal affine registered to this subject using FSL FLIRT (Jenkinson and S. M. Smith 2001), resulting in a set of affine transforms $\{\mathbf{A}_1, \mathbf{A}_2, \dots, \mathbf{A}_{K-1}, \mathbf{I}\}$. The identity transform represents the transformation from the initial reference subject to itself. Now, it may be that the initial reference subject was larger/smaller/taller/shorter etc. than the group average, and is therefore a biased space. However, a number of methods exist for finding the average of a set of affine transforms $\bar{\mathbf{A}}$. One such method implemented as part of FSL FLIRT is `midtrans`. `midtrans` performs a two step process - first solving for the best rigid body approximation to the average transform ($\bar{\mathbf{Q}}$) using the Procrustes method (Dryden and Mardia 2016; Hill et al. 2001), and then finding the optimal affine transformation $\bar{\mathbf{B}}$ to apply to $\bar{\mathbf{Q}}$, such that

the distance between the voxels in the unbiased template and the corresponding voxels in the subjects is minimised in the least squared sense (Golub and Van Loan 2013). The average transform is then calculated as $\bar{\mathbf{A}} = \bar{\mathbf{B}}\bar{\mathbf{Q}}$. Assuming that $\mathbf{A}_k$ is the transformation *from* the biased space *to* each subject, then the inverse average transform $\bar{\mathbf{A}}^{-1}$ represents the transformation *from* the unbiased space *to* the biased space. Then the transformation *from* the unbiased space *to* each subject is given by $\mathbf{A}_k\bar{\mathbf{A}}^{-1}$. The initial affine template for each modality is then:

$$T_j^0 = \frac{1}{K} \sum_{k=1}^K M_j \circ \mathbf{A}_k \bar{\mathbf{A}}^{-1} \quad (5.1)$$

5.5.2 Nonlinear Template Iterations

The nonlinear template stage follows a coarse to fine resolution iterative strategy, with the template being refined at each iteration.

The initial affine template T^0 is used as the reference for the first nonlinear iteration, T^1 . The unbiased affine transforms $\mathbf{A}_k\bar{\mathbf{A}}^{-1}$ are used to initialise MMORF. A nonlinear warp from the T^0 space to each subject is then calculated, resulting in a set of nonlinear transformations $\{\phi_1, \phi_2, \dots, \phi_K\}$. Although the T^0 space is unbiased in terms of shape under affine transformations, it may still have local shape biases under a nonlinear transform. In a similar way to the affine case, the average nonlinear transform $\bar{\phi}$ can be calculated according to:

$$\bar{\phi} = \frac{1}{K} \sum_{k=1}^K \phi_k \quad (5.2)$$

In other words, the average warp, or *bias*, is the Euclidean mean over transforms. $\bar{\phi}$ points *from* T^0 *to* the unbiased space of T^1 , and therefore $\bar{\phi}^{-1}$ is the warp *from* the shape unbiased T^1 space *to* the space in which the MMORF warps were calculated.

Now, a disadvantage of defining the average warp in this manner is that, even if each ϕ_k is diffeomorphic, $\bar{\phi}$ is not guaranteed to be diffeomorphic, as Euclidean statistics on deformations do not automatically produce diffeomorphisms. I overcome this during the inversion process using the `invwarp` method that is part of FSL FNIRT (Andersson et al. 2019). If $\bar{\phi}$ is not diffeomorphic, `invwarp` uses the

method proposed by Karacali and Davatzikos (2004) to project $\bar{\phi}^{-1}$ onto the closest warp which *is* diffeomorphic.

I recognise that from a truly diffeomorphic perspective, the Euclidean average of the deformation is not strictly the correct way of calculating the average warp. As mentioned previously, the Euclidean average of K diffeomorphisms is not guaranteed to be a diffeomorphism itself. Additionally, while applying this average to the template takes it into the space where the displacement of each point in the template is zero on average, this does not necessarily correspond to the space where the average energy (or regularisation penalty) is minimised - which is a more accurate measure of shape unbiasing. To account for this, Arsigny et al. (2006a) developed a log-Euclidean framework for performing statistical operations (such as averaging) on diffeomorphisms. This framework extends the principles of log-Euclidean metrics on symmetric positive definite tensors (Arsigny et al. 2005; Arsigny et al. 2006b) to diffeomorphisms, and generalises the concept of a matrix exponential and matrix logarithm to vector fields. In the same way that the correct method of averaging diffusion tensors is in the log domain, the correct method of averaging diffeomorphic warps is in the log domain too. In this framework, the average warp is then:

$$\bar{\phi}_{diff} = \exp \left(\sum_{k=1}^K \log (\phi_k) \right). \quad (5.3)$$

Calculating the logarithm of the diffeomorphism is, however, a complex optimisation problem in its own right, and there is no closed form solution for finding it. Therefore, calculating $\bar{\phi}_{diff}$ can take as long as (or longer than) running the registrations to find the set of ϕ_k .

However, Bossa and Olmos (2008) demonstrated, for a number of examples, that simply approximating the logarithm by $\phi_k - x$ (i.e. the displacement portion of the diffeomorphism) was sufficiently accurate for sub-mm RMS accuracy. Furthermore, Arsigny et al. (2006a) showed that, in practice, when aligning subjects to a template, the difference between $\bar{\phi}$ and $\bar{\phi}_{diff}$ tends to be small. The largest errors (differences of $\approx 30\%$ max) were confined to regions of maximum average dissimilarity between the template and the subjects (e.g., the ventricles when registering an older adult

dataset to a young adult derived template). In the case of template construction with unbiasing, the template *by definition* converges to be similar to the entire group. Therefore, at later iterations the error introduced by using $\bar{\phi}$ over $\bar{\phi}_{diff}$ is minimal, and at earlier iterations the errors due to coarse alignment should outweigh any errors due to averaging.

The i^{th} iteration of the nonlinear template for each modality is then:

$$T_j^i = \frac{1}{K} \sum_{k=1}^K M_j \circ \bar{\phi} \circ \phi_k^i \circ \mathbf{A}_k \bar{\mathbf{A}}^{-1}, \quad (5.4)$$

which is both the shape and appearance average template

Scalar modalities are simply voxelwise averaged, whereas tensor modalities are averaged in the log-Euclidean domain. As both the scalar and tensor cost functions are essentially squared L_2 norms, this simple averaging is the correct method of determining the appearance average, as shown by Joshi et al. (2004) for scalars and Irfanoglu et al. (2016) for tensors.

5.6 Lemur and Macaque Template

In this section I will present the creation of two non-human primate templates of the ring-tailed lemur (*Lemur catta*) and rhesus macaque (*Macaca mulatta*), focusing on differences between scalar driven, diffusion driven and multimodal driven templates. This is based largely on work published in F. J. Lange et al. (2020b), and is reprinted/adapted by permission from Springer Nature: Springer International Publishing, Biomedical Image Registration by Špiclin, McClelland, Kybic, Goksel ©Springer Nature Switzerland AG 2020.

5.6.1 Ethical Approval

All brains were obtained from animals that were euthanised for causes unrelated to research and treated in accordance with appropriate guidelines.

5.6.2 Methods

5.6.2.1 Data Acquisition

Full extracted brains from three ring-tailed lemurs (*Lemur catta*) and three rhesus macaques (*Macaca mulatta*) were obtained. Brains were perfusion fixed using paraformaldehyde after euthanasia for causes unrelated to the current research. Data were acquired on a 7T magnet with an Agilent DirectDrive console¹. A 2D diffusion-weighted spin-echo protocol with single line readout was used (DW-SEMS; TE/TR: 25 ms/10 s; matrix size: 128 x 128; number of slices: 128; resolution: 0.5mm (lemur) or 0.6mm (macaque) isotropic. Sixteen non-diffusion-weighted ($b = 0$ s/mm²) and 128 diffusion-weighted ($b = 4000$ s/mm²) volumes were acquired with diffusion directions distributed over the whole sphere. The brains were stored in PBS before scanning and placed in Fluorinert during the scan. Note that due to the way post mortem tissue is fixed, the data are not directly comparable to in vivo scans at the same b-value. For example, Rane and Duong (2011) show that an ex vivo $b = 4700$ s/mm² sequence would yield results similar to an in vivo $b = 1700$ s/mm² sequence.

5.6.2.2 Data Preprocessing

Diffusion tensors were fit to the data using the FSL FDT toolbox (Jenkinson et al. 2012). Additionally, FDT was used to generate an average “no-diffusion” $b=0$ image with T2 contrast. The T2 images were then bias-field corrected using FSL FAST (Y. Zhang et al. 2001). Note that T2 and DTI images *within* each subject are already co-registered by virtue of the ex vivo brain being a motionless object in the scanner.

5.6.2.3 Template Creation

A combined T2 and DTI template was created for each species individually, using three subjects per species. Template creation followed a multi-resolution iterative approach. An initial template space was chosen by affine alignment of the T2 images to one randomly chosen subject using FSL FLIRT (Jenkinson and S. M. Smith 2001).

¹Agilent Technologies, Santa Clara, CA, USA

T2 images were intensity normalised to a mean voxel intensity of 1000 before being resampled into this space, followed by voxelwise averaging across subjects. DTI images were resampled with reorientation using FSL `vecreg`, followed by log-tensor averaging. Each subject was then non-linearly aligned to the initial template at a warp resolution of 16mm isotropic, and averaged in the same way to create a new template. This process was repeated, doubling the warp resolution each time, to a final resolution of 0.5mm isotropic. At each iteration all images were smoothed using an isotropic Gaussian kernel, with a full width at half maximum (FWHM) equal to a quarter of the current warp resolution. The amount of regularisation was empirically decreased as warp resolution increased. Spatial unbiasing of the template was carried out after each iteration, such that the average displacement of every voxel in the template was approximately 0.

The template creation process was carried out three times using different combinations of the modalities to drive the registration: T2 image only, DTI image only, and multimodal using both T2 and DTI. I will refer to these methods and resulting templates as T_{T2} , T_{DT} and T_{MM} respectively. The weighting of individual modalities at $\lambda = 1$ and the amount of regularisation was the same across all methods.

5.6.2.4 Template Quality Assessment

The quality of the T2 and DTI templates were visually evaluated, and examined for obvious inconsistencies. The T2 template quality was then quantified using the average *mutual information* (MI) between each warped subject and the template as calculated by FSL `FLIRT`. MI ranges between 0 (worst) and ∞ . The DTI template quality was evaluated using the average *overlap of eigenvalue-eigenvector pairs* (OVL) metric between each warped subject and the template (Basser and Pajevic 2000). OVL is defined in equation (5.5), where λ_i and $\mathbf{e}_i$ are the i^{th} eigenvalue and eigenvector of the tensor respectively. This provides a voxelwise similarity measure of the complete eigenvalue-eigenvector tensor descriptor. It ranges between 0 and 1, with 0 representing complete dissimilarity and 1 representing identical

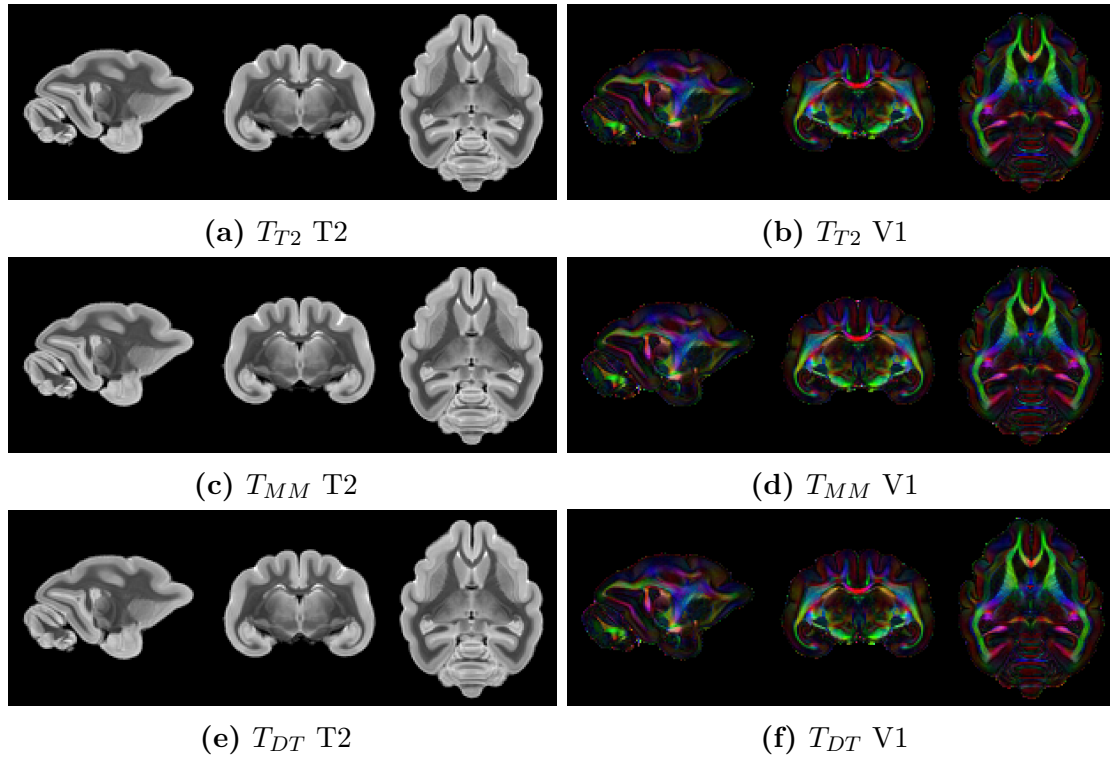


Figure 5.5: Lemur template - T2 and V1 DEC images for each of the three templating methods. Visually the results are strikingly similar when viewed at this scale.

tensors. OVL was evaluated in 3 regions, namely the entire brain, the entire brain weighted by FA, and within a mask where $FA > 0.2$.

$$OVL = \frac{\sum_{i=1}^3 Li_1 Li_2 (Vi_1^T Vi_2)^2}{\sum_{i=1}^3 Li_1 Li_2} \quad (5.5)$$

5.6.3 Results and Discussion

5.6.3.1 Visual Evaluation

Figures 5.5 and 5.6 show the T_{T2} , T_{DT} and T_{MM} results for the lemur and macaque respectively. Note that there is little to visually distinguish between results obtained from different templating methods at this scale. I therefore first describe the *common* appearance of the templates. I then focus on selected regions which demonstrate clear differences, which are highlighted in Figure 5.7.

