Efficient Dense Non-Rigid Registration using the Free-Form Deformation Framework

Marc Modat

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
of
University College London.

Centre for Medical Image Computing
University College London

January 22, 2012

I, Marc Modat, confirm that the work presented in this thesis is my own.
Where information has been derived from other sources,
I confirm that this has been indicated in the thesis.

To Aurélie, my parents and Jérôme

Abstract

Medical image registration consists of finding spatial correspondences between two images or more. It is a powerful tool which is commonly used in various medical image processing tasks. Even though medical image registration has been an active topic of research for the last two decades, significant challenges in the field remain to be solved. This thesis addresses some of these challenges through extensions to the Free-Form Deformation (FFD) registration framework, which is one of the most widely used and well-established non-rigid registration algorithm.

Medical image registration is a computationally expensive task because of the high degrees of freedom of the non-rigid transformations. In this work, the FFD algorithm has been re-factored to enable fast processing, while maintaining the accuracy of the results. In addition, parallel computing paradigms have been employed to provide near real-time image registration capabilities. Further modifications have been performed to improve the registration robustness to artifacts such as tissues non-uniformity. The plausibility of the generated deformation field has been improved through the use of bio-mechanical models based regularization. Additionally, diffeomorphic extensions to the algorithm were also developed.

The work presented in this thesis has been extensively validated using brain magnetic resonance imaging of patients diagnosed with dementia or patients undergoing brain resection. It has also been applied to lung X-ray computed tomography and imaging of small animals.

Alongside with this thesis an open-source package, NiftyReg, has been developed to release the presented work to the medical imaging community.

Acknowledgements

I would like to thank my primary supervisor, Sébastien Ourselin, for his advices and guidance over the last few years. I am also very grateful for his support and friendship.

My gratitude goes as well to my secondary supervisors, Prof. Nick Fox and Prof. Dave Hawkes, who have been giving me valuable guidance and support over the course of my thesis.

I would like to thank as well my lab-mates, Jorge, Pankaj, Stefano, Abigail, Andrew, Jamie, Albert and all the others, who made our group an excellent environment to work in. I also really enjoyed working closely with Ged R., Zeike T., John A. and Jon C. and I would like to thank them for that.

I am very thankful to everyone I collaborated with over these last years within the Centre for Medical Image Computing, the Dementia Research Centre, the Wellcome Trust Centre for Neuroimaging or the Centre for Advanced Biomedical Imaging.

I am thankful to everyone that downloaded the NiftyReg package and sent me some feedback to make it a better tool.

A very special thanks to Aurélie.

Publication list

Peer-reviewed Journal Papers

[J1] G. P. Winston, P. Daga, J. Stretton, **M. Modat**, M. R. Symms, A. W. McEvoy, S. Ourselin, and J. S. Duncan, “Optic radiation tractography and vision in anterior temporal lobe resection,” Annals of Neurology, 2011, accepted.

[J2] K. Murphy, B. van Ginneken, J. Reinhardt, S. Kabus, K. Ding, X. Deng, K. Cao, K. Du, G. Christensen, V. Garcia, T. Vercauteren, N. Ayache, O. Commowick, G. Malandain, B. Glocker, N. Paragios, N. Navab, V. Gorbunova, J. Sporring, M. de Bruijne, X. Han, M. Heinrich, J. Schnabel, M. Jenkinson, C. Lorenz, **M. Modat**, J. McClelland, S. Ourselin, S. Muenzing, M. Viergever, D. De Nigris, D. Collins, T. Arbel, M. Peroni, R. Li, G. Sharp, A. Schmidt-Richberg, J. Ehrhardt, R. Werner, D. Smeets, D. Loeckx, G. Song, N. Tustison, B. Avants, J. Gee, M. Staring, S. Klein, B. Stoel, M. Urschler, M. Werlberger, J. Vandemeulebroucke, S. Rit, D. Sarrut, and J. Pluim, “Evaluation of registration methods on thoracic ct: The empire10 challenge,” Medical Imaging, IEEE Transactions on, 2011, in press.

[J3] H. R. Roth, J. R. McClelland, D. J. Boone, **M. Modat**, M. J. Cardoso, T. E. Hampshire, M. Hu, S. Punwani, S. Ourselin, G. G. Slabaugh, S. Halligan, and D. J. Hawkes, “Registration of the endoluminal surfaces of the colon derived from prone and supine ct colonography,” Medical Physics, vol. 38, no. 6, pp. 3077-89, Jun 2011.

[J4] K. K. Leung, J. Barnes, **M. Modat**, G. R. Ridgway, J. W. Bartlett, N. C. Fox, S. Ourselin, and ADNI, “Brain MAPS: An automated, accurate and robust brain extraction technique using a template library,” NeuroImage, vol. 55, no. 3, pp. 1091-108, Apr 2011.

[J5] B. A. Thomas, K. Erlandsson, **M. Modat**, L. Thurfjell, R. Vandenberghe, S. Ourselin, and B. F. Hutton, “The importance of appropriate partial volume correction for PET quantification in Alzheimers disease,” European journal of nuclear medicine and molecular imaging, vol. 38, no. 6, pp. 1104-19, Jun 2011.