Visually the T2 volumes are sharp, showing good contrast between tissue types, even within fairly complex structures such as the cerebellum. This is particularly

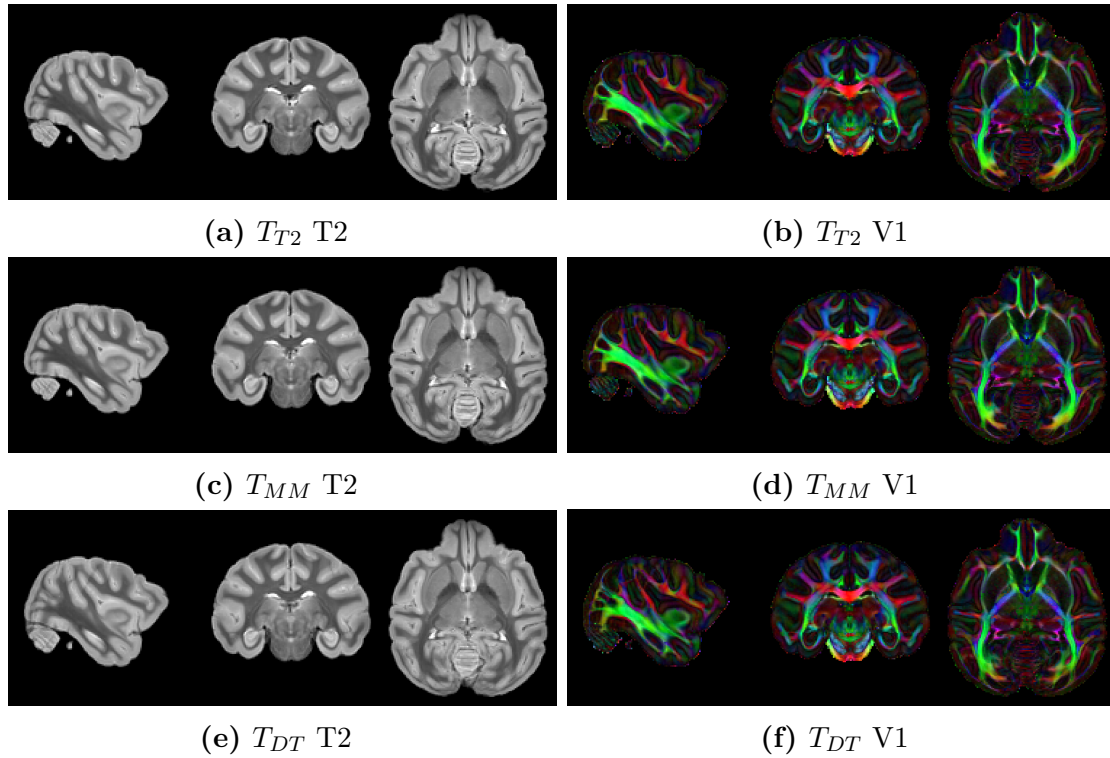


Figure 5.6: Macaque template - T2 and V1 DEC images for each of the three templating methods. Again, the results are visually very similar between methods when viewed at this scale.

true for the lemur template where the relatively simpler gyrification leads to smaller differences between subjects, making spatial normalisation somewhat easier. The occipital and parietal lobes of the macaque brain showed the highest variability between subjects and thus is unsurprisingly the least sharp template region. In general the T2 volume of the T_{MM} and T_{T2} templates do appear slightly sharper than T_{DT} . This is in line with what might be expected: the visual perception of sharpness is driven by the tissue type contrast, and that is smaller in the diffusion than in the T2 images. So a registration based exclusively on the former would result in a less sharp appearance. A clear example of this in the lemur is shown in the bottom left of figure 5.7. Here, T_{DT} has been unable to correctly align one of the sulci, whereas both T_{MM} and T_{T2} have had no such difficulty.

The DTI volumes are of excellent quality. High FA within white matter indicates that not only are the anisotropic regions of the individual subjects brought to the correct positions, but that they arrive there with a consistent orientation. Perhaps

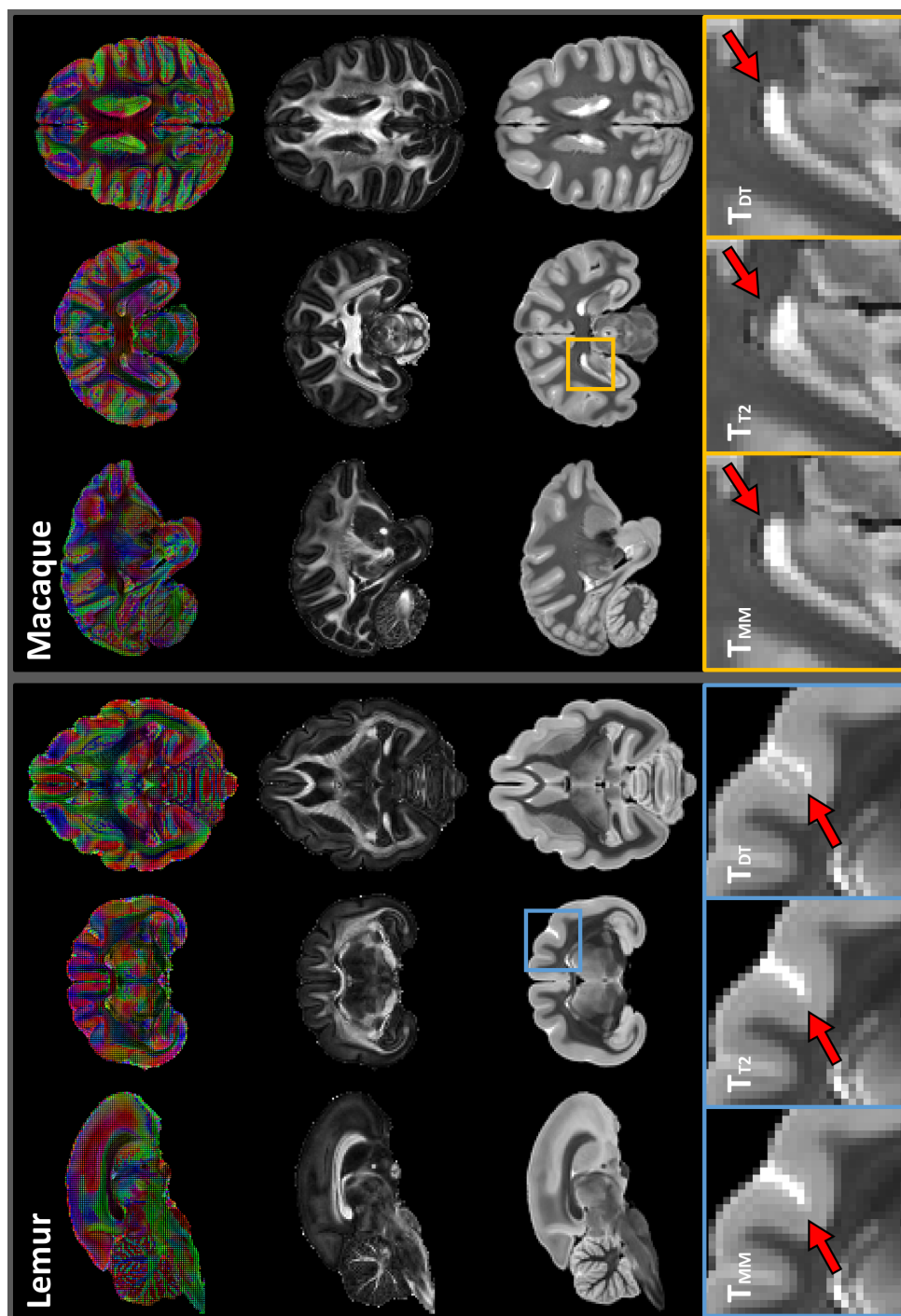


Figure 5.7: DTI (1st row), FA (2nd row) and T2 (3rd row) images from the T_{MM} template for lemur (left) and macaque (right). In the lemur I highlight a region where T_{DT} was unable to correctly align the sulci due to insufficient contrast between grey matter and fluid outside the brain (bottom left). In the macaque I highlight a region where T_{T2} was unable to correctly align the ventricles due to differences in T2 contrast of the ventricular fluid between subjects. In both cases the T_{MM} template has aligned the region correctly.

somewhat surprisingly, the T_{T2} DTI template is not clearly worse than the other methods within these high FA regions. I suggest two possible reasons for this. Firstly, the regularisation method is primarily focussed on maintaining plausible deformations. This means that the interior of the white matter is transformed in a sensible manner as it follows the tissue boundaries during registration of the T2 images. Secondly, the areas of anisotropy are always quite near tissue boundaries due to the small size of the lemur and macaque brains. Therefore, the possible deformations which correctly match tissue types are constrained in terms of allowable rotational effects. However, within some comparatively isotropic regions, larger differences are observed. In the macaque, the fluid in the ventricle of one subject had a significantly lower (darker) T2 signal than the others. The bottom right of figure 5.7 shows how this has led to ghosting around the ventricle in T_{T2} . In contrast, information in the DTI modality has allowed both T_{MM} and T_{DT} to correctly align the structure.

From this, a clear benefit, in terms of anatomical correctness, that multimodal registration approach provides over its unimodal counterparts, is evident.

5.6.3.2 Quantitative Evaluation

Quantification of how well the subject data aligns to the templates is shown for the lemur in table 5.1 and for the macaque in table 5.2. Both species show similar trends between the methods indicating that these results generalise well.

Unsurprisingly, T_{T2} has the highest average MI between the T2 images, followed by T_{MM} and finally T_{DT} . This is in line with the visual observations of the T2 image sharpness in the respective templates. However, as shown by the ventricular ghosting above, the higher MI value does not necessarily mean better global anatomical correctness.

Whole brain OVL is consistently highest for T_{MM} indicating that both high and low FA regions are being well aligned by this method. The lower performance of unimodal methods can be attributed on the one hand to T_{T2} having no knowledge

Table 5.1: Average MI and OVL measures in the lemur

	MM Template	T2 Template	DTI Template
T2 Mutual Information	1.357	1.407	1.268
OVL Whole Brain	0.859	0.854	0.850
OVL FA Weighted	0.968	0.966	0.967
OVL FA Masked	0.946	0.940	0.944

Table 5.2: Average MI and OVL measures in the macaque

	MM Template	T2 Template	DTI Template
T2 Mutual Information	1.151	1.210	1.072
OVL Whole Brain	0.784	0.769	0.774
OVL FA Weighted	0.964	0.949	0.960
OVL FA Masked	0.891	0.870	0.880

of rotations introduced by the warps, and on the other hand to poorer overlap of tissue types in T_{DT} .

When the OVL calculation is restricted to high-FA regions, either through FA-weighting or explicit FA masking, the results for all methods are higher than for the brain as a whole. As might be expected, T_{DT} performance is very similar to T_{MM} in these regions, and yet the multimodal approach still has an advantage. Interestingly, the performance deficit between T_{T2} and the other methods is greater for the macaque brain compared to the lemur. This may be due to an increase in the amount of uncontrolled rotation induced by the T_{T2} warps during alignment of the comparatively more complex cortical folding patterns in the macaque.

One point to note is that the OVL and MI measures do not have comparable units. Therefore, the fact that the differences in MI are greater than the differences in OVL should not be interpreted as meaning that the T2 volumes are in some way more affected by the templating method than the DTI volumes.

5.6.3.3 Overall Evaluation

Using multimodal data to drive the registration successfully overcame shortcomings in both the scalar and tensor only methods, whilst preserving the best aspects of each. DTI similarity measures improved across all areas of the template brains. I therefore believe that a multimodal approach to registration can only be an advantage in

the analysis of diffusion MRI data. Whilst scalar registration appears to have a slight advantage over multimodal in terms of MI scores, the lower OVL scores and some clear anatomical inconsistencies in the T2 driven template suggest that this might be attributable to over-fitting. As the goal of registration in neuroimaging is anatomical consistency rather than outright image similarity, this regularisation effect of the tensor data in the multimodal method is in fact a desirable quality.

5.6.4 Conclusion

I have shown that multimodal registration is a powerful tool for template creation, capable of leveraging complementary imaging contrasts to find a common space which is truly representative of the group data. The combination of scalar, tensor and regularisation cost functions allows for the optimisation of multiple aspects of the deformations simultaneously such that the most anatomically plausible mapping to this common space can be found. The multimodal-driven lemur and macaque templates I created show improved consistency with individual subject scans compared to both unimodal scalar and tensor driven templates.

The data used to create these templates is quite unique, with the lemur template in particular being the first of its kind for this species. I hope that by having a multimodal template available, any future analysis done on this species will be able to take advantage of increased anatomical consistency when analysing and reporting results of both individual and group studies.

5.7 UK Biobank Template

In this section I will describe the procedure by which I created a multimodal T1, T2-FLAIR (Fluid Attenuated Inversion Recovery, which I will refer to simply as T2) and DTI template using 25 randomly selected subjects from the UK Biobank project (K. L. Miller et al. 2016). I will refer to this as the UKB-25 template. This template will form the basis for the comparison between unimodal and multimodal registration accuracy in Chapter 6. This work was done in collaboration with

Christoph Arthofer (christoph.arthofer@ndcn.ox.ac.uk) who wrote the scripts to coordinate the template generation.

5.7.1 Data Acquisition and Preprocessing

The acquisition and preprocessing was performed as per Alfaro-Almagro et al. (2018). All scans were acquired on one of three identical 3T Siemens Skyra scanners. All scans were anonymised using automated defacing, and cropped down to the approximate ICBM-152 field of view. Gradient distortion correction was applied post-hoc to all images.

5.7.1.1 T1

T1 weighted images were acquired using a 3D MPRAGE acquisition with 1 mm isotropic voxel size, $2\times$ in plane acceleration and $TI/TR = 880/2000$ ms. T1 images were intensity bias field corrected using FSL FAST (Y. Zhang et al. 2001).

5.7.1.2 T2-FLAIR

T2 weighted FLAIR images were acquired using a 3D SPACE acquisition with $1.05 \times 1 \times 1$ mm voxel size, $2\times$ in plane acceleration, $TI/TR = 1800/5000$ ms, $7/8$ partial Fourier, and fat saturation. These images differ from non-FLAIR T2 weighted images in that CSF signal is nulled and therefore appears dark rather than bright. The same bias field estimated for the T1 images is applied to the T2 images.

5.7.1.3 dMRI

Diffusion MRI (dMRI) images were acquired with 2 mm isotropic voxel size, multi-band factor of 3, no in plane acceleration, $TE/TR = 92/3600$ ms, $6/8$ partial Fourier, fat saturation, 8 non-diffusion weighted $b = 0 \text{ s mm}^{-2}$ (3 of which are phase-encoding reversed) and 50 diffusion weighted images at $b = 1000 \text{ s mm}^{-2}$, 50 diffusion weighted images at $b = 2000 \text{ s mm}^{-2}$ (for 100 distinct directions). FSL TOPUP (Andersson et al. 2003; S. M. Smith et al. 2004) was used to correct for susceptibility induced off-resonance distortion using the opposite phase encoded of the $b = 0 \text{ s mm}^{-2}$ images. FSL EDDY (Andersson and Sotiropoulos 2015; Andersson and Sotiropoulos

2016; Andersson et al. 2016) was used to correct the data for eddy currents, head motion and to replace outlier volumes. DTI fitting was performed using FSL DTIFIT (according to the process outlined in Basser et al. (1994b)) using the $b = 1000 \text{ s mm}^{-2}$ data.

5.7.2 Methods

5.7.2.1 Initial Affine Template

The initial affine template space was created using the brain-extracted T1 images from each subject, and was spatially unbiased as described in Section 5.5.1. This unbiased template was then rigidly aligned to the ICBM-152 space. The T2 images were rigidly aligned to the T1 images within each subject. The $b=0$ diffusion image was used to rigidly align the DTI image to the T2 image for each subject. The total affine transformation for each modality to ICBM-152 space was then calculated by the composition of the intermediate transformations.

5.7.2.2 Nonlinear Template

18 nonlinear template iterations over 6 warp resolutions (3 iterations per warp resolution) were used to refine the template. The coarsest warp resolution was 32 mm isotropic. This resolution was doubled every 3 template iterations until a final resolution of 1 mm isotropic was reached. At each template iteration a multimodal, multi-resolution pyramid registration was performed for each subject, with pyramid levels ranging from 32 mm isotropic to the current iteration's warp resolution. Ten parameter update steps per pyramid level were performed. The Levenberg optimisation strategy was used for warp resolutions ≥ 4 mm isotropic, changing to the Majorise-Minimise strategy thereafter. Shape and appearance unbiasing of the template was performed after each iteration. All images were smoothed with a Gaussian kernel with a FWHM of $1/4$ of the pyramid resolution level. Regularisation was kept the same across iterations, and was empirically reduced for each of the finer levels of the resolution pyramid.

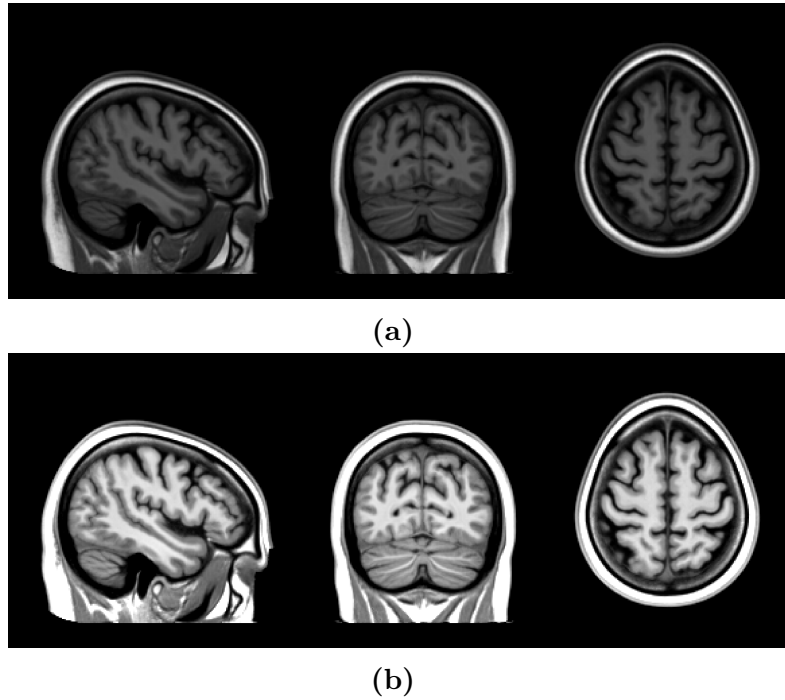


Figure 5.8: UKB-25 scalp signal - (a) shows a T1 image scaled such that no intensities are clipped. (b) shows a T1 image scaled such that intensities within the brain fall just within the clipping range. Note the distinctly high scalp signal in (a), and how dark the brain appears compared to (b). This has a direct effect on how “important” it is to correctly align the brain in order to minimise the MSQ cost function used by MMORF.

Non-brain extracted T1 and T2 images were wanted for the final template, and were therefore used to drive the templating process. However, a unique characteristic of the T1 images in the UKB required an interesting adaptation of the process. The T1 images have an uncharacteristically high signal in the scalp compared to the rest of the image, as shown in Figure 5.8. This has the unfortunate effect that the scalp dominates MMORF’s intensity based MSQ cost function. As a consequence, the relative regularisation of non-scalp regions in the T1 image is greatly increased which has a detrimental effect on detailed within-brain alignment. To counteract this, the brain extracted T1 image was added as an additional modality in the registration process. The complete set of modalities driving the nonlinear registration was therefore: T1 (whole head), T1 (brain only), T2 (whole head) and DTI.

5.7.3 Results and Discussion

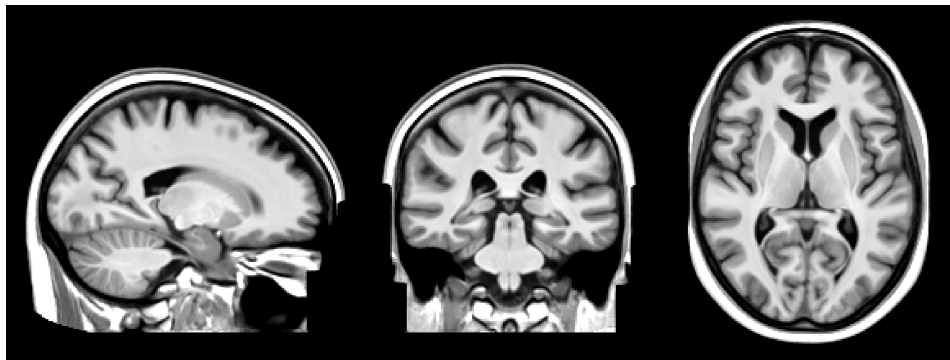
Figure 5.9 shows the final 1 mm isotropic UKB-25 template. The MD and DEC V1 maps of the DTI template are shown, as these are easier to interpret in a figure such as this than tensor glyphs. Note the sharpness of the T1 and T2 modalities - where both the subcortical structures and the cortex show excellent levels of detail, particularly compared to the SRI24 template in Figure 5.2 which used a similar number of subjects.

Evidence of the shape unbiasing is visible in the size of the ventricles, which are considerably larger than those of the TORTOISE template in Figure 5.4 - accurately reflecting the fact that the UKB-25 template is representative of an older adult group compared to the young adults of the HCP.

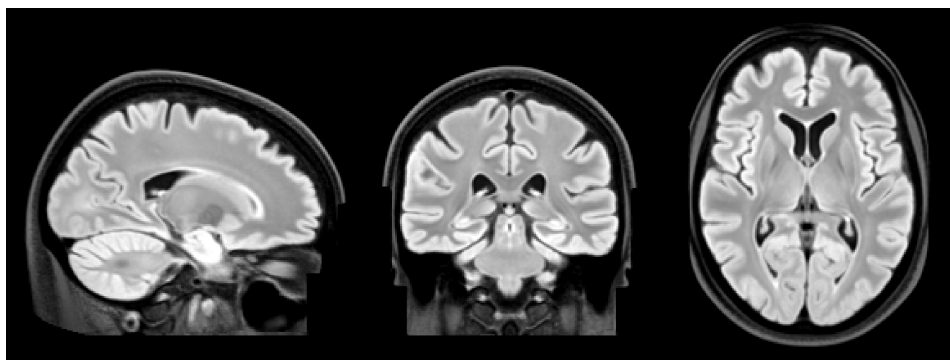
Notice that the template is sharp both within and outside of the brain. This demonstrates that the addition of the brain extracted T1 image has had the desired effect of allowing both regions to be aligned well. The MD and DEC V1 maps are clear and sharp, although not quite as sharp as the structural images - most likely due to the underlying differences in the scalar (1 mm isotropic) and diffusion (2 mm isotropic) acquisition resolution.

5.7.4 Conclusion

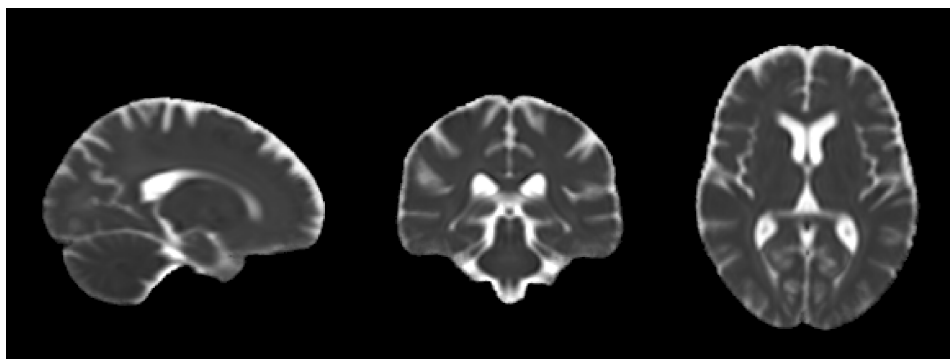
Overall, the combination of the quality and truly multimodal nature of this template make it ideal for the unimodal versus multimodal registration accuracy test presented in the next chapter.



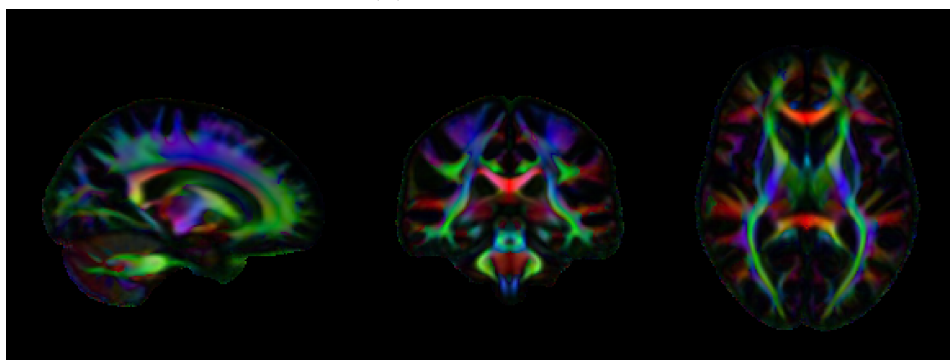
(a) UKB-25 T1



(b) UKB-25 T2



(c) UKB-25 MD



(d) UKB-25 V1

Figure 5.9: UKB-25 template

6

Multimodal vs Unimodal Accuracy

Contents

6.1	Introduction	196
6.2	Motivation	197
6.3	Methods	197
6.3.1	Template	198
6.3.2	Test Dataset	198
6.3.3	Affine Registration	199
6.3.4	Nonlinear Registration	200
6.3.5	Resampling	200
6.3.6	Group tfMRI Analysis	200
6.3.7	Warp Aggressiveness	203
6.4	Results	203
6.5	Discussion	210
6.6	Conclusion	214

6.1 Introduction

In the previous chapter I described the methodology for creating a multimodal template using MMORF. In this chapter I will now demonstrate that performing a multimodal registration to this template provides improved anatomical alignment over standard unimodal registration.

6.2 Motivation

In Chapter 5 I demonstrated how using a combination of scalar and tensor modalities to drive template construction can lead to results with improved anatomical consistency of the driving modalities compared to when using the scalar or tensor modalities in isolation. While I believe those improvements are important to demonstrate, the argument that this corresponds to an improvement in anatomical consistency is somewhat circular - in all cases I am using the data that drove the registration to evaluate the registration. This is not a problem that can be overcome by retesting with a dataset of held-out subjects, as that would still not address the issue of using the same modality to both drive and evaluate the registration.

Fortunately, the UK Biobank project (K. L. Miller et al. 2016) contains not only structural scalar and tensor modalities, but functional modalities too. Task functional MRI (tfMRI) provides a truly independent held-out modality which can be used to compare registration accuracy. The reasoning for this is as follows:

- All subjects performed the same task in the scanner
- We assume that the same anatomical regions are active during the task for all participants
- Therefore, better anatomical alignment will lead to a greater spatial overlap of activation across a group of subjects.

6.3 Methods

I will compare three registration methods in order to determine whether multimodal registration results in more accurate anatomical alignment than conventional scalar unimodal registration. The methods are based on different weightings (lambdas) of the individual modalities in the cost function:

$$\mathcal{C} = \lambda_{DTI}\mathcal{C}_{DTI} + \lambda_{T1}\mathcal{C}_{T1} + \lambda_R\mathcal{C}_R. \quad (6.1)$$

In all cases the same λ_R was used to weight the amount of regularisation.

The three methods tested are:

1. Equally weighted T1 and DTI multimodal registration (MM): $\lambda_{DTI} = \lambda_{T1} = 1$
2. T1 only registration (T1): $\lambda_{DTI} = 0, \lambda_{T1} = 1$
3. T1 only registration, with T1 weighted twice as strongly as in method 2 (T1_x2): $\lambda_{DTI} = 0, \lambda_{T1} = 2$

The inclusion of method 3 allows a distinction to be made between whether the multimodal and T1 results differ due to new information introduced by the DTI modality, or simply due to the registration being driven more strongly by the similarity term.

6.3.1 Template

The template used in this study is the 25 subject UKB multimodal template described in Chapter 5. This template is appropriate for use in this instance as it was generated using both scalar (T1, T2) and tensor (DTI) modalities. To reduce the number of variables affecting the study, and remove the impact of high scalp signal in the T1 images (see Figure 5.8), only the brain extracted version of the template was used.

6.3.2 Test Dataset

The test dataset consisted of a randomly selected sample of 100 subjects from the UKB for whom T1, DTI and and tfMRI data was available. Note that this does not include the subjects used to create the template, and therefore the template represents an approximation to the population average unbiased brain, rather than this specific group average.

T1 images were intensity bias corrected using FSL FAST prior to registration. MMORF is capable of performing bias field estimation as part of the optimisation, but since the bias field that is estimated at each iteration is based on the current “best guess” registration of the previous iteration, there is a dependency between registration accuracy and bias field estimation accuracy. I therefore chose to rather use bias corrected images to eliminate that dependency’s influence on the results.

tfMRI images were acquired with $2.4 \times 2.4 \times 2.4$ mm voxel size, multiband factor 8, no in plane acceleration, flip angle of 52° , TE/TR = 39/735 ms, and fat saturation (Alfaro-Almagro et al. 2018).

The tfMRI task was the Hariri et al. (2002) shapes and faces paradigm and used a version of the HCP protocol (Barch et al. 2013) adapted to have a shorter duration by reducing the number of stimulus block repeats (Alfaro-Almagro et al. 2018). This paradigm consists of two tasks - requiring participants to match either an emotionally salient (scared/angry) face or a shape to a pair of images including the original face/shape and a similar one. The tasks are performed in blocks of 6 presentations, each lasting for 3 seconds. Each subjects data was analysed using the first-level General Linear Model as implemented in FSL FEAT (Woolrich et al. 2001).

Contrasts of parameter estimates were generated for three contrasts of interest, namely:

1. Face matching > implicit baseline (F)
2. Shape matching > implicit baseline (S)
3. Face matching > shape matching ($F - S$)

6.3.3 Affine Registration

Affine (AF) registration to the template was performed using FSL FLIRT, and acted as the initialisation for the nonlinear registration as well as a registration method in its own right. Only the T1 brain volume was used to drive the affine registration to the template, and therefore all nonlinear methods used the same initial affine transform for the T1 images. Additionally, rigid registration was performed between the DTI and T1 volumes for each subject, using the b=0 image and a normalised mutual information cost function. The rigid and T1 affine matrices were combined using FSL `convert_xfm` and used as the initial affine transform for the DTI images.

6.3.4 Nonlinear Registration

The data of the 100 test subjects were registered to the template using three different MMORF configurations. The first utilised both the T1 and the DTI volumes in a multimodal setting, with each modality being weighted equally (MM). The second utilised the T1 volume only, weighted identically to the first method (T1). The third also used the T1 volume only, but weighted it twice as strongly as the first method (T1_x2).

All remaining configuration options were kept identical between the methods. A multi-resolution optimisation approach was used with warp knot spacing starting at 32 mm isotropic and reducing by a factor of 2 at each iteration, to a final knot spacing of 2 mm isotropic. Equal amounts of smoothing were applied to the template and subject images, using a Gaussian kernel with FWHM equal to a quarter of the knot spacing of the current resolution. Five Gauss-Newton iterations were performed at each step of the resolution pyramid.

6.3.5 Resampling

The tfMRI contrast volumes for each subject were resampled into template space using the warps generated by each of the three nonlinear methods. Additionally, an affine transformed version of each contrast was also calculated. Trilinear rather than spline interpolation was used in order ensure that the maximum bounds of the parameter estimates were not exceeded.

6.3.6 Group tfMRI Analysis

Group level mean activation maps for each combination of contrast and registration method were calculated using FSL RANDOMISE (Winkler et al. 2014; Nichols and Holmes 2002). RANDOMISE is a non-parametric permutation method for drawing inference. As only the mean value for each contrast was being tested (equivalent to a GLM design matrix with a single column of 1s), the permutations consisted of randomly sign-flipping a subset of subjects at each iteration. Two types of significance tests were performed for each contrast - namely, voxelwise

and threshold-free cluster enhancement (TFCE). It is important to note that, in both cases, p-values have been family-wise error corrected for multiple comparisons across voxels. Voxelwise tests are based on the maximum t-statistic distribution as determined by the permutations. The significance of each voxel can be meaningfully interpreted using this method. TFCE enhances the raw p-values of clusters of connected voxels that would not otherwise have passed significance testing. As such, conclusions can only be drawn based on cluster-wide significance, as opposed to individual voxels. For both tests, 5000 permutations were performed for each combination of contrast and registration method.

As there is no ground truth activation against which to compare the methods, an alternative method of comparing the results was necessary. As variations in the spatial distribution of activation between methods is not easily interpretable as being better or worse, a region of interest (ROI) based omnibus test of the average t-statistic was chosen. The process for comparing two registration methods is as follows:

1. Select an ROI in template space
2. Calculate the mean of the voxelwise t-statistic within the ROI for the two methods being compared
3. Calculate the difference in the mean t-statistic between the methods

A larger mean difference between methods indicates that the evidence of activation is stronger for one method than for the other. As it is the mean t-statistic which is being compared, higher values correspond to more spatially consistent positive activation across the aligned subjects. Assuming that activation occurs in the same anatomical location in all subjects, this can be considered a measure of registration accuracy which is completely independent of the modalities used to drive the registration. In addition to comparing the resulting group activation maps for the non-linear methods I also compare them to affine registration. The reason for that comparison is to serve as a yardstick for any differences I find between the non-linear methods.

6.3.6.1 Primary ROI

A sensible choice of ROI is crucial for this comparison to be meaningful. I have chosen those voxels where the multiple comparison corrected voxelwise significance for *both* registration methods is < 0.05 as the primary ROI within which to calculate the mean t-statistic. This is the intersection of the voxelwise significance masks, This ROI represents those regions where one is almost certain there is a true activation, and is therefore the most fair region within which to compare methods This is the only ROI in which significance testing of the difference in mean t-statistic is performed.

6.3.6.2 Secondary ROIs

Three other ROIs can be generated using the RANDOMISE results, and these are treated as secondary ROIs - that is, they are used to observe general trends, but no significance testing is performed.

The first ROI is those voxels where the voxelwise significance for *either* method is < 0.05 , which is the union of the voxelwise significance masks. The areas where only one method was significant could either correspond to regions which are anatomically part of the intersection area but were consistently misaligned in a subset of subject, or to regions which were only correctly aligned by one of the two methods. This makes the interpretation of this ROI less straightforward.

The third and fourth ROIs are the TFCE equivalent of the two voxelwise methods. The somewhat opaque nature of TFCE makes these methods less interpretable than the voxelwise methods, but they are included for completeness.

6.3.6.3 Significance Testing

Significance of the mean difference in t-statistic is not straightforward to determine. The most obvious method is to do a voxelwise paired t-test between the t-statistic images for each pair of methods - however due to spatial autocorrelation of the tfMRI signal this violates the assumption that samples are independent, causing the degrees of freedom to be overestimated and significance to be inflated.

To overcome this, I adopted the following approach. First, I restricted the analysis to only the primary ROI containing the intersection of the voxelwise significance masks for a pair of methods, as defined in Section 6.3.6.1. As this ROI is the best approximation to the region which is always activated during a particular contrast, the average t-statistic across all voxels in this ROI is representative of how well the corresponding ROI has been aligned to the template. The null hypothesis then becomes that both methods being considered have equally well aligned this ROI, and that the true difference in the average t-statistic is zero. Under this hypothesis, the registration result for either method should be interchangeable. The dataset can therefore be permuted by randomly swapping the labels on each pair of warped contrast images. Note that this is not the same as simply allowing exchangeability between the two groups. Instead, exchangeability blocks are restricted to be within-subject only.

5000 permutations were performed for each pair of registration methods, and the difference of the average t-statistic was calculated at each permutation. These 5000 measures then represent the null distribution. Significance was evaluated by counting the number of times the difference of the average t-statistic was larger than the test value, and dividing this by the number of permutations.

6.3.7 Warp Aggressiveness

How aggressively each image was deformed was measured by the 5th to 95th percentile range of the log-Jacobian determinant for voxels within the brain. The distribution of this measure across subjects was then compared between registration methods.

Spatial maps for a randomly selected subject were also compared visually to determine any gross differences in how the brain was deformed.

6.4 Results

Figure 6.1 shows the group level activation maps for affine, T1 only and multimodal registration of the F contrast, as well as the masks used for calculating the mean t-statistic method comparisons. The intensity of the activation corresponds to the

voxelwise t-statistic, and the extent corresponds to areas which were determined to be significantly active using voxelwise maximum t-statistic significance testing. The masks correspond to those regions which were found to be significantly activate using both registration methods (i.e., T1 and affine in row 2, and multimodal and T1 in row 4).

Even when using only affine registration, the activations are largely consistent with the nonlinear methods. Comparing the activations between the nonlinear methods, they are almost indistinguishable by eye. This suggests that for a sufficiently large sample size, almost any registration method will correctly identify activated regions at the group level.

Looking within the activated regions of the affine map compared to the nonlinear methods, the t-statistics are noticeably lower. Since the subject data are the same for each method, this indicates higher variability of the localisation of activation across subjects - which is consistent with less accurate registration of individual subjects to the template. The nonlinear methods are harder to visually distinguish, indicating that both methods have done a good job at aligning to the template.

Figure 6.2 presents Bland-Altman plots of the t-statistic within the primary ROI mask for T1 vs affine (6.2a to 6.2c), multimodal vs T1 (6.2d to 6.2f), multimodal vs T1_x2 (6.2g to 6.2i), and T1 vs T1_x2 (6.2j to 6.2l). The first column compares the F contrast, the second column the S contrast, and the third column the $F - S$ contrast. All of the differences were found to be highly significant ($p \leq 0.001$) using nonparametric permutation testing.

As expected, the average t-statistic within the ROI was higher for T1 vs AF for all contrasts. This serves as a reassuring sanity check that the comparison metric correctly ranks nonlinear registration higher than affine, as well as providing an intuition of what magnitude differences in mean t-statistic can be expected. In this instance, the difference is in the region of 0.5 for the three contrasts of interest.

In the MM vs T1 comparison, the mean t-statistic favours the multimodal method in all three contrasts, with average differences of 0.123, 0.130 and 0.040. In the context of the T1 vs AF comparison, the magnitude of these differences are

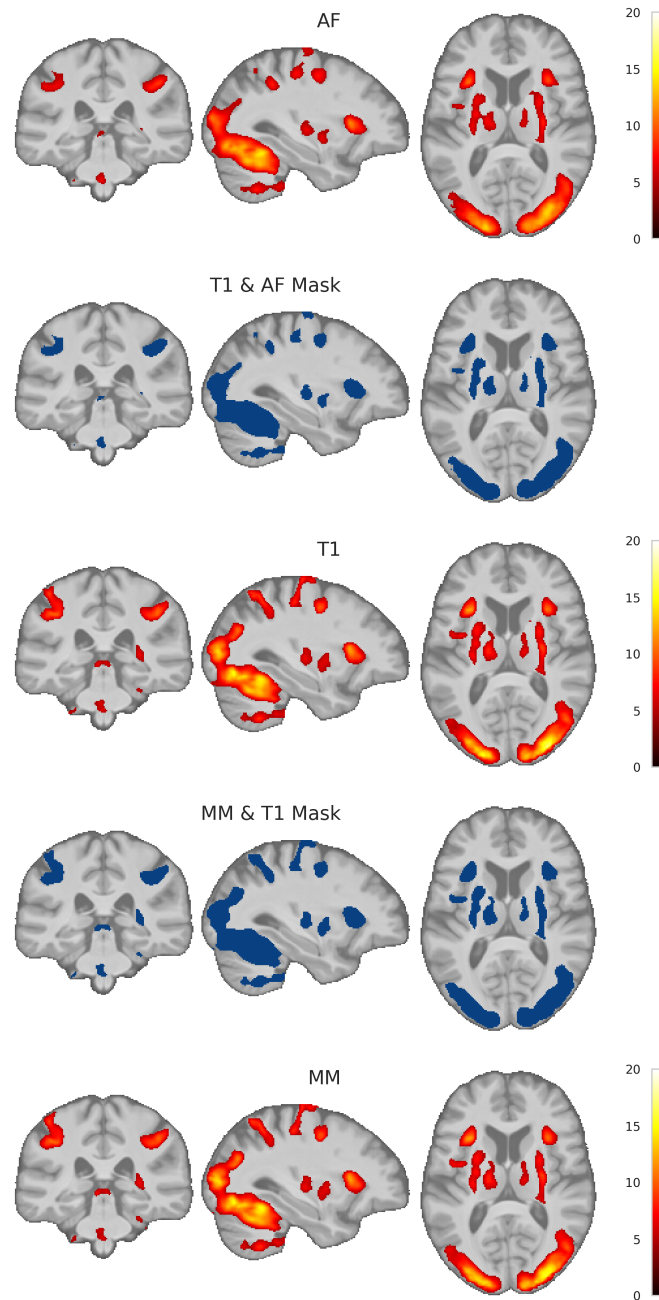


Figure 6.1: Voxelwise group level faces > baseline contrast activation - Group level t-statistics, masked to show those areas where $p < 0.05$ as determined by voxelwise nonparametric testing after affine (AF), T1 only (T1) and multimodal T1 and DTI (MM) registration are shown in rows 1, 3 and 5 respectively. Primary ROI masks of the intersection between significantly activated regions for T1 and AF, and MM and T1 are shown in rows 2 and 4. While affine registration is clearly the least sensitive method, with 100 subjects the pattern of activation is still remarkably similar to the nonlinear methods. Differences in the extent of the activation between the two nonlinear methods are almost indistinguishable by eye.

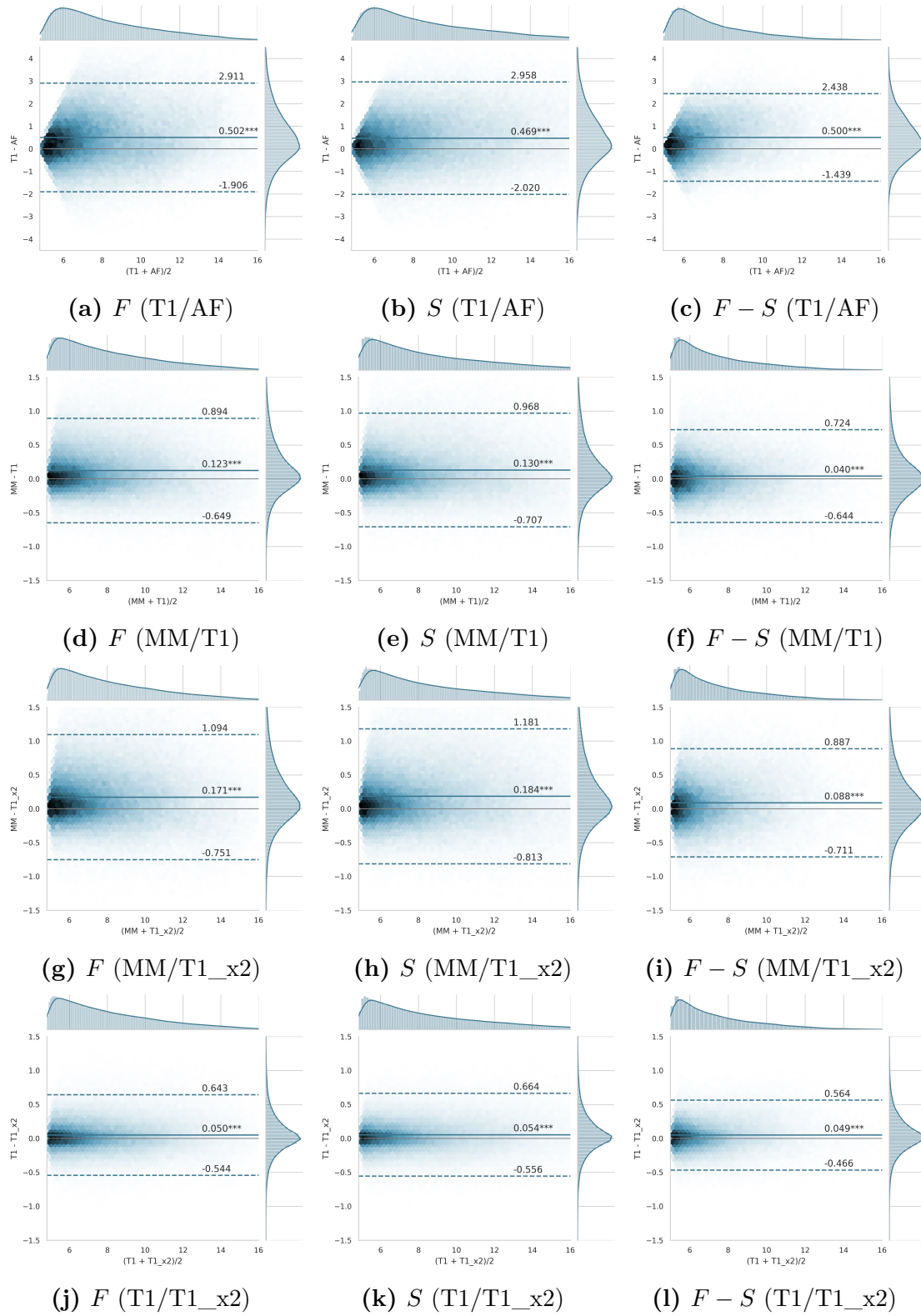


Figure 6.2: Bland-Altman masked t-statistic comparison - Columns correspond to contrasts. Rows correspond to pairs of registration methods. Solid lines show mean differences. Dashed lines show $\pm 1.96\sigma$.

quite substantial, particularly for the F and S contrasts. The addition of the DTI modality has significantly improved the registration quality within this ROI. A possible explanation for why the F and S improvements are larger than $F - S$ is the location of the ROI in each contrast. In the first two cases a substantial portion of the ROI is located in the cortex - a region with highly variable anatomy, and likely to benefit more from improved registration - whereas the third contrast is localised more in the deeper and less variable grey matter regions. Additionally the $F - S$ signal contrast is inherently more variable and therefore less likely to improve greatly from finer alignment.

When MM is evaluated against the upweighted T1_x2 results, the mean t-statistic difference increases (compared to MM vs T1) to 0.171, 0.184 and 0.088 for the three contrasts respectively. This indicates that the addition of the DTI modality does not improve registration by simply “driving the registration harder”. In fact, this suggests that just increasing the weighting of the similarity metric (or equivalently, decreasing the regularisation) has had a detrimental effect.

Comparing T1 vs T1_x2 confirms that upweighting the T1 modality has indeed decreased registration accuracy - although the difference is relatively subtle at 0.050, 0.054 and 0.049 for the three contrasts respectively.

Table 6.1 presents the difference in average t-statistic within both the primary and secondary ROI masks. Note that only the primary ROI comparisons were tested for significance. All ROI choices do however display the same trend. Multimodal registration outperforms the other methods, followed by the standard weighted T1 registration and then the upweighted T1_x2 method.

Figure 6.3 shows the distribution of 5th to 95th log-Jacobian determinant percentile range for each of the three nonlinear registration methods. T1 only registration has the smallest range, followed by multimodal and then upweighted T1. The addition of the DTI modality has clearly increased the amount of deformation being applied by the warps, although not as much as simply doubling the relative importance of the image similarity metric compared to the regularisation.

Mask	Contrast	T1 - AF	MM - T1	MM - T1_x2	T1 - T1_x2
$\text{Vox} \cap$	F	0.502***	0.123***	0.171***	0.050***
	S	0.469***	0.130***	0.184***	0.054***
	$F - S$	0.500***	0.040***	0.088***	0.049***
$\text{Vox} \cup$	F	0.557	0.134	0.185	0.051
	S	0.476	0.140	0.196	0.055
	$F - S$	0.460	0.044	0.096	0.052
$\text{TFCE} \cap$	F	0.378	0.111	0.141	0.032
	S	0.329	0.116	0.148	0.033
	$F - S$	0.195	0.022	0.055	0.030
$\text{TFCE} \cup$	F	0.379	0.120	0.152	0.033
	S	0.335	0.122	0.160	0.036
	$F - S$	0.071	0.017	0.045	0.032

Table 6.1: Further t-statistic comparisons - Difference in average t-statistic between registration methods for ROIs defined by the intersection ($\cap$) or union ($\cup$) of voxelwise or TFCE significance masks for *Faces*, *Shapes* and *Faces - Shapes* contrasts. Note that only the $\text{Vox} \cap$ ROI results were tested for significance. The remaining results are provided to demonstrate that the trend confirmed for the $\text{Vox} \cap$ ROI is observed consistently across all ROIs of interest.

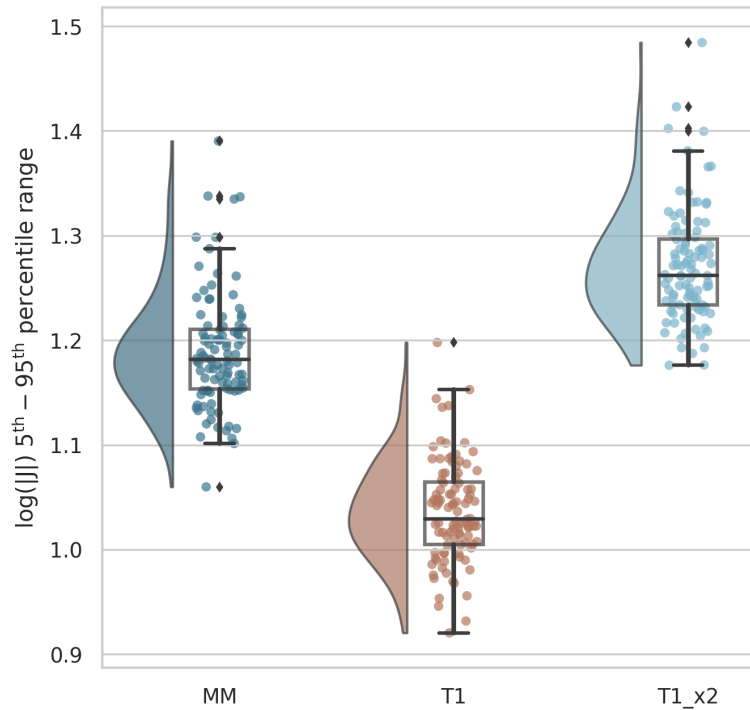


Figure 6.3: 5th to 95th log-Jacobian determinant percentile range - Percentile range is shown for the registration of each subject to the template using multimodal (MM), T1 only (T1) and upweighted T1 (T1_x2). Addition of DTI in MM registration increases the log-Jacobian determinant range (or aggressiveness) of the deformation, but not as much as doubling the scalar cost function weighting in T1_x2.

Figure 6.4 shows the Jacobian determinant map for a randomly selected subject, transformed by each of the three nonlinear methods. The T1_x2 map very closely resembles the standard T1 map in terms of which areas are expanded or contracted, but with more extreme ranges for both. The MM map appears more similar to the T1 map than T1_x2 in terms of the general range of determinant values, but with some spatially distinct variations. This suggests that the higher MM determinant ranges compared to T1 only seen in Figure 6.3 are due to the additional information from the DTI image more fundamentally changing *where* brain regions are warped to - while the higher T1_x2 determinants are due to regions that were already being compressed being compressed more and regions that were already being expanded being expanded more. In other words, multimodal registration produces a more complex deformation in order to better generalise, or fit, to the complementary data from both T1 and DTI modalities. In contrast, the upweighted T1 registration produces a more complex deformation only to improve the appearance of a single modality, which has apparently led to over-fitting.

This observation is supported by the Bland-Altman comparison of the log-Jacobian determinant shown in Figure 6.5, for the same randomly selected subject. The voxelwise difference in $\log |J|$ displays little relation to the voxelwise average value when comparing multimodal and T1 only registration. In other words, whether a voxel is deformed more or less by the multimodal warp is not strongly related to how much it is deformed by the T1 only registration. By contrast, the amount that a voxel is deformed by the upweighted T1 registration compared to the standard T1 registration is positively related to how much that voxel is deformed on average by the two methods. In line with this behaviour, the Pearson's correlation coefficient for the MM vs T1 maps is $r = 0.776$, and for the T1_x2 vs T1 maps is $r = 0.914$.

This behaviour can be summarised as:

- Adding DTI to T1 registration causes more deformation than T1 alone, but it alters where that deformation occurs.

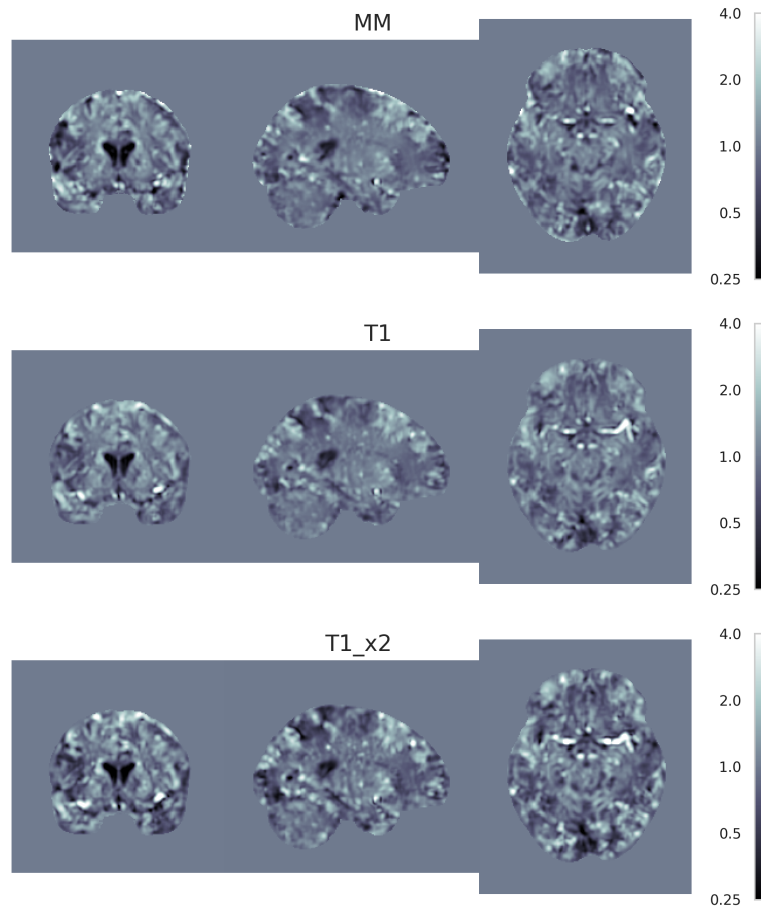


Figure 6.4: Jacobian determinant maps - The T1 and T1_x2 determinant maps display a very similar spatial pattern, but with more extreme values when the T1 cost function is upweighted. The MM and T1 maps display the opposite characteristic, in that they appear more similar in terms of the extremes in determinant values, but differ more in terms of their spatial distribution.

- Doubling the weighting of the T1 cost function also increases the degree of deformation, but it tends to just deform the same regions more harshly.

6.5 Discussion

The purpose of the work presented in this chapter has been to explore whether multimodal, simultaneous scalar and tensor, registration leads to more accurate anatomical alignment than unimodal, scalar only, registration.

There is little doubt that T1-driven registration is generally better at tissue type alignment than DTI-driven registration, and that DTI is generally better

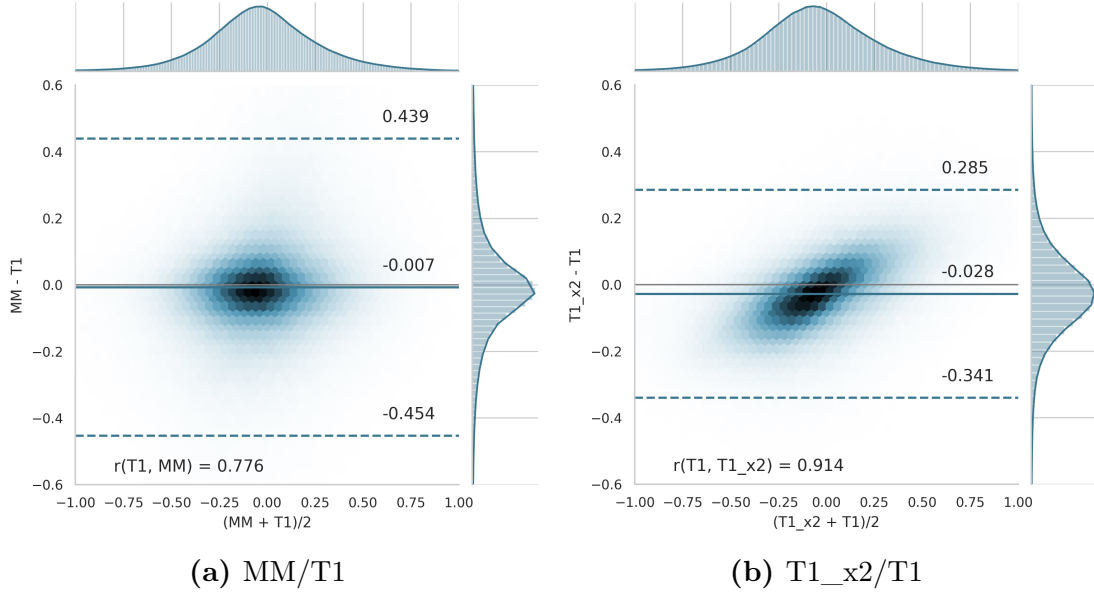


Figure 6.5: Bland-Altman log-Jacobian determinant comparison - Voxelwise comparison of $\log |J|$ between MM and T1 in (a) and T1_x2 and T1 in (b). Both comparisons show a small difference on average, as would be expected in a volume preserving deformation, but have distinct characteristics. The MM/T1 comparison shows little relationship between the average value at a voxel and the difference between methods. The T1_x2/T1 comparison, however, shows a linear correlation between the average value and the difference at each voxel. This observation is in line with a higher Pearson's correlation between the T1_x2 and T1 values ($r = 0.914$) compared to the MM and T1 values ($r = 0.776$).

at within tissue type alignment (at least for white matter) than T1. However, it is less clear whether combining modalities really leads to an improvement in overall anatomical consistency. Most evaluations of multimodal (or multichannel) registration tend to be somewhat circular in nature - using measures derived from the driving modalities to determine registration quality. Recent work by Uus et al. (2020), for example, compares unimodal T2 registration to multichannel T2, FA and diffusion derived orientation distribution functions (ODF) registration for a within subject, longitudinal, neonatal dataset. Their evaluation measures were the Dice overlap coefficients of gross anatomical structures (cortical grey matter (C-GM), hippocampus (H) and internal capsule (IC)) and the angular correlation between ODFs. The Dice coefficients are based on regions with either inherent T2 (C-GM, H) or FA contrast (IC), or distinct directionality compared to surrounding tissue (IC) at their boundaries - and those boundaries would therefore be natural

driving factors in the registration. Similarly, the angular correlation between ODFs is the exact metric used in the registration.

This is not to say that such measures are not useful or interesting, particularly when the combined modalities perform better in metrics which should only be directly influenced by one of the modalities - for example when the addition of scalar modalities improves directional consistency within white matter. However, the real goal of image registration is not to maximise image similarity, but anatomical consistency. In fact, correct anatomical consistency may lead to a decrease in image similarity metrics - but knowing that the anatomy is correctly aligned means that these dissimilarities themselves then become interesting differences to study.

As stated in Section 6.2, I believe that tfMRI provides a way to overcome the circular nature of previous evaluations. The registration warps are independent of the tfMRI modality, but a group-level tfMRI analysis is sensitive to the registration quality. Because the tfMRI signal is generally considered to originate within the grey matter only, accurate *between* tissue type alignment is necessary to correctly align this modality. But as only certain regions of the grey matter are active during a particular task, accurate *within* tissue type alignment is also necessary. Therefore, only if multimodal registration is able to better unimodal registration one or both of these counts would an improvement in tfMRI alignment be expected.

Across all three tfMRI contrasts tested, multimodal registration produced consistently higher average group-level activation than both unimodal methods. Note that even though improved group statistics is valuable in itself, this was *not* the reason for using it for validation. It is used as a proxy for improved anatomical alignment that is completely independent of the data driving the registration. The reason for including the affine registration in the comparison is to provide an intuition for what a change in average group t-statistic translates to in terms of improved anatomical alignment. Anyone who has visually compared an affine and a (decent) nonlinear registration has seen what that effect is. So, to observe a difference in average t-statistic between multimodal and unimodal non-linear registration that is

$\approx 1/4$ of the difference seen between affine and nonlinear registration implies that the improvement is much greater than is obvious from visual inspection of the two.

I believe that the voxelwise intersection ROI was the fairest choice for calculating the average t-statistic, as it leaves little doubt as to whether the comparison is being performed in an anatomical area which is *really* associated with the contrast. It is however reassuring that regardless of which ROI is used the results remain the same - although as significance testing was not performed for the other ROIs, I do not make any claims beyond acknowledging this trend has been observed.

The log-Jacobian determinant ranges clearly show that MM and T1_x2 registration do produce greater amounts of volumetric deformation than standard T1, but in the first instance this is accompanied by an improvement in registration accuracy, and in the second by a worsening. The fact that increasing the weighting of the T1 cost function in the unimodal registration *reduces* the average t-statistic provides a clear indication that the improved multimodal results are not simply being driven by the DTI modality acting to effectively reduce the amount of regularisation. If that were the case, then the T1_x2 results would have been expected to improve on those of the standard T1 weighting. The random subject's voxelwise log-Jacobian determinant comparison provides an intuition for this. Upweighting the T1 has caused the registration to do "more of the same" - that is voxels which were compressed/expanded with the standard weighting are compressed/expanded to a greater extent. One possible cause of such behaviour is that beyond a certain point, upweighting the T1 (or equivalently, down-weighting the regularisation) is no longer improving between tissue type alignment - rather it is "improving" the similarity of local within tissue type intensity variations, which is very likely a form of overfitting. By contrast, the addition of DTI has produced a different pattern of compression/expansion compared to T1 alone - a difference which has led to improved registration accuracy. This is potentially both an important and an interesting result. It is important because it offers a potential explanation of the mechanism by which multimodal registration improves registration accuracy. It is interesting because the addition of a modality which is most informative in

the white matter has improved a registration accuracy metric which is measured in the grey matter. This suggests that it is DTI's sensitivity to differences in structural connectivity (i.e., how white matter tracts connect distinct functional regions in the grey matter) which has driven this improvement, although further validation is necessary to determine this conclusively.

Considered together, these results make for a very compelling argument that multimodal registration provides more anatomically correct alignment than unimodal registration - and that it is unique information contained in the DTI modality which drives this improvement, rather than simply pushing the registration harder.

6.6 Conclusion

In this chapter I have demonstrated that multimodal registration outperforms unimodal registration as measured by group-level tfMRI activation of 100 UK Biobank subjects - an entirely independent, held-out modality. This result was shown to be highly significant across three separate contrasts within a region of interest defined by the intersection of significant voxels for the multimodal and unimodal methods ($p < 0.05$, calculated using nonparametric permutation testing). The mechanism leading to the performance improvement was shown not to be a simple decrease in regularisation allowing increased freedom of deformation, but rather that it is related to unique information contributed by the diffusion tensor. Intriguingly, this suggests the possibility that structural connectivity information has led to a more consistent estimation of functional activity by improving the anatomical accuracy of the registration.

In conclusion, this chapter makes a strong case for utilising multimodal registration whenever possible as a method of improving the anatomical consistency of nonlinear registration.

7

Conclusion and Future Directions

Contents

7.1	Summary of Contributions	215
7.1.1	MMORF	216
7.1.2	SPRED	216
7.1.3	DTI Registration	217
7.1.4	Multimodal Template Creation	217
7.1.5	Unimodal versus Multimodal Accuracy	217
7.2	Future Directions	218
7.3	Final Conclusion	219

7.1 Summary of Contributions

The contributions made in this thesis are all built upon the MultiMOdal Registration Framework (MMORF). MMORF is my implementation of a novel multimodal registration tool, capable of simultaneous inter-subject alignment of an arbitrary number of scalar and tensor magnetic resonance imaging (MRI) modalities. An advantage of developing this new framework from scratch is that it has allowed me to engineer the software to utilise graphics processing unit (GPU) parallelism from the outset, greatly reducing application runtimes. The performance improvements were not, however, treated as ends in and of themselves. Instead I leveraged them for the purpose of exploring how complex models of image deformation, that would

otherwise have been too computationally burdensome for practical use, can be used to find ways of transforming images of the brain that are more biologically plausible than current state-of-the-art methods.

7.1.1 MMORF

In Chapter 2 I presented the design and implementation decisions I have made during the development of MMORF. I showed how combining a small-deformation, B-spline parametrised transformation model with a Gauss-Newton optimisation approach can be efficiently parallelised for execution on GPUs - leading to a 10 – 20 $\times$ performance improvement at the computational bottlenecks of the algorithm. Using MMORF, 1 mm isotropic multimodal registration of the human brain can be carried out in under 45 min - fast enough to allow MMORF to be usefully applied in population studies such as the UK Biobank (UKB) project. For context, a 1 mm isotropic unimodal registration takes under 10 min using MMORF, compared to approximately 1 h for FNIRT and 6 h for ANTs-CC, despite using the computationally costly SPRED regularisation penalty - although it must be noted that this is compared to a single threaded CPU implementation of FNIRT and ANTs-CC.

7.1.2 SPRED

In Chapter 3 I showed how a symmetric prior for the regularisation of elastic deformations (SPRED), which penalises the log of the singular values of a warp, produces unimodal registration results which are as accurate as the current state-of-the-art, but with improved anatomical plausibility. The warps generated when using SPRED are invertible and smooth, with excellent control over the degree of both volumetric and shape distortions. The advantages of using this penalty are only made possible through MMORF's GPU-centric software design strategy.

7.1.3 DTI Registration

In Chapter 4 I showed how diffusion tensor imaging (DTI) registration has been implemented within MMORF. Using a similar approach to the SPRED regularisation, I show how the rotational components of the warp field are explicitly optimised during the registration process, producing results in line with other widely used DTI registration tools

7.1.4 Multimodal Template Creation

In Chapter 5 I describe the method I have chosen for creating population unbiased multimodal templates - a prerequisite for being able to use multimodal registration in group studies. I present three such templates for the ring-tailed lemur, Rhesus macaque and human. This is the first such template to be created for this lemur species. I demonstrate how using multiple modalities to drive the registration during template creation results in a template which is a better representation of the population average than when using only a single modality and assuming this correctly transforms all subject modalities into the unbiased space.

7.1.5 Unimodal versus Multimodal Accuracy

Finally, in Chapter 6 I make what I believe to be the most important contribution of this thesis - providing a quantitative demonstration that multimodal registration is able to improve the anatomical accuracy of registration. Using task functional MRI (tfMRI), an independent modality not used to drive the image registration, I am able to show that group level brain activation is measurably and significantly improved when using a combination of T1 and DTI images to drive registration to a template compared to when using T1 alone. Importantly, I show that this is not simply an effect of altering the relative amount of regularisation, but a true effect driven by the added information in the DTI modality.

7.2 Future Directions

A key aspect of this work has been unifying the approach taken to the registration of multiple modalities - moving away from using either a single modality as a proxy or having separate templates in different spaces for each modality, and towards a single, simultaneous alignment of all available images. However, there exists a major divide preventing a truly unified framework for performing medical image registration - that of the vastly different transformation approaches taken by volumetric and surface based registration methods.

Surface based methods (Fischl et al. 1999; Yeo et al. 2008; Robinson et al. 2014) have proven particularly effective in studying brain function (Coalson et al. 2018), and treat the cortical grey matter as a 2D surface that has been folded into a 3D configuration. By “inflating” this surface into a sphere, they estimate a simpler 2D mapping between brains which better constrains cortical deformations, but effectively disregards the underlying white matter connections.

These 2D and 3D image alignment methodologies are currently so different that they require the use of entirely separate suites of software tools, which provide no way of combining their results. Going forward, my aim is to bridge this divide between the methods by extending MMORF to be able to estimate volumetric warps which simultaneously respect the constraints of both surface and volumetric models of the brain.

As a first step, I propose embedding the position of the 2D surface into 3D space. Geometric information (such as curvature), anatomical information (such as myelin content), and functional activity measures can then be used as a similarity measure specific to cortical regions when driving the registration. Additionally, knowing where the cortex is in 3D will allow me to add regularisation constraints specifically at those points, thus limiting the allowable deformations to translations along the cortical sheet. In this way, I hope to preserve the consistency of white matter connectivity by controlling the deformation in 3D, while simultaneously ensuring that the 2D topology of the cortex is preserved.

An area which is becoming of increasing interest to me is the development of a deep learning based version of MMORF. I favour the idea of unsupervised methods, rather than learning to mimic MMORF outputs based on a supervised training set, as I think it may offer more long-term flexibility than the latter. MMORF would then stand as a benchmark against which to compare such a method, which I see as an important use of my software in and of itself. I believe that the deep understanding of classical registration that I have gained during this DPhil would provide a distinct advantage in both designing and assessing the performance of any future deep learning methods I may develop.

7.3 Final Conclusion

This thesis provides a strong argument for multimodal registration becoming the standard approach to volumetric MRI registration, whenever possible, due to the improved anatomical accuracy of the resulting warps. My hope is that the MMORF software I have produced will provide a route towards this, and that the care I have put into MMORF's design means that as new ways of improving the anatomical accuracy are developed, they can be easily included into the framework in a robust and efficient manner.

References

- Alexander, Andrew L. et al. (2000). “A geometric analysis of diffusion tensor measurements of the human brain”. In: *Magnetic Resonance in Medicine* 44.2, pp. 283–291.
- Alexander, Daniel C. and James C. Gee (2000). “Elastic matching of diffusion tensor images”. In: *Computer Vision and Image Understanding* 77.2, pp. 233–250.
- Alexander, Daniel C., James C. Gee, and Ruzena Bajcsy (1999). “Similarity Measures for Matching Diffusion Tensor Images”. In: *Proceedings of the British Machine Vision Conference 1999*. British Machine Vision Association, pp. 10.1–10.10.
- Alexander, Daniel C. et al. (2001). “Spatial transformations of diffusion tensor magnetic resonance images”. In: *IEEE Transactions on Medical Imaging* 20.11, pp. 1131–1139.
- Alfaro-Almagro, Fidel et al. (2018). “Image processing and Quality Control for the first 10,000 brain imaging datasets from UK Biobank”. In: *NeuroImage* 166. April 2017, pp. 400–424.
- Amit, Yali, Ulf Grenander, and Mauro Piccioni (1991). “Structural image restoration through deformable templates”. In: *Journal of the American Statistical Association* 86.414, pp. 376–387.
- Andersson, Jesper L. R., Mark Jenkinson, and Stephen M. Smith (2007). *Non-linear registration, aka spatial normalization*. Tech. rep.
- (2019). “High resolution nonlinear registration with simultaneous modelling of intensities”. In: *bioRxiv*, p. 646802. URL: <http://biorxiv.org/content/early/2019/05/22/646802.abstract>.
- Andersson, Jesper L. R., Stefan Skare, and John Ashburner (2003). “How to correct susceptibility distortions in spin-echo echo-planar images: Application to diffusion tensor imaging”. In: *NeuroImage* 20.2, pp. 870–888.
- Andersson, Jesper L. R. and Stamatiou N. Sotiropoulos (2015). “Non-parametric representation and prediction of single- and multi-shell diffusion-weighted MRI data using Gaussian processes”. In: *NeuroImage* 122, pp. 166–176.
- (2016). “An integrated approach to correction for off-resonance effects and subject movement in diffusion MR imaging”. In: *NeuroImage* 125, pp. 1063–1078.
- Andersson, Jesper L. R. et al. (2016). “Incorporating outlier detection and replacement into a non-parametric framework for movement and distortion correction of diffusion MR images”. In: *NeuroImage* 141, pp. 556–572.
- Ardekani, Babak A. et al. (2005). “Quantitative comparison of algorithms for inter-subject registration of 3D volumetric brain MRI scans”. In: *Journal of Neuroscience Methods* 142.1, pp. 67–76.
- Arsigny, Vincent et al. (2005). “Fast and simple calculus on tensors in the log-euclidean framework”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Vol. 3749 LNCS, pp. 115–122.

- Arsigny, Vincent et al. (2006a). “A log-euclidean framework for statistics on diffeomorphisms”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Vol. 4190 LNCS, pp. 924–931.
- Arsigny, Vincent et al. (2006b). “Log-Euclidean metrics for fast and simple calculus on diffusion tensors”. In: *Magnetic Resonance in Medicine* 56.2, pp. 411–421.
- Ashburner, John (2007). “A fast diffeomorphic image registration algorithm”. In: *NeuroImage* 38.1, pp. 95–113.
- Ashburner, John, Jesper L. R. Andersson, and Karl J. Friston (1999). “High-dimensional image registration using symmetric priors”. In: *NeuroImage* 9.6 I, pp. 619–628.
- (2000). “Image registration using a symmetric prior - In three dimensions”. In: *Human Brain Mapping* 9.4, pp. 212–225.
- Ashburner, John and Karl J. Friston (1999). “Nonlinear spatial normalization using basis functions”. In: *Human Brain Mapping* 7.4, pp. 254–266.
- (2005). “Unified segmentation”. In: *NeuroImage* 26.3, pp. 839–851.
- (2011). “Diffeomorphic registration using geodesic shooting and Gauss-Newton optimisation”. In: *NeuroImage* 55.3, pp. 954–967.
- Ashburner, John and Gerard R. Ridgway (2013). “Symmetric diffeomorphic modeling of longitudinal structural MRI”. In: *Frontiers in Neuroscience* 6.FEB, pp. 1–19.
- Avants, Brian B. et al. (2008). “Symmetric diffeomorphic image registration with cross-correlation: Evaluating automated labeling of elderly and neurodegenerative brain”. In: *Medical Image Analysis* 12.1, pp. 26–41.
- Avants, Brian B. et al. (2010). “The optimal template effect in hippocampus studies of diseased populations”. In: *NeuroImage* 49.3, pp. 2457–2466.
- Avants, Brian B. et al. (2011). “A reproducible evaluation of ANTs similarity metric performance in brain image registration”. In: *NeuroImage* 54.3, pp. 2033–2044.
- Avants, Brian B. et al. (2014). “The Insight ToolKit image registration framework”. In: *Frontiers in Neuroinformatics* 8.APR, pp. 1–13.
- Babaud, Jean et al. (1986). “Uniqueness of the Gaussian Kernel for Scale-Space Filtering”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-8.1, pp. 26–33.
- Bajcsy, Ruzena and Stane Kovačič (1989). “Multiresolution elastic matching”. In: *Computer Vision, Graphics, and Image Processing* 46.1, pp. 1–21.
- Bajcsy, Ruzena, Robert Lieberman, and Martin Reivich (Aug. 1983). “A Computerized System for the Elastic Matching of Deformed Radiographic Images to Idealized Atlas Images”. In: *Journal of Computer Assisted Tomography* 7.4, pp. 618–625.
- Balakrishnan, Guha et al. (2018). “An Unsupervised Learning Model for Deformable Medical Image Registration”. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 9252–9260. arXiv: 1802.02604.
- Barch, Deanna M. et al. (2013). “Function in the human connectome: Task-fMRI and individual differences in behavior”. In: *NeuroImage* 80, pp. 169–189.
- Basser, Peter J., James Mattiello, and D. LeBihan (1994a). “MR diffusion tensor spectroscopy and imaging”. In: *Biophysical Journal* 66.1, pp. 259–267.
- Basser, Peter J., James Mattiello, and Denis LeBihan (1994b). “Estimation of the Effective Self-Diffusion Tensor from the NMR Spin Echo”. In: *Journal of Magnetic Resonance, Series B* 103.3, pp. 247–254.

- Basser, Peter J. and Sinisa Pajevic (2000). “Statistical artifacts in diffusion tensor MRI (DT-MRI) caused by background noise”. In: *Magnetic Resonance in Medicine* 44.1, pp. 41–50.
- Beg, M. Faisal et al. (2005). “Computing large deformation metric mappings via geodesic flows of diffeomorphisms”. In: *International Journal of Computer Vision* 61.2, pp. 139–157.
- Bell, Nathan and Michael Garland (Dec. 2008). *Efficient Sparse Matrix-Vector Multiplication on CUDA*. NVIDIA Technical Report NVR-2008-004. NVIDIA Corporation.
- (2012). *Cusp: Generic Parallel Algorithms for Sparse Matrix and Graph Computations*. URL: <http://cusp-library.googlecode.com> (visited on 08/18/2018).
- Bohm, C. et al. (1983). “Adjustable computerized stereotaxic brain atlas for transmission and emission tomography.” In: *American Journal of Neuroradiology* 4.3, pp. 731–733.
- Bookstein, Fred L. (1997a). “Landmark methods for forms without landmarks: Morphometrics of group differences in outline shape”. In: *Medical Image Analysis* 1.3, pp. 225–243.
- (1997b). “Quadratic variation of deformations”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Vol. 1230, pp. 15–28.
- Borzi, Alfio, Kazufumi Ito, and Karl Kunisch (2003). “Optimal control formulation for determining optical flow”. In: *SIAM Journal on Scientific Computing* 24.3, pp. 818–847.
- Bossa, Matias and Salvador Olmos (2008). “A new algorithm for the computation of the group logarithm of diffeomorphisms”. In: *Mfca*. 1, pp. 1–11.
- Bro-Nielsen, Morten and Claus Gramkow (1996). “Fast fluid registration of medical images”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Vol. 1131, pp. 267–276.
- Brown, Robert W. et al. (Apr. 2014). *Magnetic Resonance Imaging: Physical Principles and Sequence Design: Second Edition*. Vol. 9780471720. 9. Chichester, UK: John Wiley & Sons Ltd, pp. 1–944.
- Burger, Martin, Jan Modersitzki, and Lars Ruthotto (2013). “A hyperelastic regularization energy for image registration”. In: *SIAM Journal on Scientific Computing* 35.1, B132–B148.
- Cachier, Pascal et al. (Feb. 2003). “Iconic feature based nonrigid registration: the PASHA algorithm”. In: *Computer Vision and Image Understanding* 89.2-3, pp. 272–298.
- Cao, Xiaohuan et al. (2017). “Deformable Image Registration Based on Similarity-Steered CNN Regression”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Vol. 10433 LNCS, pp. 300–308.
- Chen, Pei (2011). “Hessian matrix vs. Gauss-Newton Hessian matrix”. In: *SIAM Journal on Numerical Analysis* 49.4, pp. 1417–1435.
- Chenevert, Thomas L., James A. Brunberg, and James G. Pipe (1990). “Anisotropic diffusion in human white matter: Demonstration with MR techniques in vivo”. In: *Radiology* 177.2, pp. 401–405.
- Christensen, Gary Edward (1999). “Consistent linear-elastic transformations for image matching”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Ed. by Attila Kuba,

- Martin Šámal, and Andrew Todd-Pokropek. Vol. 1613. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 224–237.
- Christensen, Gary Edward, Richard D. Rabbitt, and Michael Miller (1996). “Deformable templates using large deformation kinematics”. In: *IEEE Transactions on Image Processing* 5.10, pp. 1435–1447.
- Christensen, Gary Edward et al. (2006). “Introduction to the Non-rigid Image Registration Evaluation Project (NIREP)”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 4057 LNCS, pp. 128–135.
- Chun, Il Yong and Jeffrey A. Fessler (2018). “Convolutional Dictionary Learning: Acceleration and Convergence”. In: *IEEE Transactions on Image Processing* 27.4, pp. 1697–1712.
- Coalson, Timothy S., David C. Van Essen, and Matthew F. Glasser (2018). “The impact of traditional neuroimaging methods on the spatial localization of cortical areas”. In: *Proceedings of the National Academy of Sciences* 115.27, E6356–E6365.
- Dryden, Ian L. and Kanti V. Mardia (2016). *Statistical shape analysis, with applications in R: Second edition*, pp. 1–454.
- Eklund, Anders et al. (2013). “Medical image processing on the GPU - Past, present and future”. In: *Medical Image Analysis* 17.8, pp. 1073–1094.
- Filiard, Pierre et al. (2007). “Clinical DT-MRI estimation, smoothing, and fiber tracking with log-euclidean metrics”. In: *IEEE Transactions on Medical Imaging* 26.11, pp. 1472–1482.
- Fischl, Bruce et al. (1999). “High-resolution intersubject averaging and a coordinate system for the cortical surface”. In: *Human Brain Mapping* 8.4, pp. 272–284.
- Fischler, M.A. and R.A. Elschlager (Jan. 1973). “The Representation and Matching of Pictorial Structures”. In: *IEEE Transactions on Computers* C-22.1, pp. 67–92.
- Fonov, Vladimir S. et al. (2011). “Unbiased average age-appropriate atlases for pediatric studies”. In: *NeuroImage* 54.1, pp. 313–327.
- Glasser, Matthew F. et al. (2013). “The minimal preprocessing pipelines for the Human Connectome Project”. In: *NeuroImage* 80.1, pp. 105–124.
- Golub, Gene H. and Charles F. Van Loan (2013). *Matrix Computations*. Fourth. Baltimore, Maryland: JHU Press.
- Gupta, Vikash et al. (2016). “A framework for creating population specific multimodal brain atlas using clinical T1 and diffusion tensor images”. In: *Mathematics and Visualization*. Vol. none, pp. 99–108.
- Haber, Eldad, Raya Horesh, and Jan Modersitzki (2010). “Numerical optimization for constrained image registration”. In: *Numerical Linear Algebra with Applications* 17.2-3, pp. 343–359.
- Haber, Eldad and Jan Modersitzki (2004). “Numerical methods for volume preserving image registration”. In: *Inverse Problems* 20.5, pp. 1621–1638.
- Hariri, Ahmad R. et al. (2002). “The amygdala response to emotional stimuli: A comparison of faces and scenes”. In: *NeuroImage* 17.1, pp. 317–323.
- Haskins, Grant, Uwe Kruger, and Pingkun Yan (2020). “Deep learning in medical image registration: a survey”. In: *Machine Vision and Applications* 31.1-2, p. 8.
- Heyde, Brecht et al. (2016). “Anatomical image registration using volume conservation to assess cardiac deformation from 3D ultrasound recordings”. In: *IEEE Transactions on Medical Imaging* 35.2, pp. 501–511.

- Hill, Derek L.G. et al. (2001). “Medical image registration”. In: *Physics in Medicine and Biology* 46.3, R1–R45.
- Hunter, David R. and Kenneth Lange (2004). “A Tutorial on MM Algorithms”. In: *American Statistician* 58.1, pp. 30–37.
- Irfanoglu, M. Okan et al. (2016). “DR-TAMAS: Diffeomorphic Registration for Tensor Accurate Alignment of Anatomical Structures”. In: *NeuroImage* 132, pp. 439–454.
- Jenkinson, Mark and Stephen M. Smith (2001). “A global optimisation method for robust affine registration of brain images”. In: *Medical Image Analysis* 5.2, pp. 143–156.
- Jenkinson, Mark et al. (2012). “FSL”. In: *NeuroImage* 62.2, pp. 782–790.
- Joshi, S.C. et al. (2004). “Unbiased diffeomorphic atlas construction for computational anatomy”. In: *NeuroImage* 23.SUPPL. 1, S151–S160.
- Karacali, B. and Christos Davatzikos (2004). “Estimating Topology Preserving and Smooth Displacement Fields”. In: *IEEE Transactions on Medical Imaging* 23.7, pp. 868–880.
- Klein, Arno et al. (2009). “Evaluation of 14 nonlinear deformation algorithms applied to human brain MRI registration”. In: *NeuroImage* 46.3, pp. 786–802.
- Klein, Stefan et al. (2010). “elastix: A Toolbox for Intensity-Based Medical Image Registration”. In: *IEEE Transactions on Medical Imaging* 29.1, pp. 196–205.
- Knutsen, Andrew K. et al. (2010). “A New Method to Measure Cortical Growth in the Developing Brain”. In: *Journal of Biomechanical Engineering* 132.10, p. 101004.
- Kybic, Jan and Michael Unser (2003). “Fast parametric elastic image registration”. In: *IEEE Transactions on Image Processing* 12.11, pp. 1427–1442.
- Lange, Frederik J. et al. (2020a). “A Symmetric Prior for the Regularisation of Elastic Deformations: Improved anatomical plausibility in nonlinear image registration”. In: *NeuroImage* 219.May, p. 116962.
- Lange, Frederik J. et al. (Jan. 2020b). “Multimodal MRI Template Creation in the Ring-Tailed Lemur and Rhesus Macaque”. In: *Image and Vision Computing*. Ed. by Žiga Špiclin et al. Vol. 19. Lecture Notes in Computer Science 1-2. Springer International Publishing, pp. 141–150.
- Leow, Alex D. et al. (2007). “Statistical Properties of Jacobian Maps and the Realization of Unbiased Large-Deformation Nonlinear Image Registration”. In: *IEEE Transactions on Medical Imaging* 26.6, pp. 822–832.
- Levenberg, Kenneth (1944). “A method for the solution of certain non-linear problems in least squares”. In: *Quarterly of Applied Mathematics* 2.2, pp. 164–168.
- Loeckx, Dirk et al. (2004). “Nonrigid Image Registration Using Free-Form Deformations with a Local Rigidity Constraint”. In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2004*. Ed. by Christian Barillot, David R Haynor, and Pierre Hellier. Vol. 3216. PART 1. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 639–646.
- Mang, Andreas and George Biros (Jan. 2016). “Constrained H1-Regularization Schemes for Diffeomorphic Image Registration”. In: *SIAM Journal on Imaging Sciences* 9.3, pp. 1154–1194.
- Mang, Andreas et al. (2018). “CLAIRE: A distributed-memory solver for constrained large deformation diffeomorphic image registration”. In: arXiv: 1808.04487. URL: <http://arxiv.org/abs/1808.04487>.
- Mansi, Tommaso et al. (2011). “iLogDemons: A Demons-Based Registration Algorithm for Tracking Incompressible Elastic Biological Tissues”. In: *International Journal of Computer Vision* 92.1, pp. 92–111.

- Martin, Robert C. (2014). *Agile software development, principles, patterns, and practices: Pearson new international edition*. London, England: Pearson Education.
- Miller, Karla L. et al. (2016). “Multimodal population brain imaging in the UK Biobank prospective epidemiological study”. In: *Nature Neuroscience* 19.11, pp. 1523–1536.
- Miller, Michael et al. (1993). “Mathematical textbook of deformable neuroanatomies.” In: *Proceedings of the National Academy of Sciences* 90.24, pp. 11944–11948.
- Miller, Michael et al. (1997). “Statistical methods in computational anatomy”. In: *Statistical Methods in Medical Research* 6.3, pp. 267–299.
- Modat, Marc et al. (2010). “Fast free-form deformation using graphics processing units”. In: *Computer Methods and Programs in Biomedicine* 98.3, pp. 278–284.
- Modat, Marc et al. (2011). “Log-Euclidean free-form deformation”. In: 79621Q.
- Modersitzki, Jan (2008). “FLIRT with rigidity-image registration with a local non-rigidity penalty”. In: *International Journal of Computer Vision* 76.2, pp. 153–163.
- Møller, Martin Fodslette (1993). “A scaled conjugate gradient algorithm for fast supervised learning”. In: *Neural Networks* 6.4, pp. 525–533.
- Mori, Susumu et al. (2008). “Stereotaxic white matter atlas based on diffusion tensor imaging in an ICBM template”. In: *NeuroImage* 40.2, pp. 570–582.
- National Institute of Health (2020). *TORTOISE HCP atlas*. URL: https://tortoise.nibib.nih.gov/sites/default/files/dataset/HCP%7B%5C_%7DAtlas.zip (visited on 10/02/2020).
- Nichols, Thomas E. and Andrew P Holmes (2002). “Nonparametric permutation tests for functional neuroimaging: A primer with examples”. In: *Human Brain Mapping* 15.1, pp. 1–25.
- Nocedal, Jorge and Stephen J. Wright (2006). *Numerical Optimization*. 2nd ed. 9781447122234. New York: Springer Science+Business Media, USA.
- NVIDIA (2019). “CUDA C Programming Guide”. In: *Compare A Journal Of Comparative Education* June. URL: http://docs.nvidia.com/cuda/pdf/CUDA%7B%5C_%7DC%7B%5C_%7DProgramming%7B%5C_%7DGuide.pdf.
- Oishi, Kenichi et al. (2008). “Human brain white matter atlas: Identification and assignment of common anatomical structures in superficial white matter”. In: *NeuroImage* 43.3, pp. 447–457.
- OMG Object Management Group (2017). “OMG Unified Modeling Language”. In: Version 2. URL: <https://www.omg.org/spec/UML/20161101/PrimitiveTypes.xmi>.
- Ou, Yangming et al. (2014). “Comparative Evaluation of Registration Algorithms in Different Brain Databases With Varying Difficulty: Results and Insights”. In: *IEEE Transactions on Medical Imaging* 33.10, pp. 2039–2065.
- Peeters, T. H. J. M. et al. (2009). “Analysis of Distance/Similarity Measures for Diffusion Tensor Imaging”. In: *Visualization and Processing of Tensor Fields*. Ed. by David Laidlaw and Joachim Weickert. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 113–136.
- Pennec, Xavier et al. (2005). “Riemannian Elasticity: A Statistical Regularization Framework for Non-linear Registration”. In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2005*. Ed. by James S Duncan and Guido Gerig. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 943–950.
- Pierpaoli, Carlo et al. (1996). “Diffusion tensor MR imaging of the human brain.” In: *Radiology* 201.3, pp. 637–648.
- Press, William H. et al. (2007). *Numerical recipes : the art of scientific computing*. Cambridge University Press, p. 1235.

- Rane, Swati and Timothy Q Duong (2011). “Comparison of In Vivo and Ex Vivo Diffusion Tensor Imaging in Rhesus Macaques at Short and Long Diffusion Times”. In: *The Open Neuroimag Journal* 5.Suppl 2, pp. 172–178.
- Roberts, Larry G. (1965). *Machine Perception of Three-Dimensional Solids*. Ed. by James T. Tippet et al. Cambridge, MA: The MIT Press, pp. 159–197.
- Robinson, Emma C et al. (2014). “MSM: A new flexible framework for Multimodal Surface Matching”. In: *NeuroImage* 100, pp. 414–426.
- Robinson, Emma C et al. (2018). “Multimodal surface matching with higher-order smoothness constraints”. In: *NeuroImage* 167.September 2017, pp. 453–465.
- Rohlfing, Torsten (2012). “Image Similarity and Tissue Overlaps as Surrogates for Image Registration Accuracy: Widely Used but Unreliable”. In: *IEEE Transactions on Medical Imaging* 31.2, pp. 153–163.
- Rohlfing, Torsten et al. (2003). “Volume-Preserving Nonrigid Registration of MR Breast Images Using Free-Form Deformation With an Incompressibility Constraint”. In: *IEEE Transactions on Medical Imaging* 22.6, pp. 730–741.
- Rohlfing, Torsten et al. (2010). “The SRI24 multichannel atlas of normal adult human brain structure”. In: *Human Brain Mapping* 31.5, pp. 798–819.
- Ruan, Dan et al. (2006). “Nonrigid registration using regularization that accomodates local tissue rigidity”. In: *Medical Imaging 2006: Image Processing*. Ed. by Joseph M. Reinhardt and Josien P. W. Pluim. Vol. 6144, p. 614412.
- Rueckert, Daniel et al. (1999). “Nonrigid registration using free-form deformations: application to breast MR images.” In: *IEEE Transactions on Medical Imaging* 18.8, pp. 712–21.
- Ruijters, Daniel and P. Thevenaz (2012). “GPU Prefilter for Accurate Cubic B-spline Interpolation”. In: *The Computer Journal* 55.1, pp. 15–20.
- Ruthotto, Lars et al. (2012). “Diffeomorphic susceptibility artifact correction of diffusion-weighted magnetic resonance images”. In: *Physics in Medicine and Biology* 57.18, pp. 5715–5731.
- Sanderson, Conrad and Ryan Curtin (2016). “Armadillo: a template-based C++ library for linear algebra”. In: *The Journal of Open Source Software* 1.2, p. 26.
- Schilling, Kurt G. et al. (2017). “The VALiDATe29 MRI Based Multi-Channel Atlas of the Squirrel Monkey Brain”. In: *Neuroinformatics* 15.4, pp. 321–331.
- Schnabel, Julia A. et al. (2001). “A Generic Framework for Non-rigid Registration Based on Non-uniform Multi-level Free-Form Deformations”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Vol. 2208, pp. 573–581.
- Sdika, Michaël (2008). “A Fast Nonrigid Image Registration With Constraints on the Jacobian Using Large Scale Constrained Optimization”. In: *IEEE Transactions on Medical Imaging* 27.2, pp. 271–281.
- (2013). “A Sharp Sufficient Condition for B-Spline Vector Field Invertibility. Application to Diffeomorphic Registration and Interslice Interpolation”. In: *SIAM Journal on Imaging Sciences* 6.4, pp. 2236–2257.
- Sederberg, Thomas W. and Scott R. Parry (1986). “Free-form deformation of solid geometric models”. In: *Proceedings of the 13th annual conference on Computer graphics and interactive techniques - SIGGRAPH '86*. Vol. 20. 4. New York, New York, USA: ACM Press, pp. 151–160.
- Shannon, C E (1948). “A mathematical theory of communication”. In: *The Bell System Technical Journal* 27.3, pp. 379–423.

- Simpson, Ivor J.A. et al. (2012). “Probabilistic inference of regularisation in non-rigid registration”. In: *NeuroImage* 59.3, pp. 2438–2451.
- Singh, Manbir et al. (1979). “A Digital Technique for Accurate Change Detection in Nuclear Medical Images - With Application to Myocardial Perfusion Studies Using Thallium-201”. In: *IEEE Transactions on Nuclear Science* 26.1, pp. 565–575.
- Smith, Stephen M. et al. (2004). “Advances in functional and structural MR image analysis and implementation as FSL”. In: *NeuroImage* 23.SUPPL. 1, S208–S219.
- Smith, W.D. and N.C. Wormald (1998). “Geometric separator theorems and applications”. In: *Proceedings 39th Annual Symposium on Foundations of Computer Science (Cat. No.98CB36280)*. IEEE Comput. Soc, pp. 232–243.
- Sotiras, Aristeidis, Christos Davatzikos, and Nikos Paragios (2013). “Deformable Medical Image Registration: A Survey”. In: *IEEE Transactions on Medical Imaging* 32.7, pp. 1153–1190.
- Staring, Marius, Stefan Klein, and Josien P.W. Pluim (2007). “A rigidity penalty term for nonrigid registration”. In: *Medical Physics* 34.11, pp. 4098–4108.
- Studholme, Colin and Valerie Cardenas (2004). “A template free approach to volumetric spatial normalization of brain anatomy”. In: *Pattern Recognition Letters* 25.10, pp. 1191–1202.
- Studholme, Colin, Derek L.G. Hill, and David J. Hawkes (1999). “An overlap invariant entropy measure of 3D medical image alignment”. In: *Pattern Recognition* 32.1, pp. 71–86.
- Szeliski, Richard and James Coughlan (1997). “Spline-Based Image Registration”. In: *International Journal of Computer Vision* 22.3, pp. 199–218.
- Thirion, J. P. (1998). “Image matching as a diffusion process: an analogy with Maxwell’s demons”. In: *Medical Image Analysis* 2.3, pp. 243–260.
- Tournier, Jacques-Donald et al. (2004). “Direct estimation of the fiber orientation density function from diffusion-weighted MRI data using spherical deconvolution”. In: *NeuroImage* 23.3, pp. 1176–1185.
- Unser, Michael, Akram Aldroubi, and Murray Eden (1993a). “B-spline signal processing. I. Theory”. In: *IEEE Transactions on Signal Processing* 41.2, pp. 821–833.
- (1993b). “B-spline signal processing. II. Efficiency design and applications”. In: *IEEE Transactions on Signal Processing* 41.2, pp. 834–848.
- Uus, Alena et al. (2020). “Multi-channel Registration for Diffusion MRI: Longitudinal Analysis for the Neonatal Brain”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Vol. 12120 LNCS. Springer International Publishing, pp. 111–121.
- Van Essen, David C. (1997). “A tension-based theory of morphogenesis and compact wiring in the central nervous system”. In: *Nature* 385.6614, pp. 313–318.
- Van Essen, David C. et al. (2012). “The Human Connectome Project: A data acquisition perspective”. In: *NeuroImage* 62.4, pp. 2222–2231.
- Van Essen, David C. et al. (2013). “The WU-Minn Human Connectome Project: An overview”. In: *NeuroImage* 80, pp. 62–79.
- Vercauteren, Tom et al. (2008). “Symmetric Log-Domain Diffeomorphic Registration: A Demons-Based Approach”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Vol. 5241 LNCS. PART 1, pp. 754–761.
- (2009). “Diffeomorphic demons: efficient non-parametric image registration.” In: *NeuroImage* 45.1 Suppl, S61–S72.

- Vovk, Uroš, Franjo Pernuš, and Boštjan Likar (2007). “A Review of Methods for Correction of Intensity Inhomogeneity in MRI”. In: *IEEE Transactions on Medical Imaging* 26.3, pp. 405–421.
- Winkler, Anderson M. et al. (2014). “Permutation inference for the general linear model”. In: *NeuroImage* 92, pp. 381–397.
- Witkin, Andrew P. (1984). “Scale-space filtering: A new approach to multi-scale description”. In: *ICASSP '84. IEEE International Conference on Acoustics, Speech, and Signal Processing*. Vol. 9. Institute of Electrical and Electronics Engineers, pp. 150–153.
- Witkin, Andrew P., Demetri Terzopoulos, and Michael Kass (1987). “Signal matching through scale space”. In: *International Journal of Computer Vision* 1.2, pp. 133–144.
- Woolrich, Mark W. et al. (2001). “Temporal Autocorrelation in Univariate Linear Modeling of fMRI Data”. In: *NeuroImage* 14.6, pp. 1370–1386.
- Yang, Guoyuan et al. (2020). “Sample sizes and population differences in brain template construction”. In: *NeuroImage* 206. November 2019, p. 116318.
- Yanovsky, Igor et al. (2007). “Topology preserving log-unbiased nonlinear image registration: Theory and implementation”. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.
- Yeo, B. T. Thomas et al. (2008). “Spherical Demons: Fast Surface Registration”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Vol. 5241 LNCS. PART 1, pp. 745–753.
- Yeo, B. T. Thomas et al. (2009). “DT-REFinD: Diffusion Tensor Registration With Exact Finite-Strain Differential”. In: *IEEE Transactions on Medical Imaging* 28.12, pp. 1914–1928.
- Yoo, Terry S. et al. (2002). “Engineering and algorithm design for an image processing API: A technical report on ITK - The Insight Toolkit”. In: *Studies in Health Technology and Informatics* 85, pp. 586–592.
- Zhang, Hui et al. (2006). “Deformable registration of diffusion tensor MR images with explicit orientation optimization”. In: *Medical Image Analysis* 10.5, pp. 764–785.
- Zhang, Yongyue, J. Michael Brady, and Stephen M. Smith (2001). “Segmentation of brain MR images through a hidden Markov random field model and the expectation-maximization algorithm”. In: *IEEE Transactions on Medical Imaging* 20.1, pp. 45–57.