[J6] M. J. Cardoso, M. J. Clarkson, G. R. Ridgway, **M. Modat**, N. C. Fox, S. Ourselin, and ADNI, “LoAd: A locally adaptive cortical segmentation algorithm,” NeuroImage, vol. 56, no. 3, pp. 1386-97, 2011.

[J7] J. R. McClelland, S. Hughes, **M. Modat**, A. Qureshi, S. Ahmad, D. B. Landau, S. Ourselin, and D. J. Hawkes, “Inter-fraction variations in respiratory motion models,” Phys Med Biol, vol. 56, no. 1, pp. 251–72, 2011.

[J8] J. O. Cleary*, **M. Modat***, F. C. Norris, A. N. Price, S. A. Jayakody, J. P. Martinez-Barbera, N. D. E. Greene, D. J. Hawkes, R. J. Ordidge, P. J. Scambler, S. Ourselin, and M. F. Lythgoe, “Magnetic resonance virtual histology for embryos: 3D atlases for automated high-throughput phenotyping,” *NeuroImage*, 2010, PMID: 20656039. *equal contribution.

[J9] J. D. Rohrer, G. R. Ridgway, **M. Modat**, S. Ourselin, S. Mead, N. C. Fox, M. N. Rossor, and J. D. Warren, “Distinct profiles of brain atrophy in frontotemporal lobar degeneration caused by progranulin and tau mutations,” *NeuroImage*, vol. 72, no. 3, pp. 1070–6, 2010.

[J10] **M. Modat**, G. R. Ridgway, Z. A. Taylor, M. Lehmann, J. Barnes, D. J. Hawkes, N. C. Fox, and S. Ourselin, “Fast free-form deformation using graphics processing units,” *Comput Meth Prog Bio*, vol. 98, no. 3, pp. 278–84, 2010.

[J11] M. Lehmann, J. D. Rohrer, M. J. Clarkson, G. R. Ridgway, R. I. Scahill, **M. Modat**, J. D. Warren, S. Ourselin, J. Barnes, M. N. Rossor, and N. C. Fox, “Reduced cortical thickness in the posterior cingulate gyrus is characteristic of both typical and atypical Alzheimer’s disease,” *Journal of Alzheimer’s disease: JAD*, vol. 20, no. 2, pp. 587–98, 2010.

[J12] J. D. Rohrer, J. D. Warren, **M. Modat**, G. R. Ridgway, A. Douiri, M. N. Rossor, S. Ourselin, and N. C. Fox, “Patterns of cortical thinning in the language variants of frontotemporal lobar degeneration,” *Neurology*, vol. 72, no. 18, pp. 1562–9, 2009.

[J13] M. Lehmann, A. Douiri, L. Kim, **M. Modat**, D. Chan, S. Ourselin, J. Barnes, and N. Fox, “Atrophy patterns in Alzheimer’s disease and semantic dementia: A comparison of freesurfer and manual volumetric measurements,” *NeuroImage*, vol. 49, no. 3, pp. 2264–74, 2009.

[J14] A. Mang, J. A. Schnabel, W. R. Crum, **M. Modat**, O. Camara-Rey, C. Palm, G. B. Caseira, H. R. Jager, S. Ourselin, T. M. Buzug, and D. J. Hawkes, “Consistency of parametric registration in serial MRI studies of brain tumor progression,” *International Journal of Computer Assisted Radiology and Surgery*, vol. 3, pp. 201–211, 2008.

[J15] J. Fripp, P. Bourgeat, O. Acosta, P. Raniga, **M. Modat**, K. E. Pike, G. Jones, G. O’Keefe, C. L. Masters, D. Ames, K. A. Ellis, P. Maruff, J. Currie, V. L. Villemagne, C. C. Rowe, O. Salvado, and S. Ourselin, “Appearance modeling of 11C PiB PET images: characterizing amyloid deposition in Alzheimer’s disease, mild cognitive impairment and healthy aging,” *NeuroImage*, vol. 43, no. 3, pp. 430–9, 2008.

Peer-reviewed Conference Papers

[C1] G. P. Winston, P. Daga, J. Stretton, **M. Modat**, M. R. Symms, A. W. McEvoy, S. Ourselin, and J. S. Duncan, “Developing optic radiation tractography to reduce visual field deficits following anterior temporal lobe resection,” in Journal of neurology, vol. 258, May 2011, pp. 54-54.

[C2] M. J. Cardoso, M. J. Clarkson, **M. Modat**, and S. Ourselin, “On the extraction of topologically correct thickness measurements using Khalimskys cubic complex,”in IPMI, 2011.

[C3] H. M. Fonteijn, M. J. Clarkson, **M. Modat**, J. Barnes, M. Lehmann, S. Ourselin, N. C. Fox, and D. C. Alexander, “An event-based disease progression model and its application to familial Alzheimer’s disease,” in IPMI, 2011.

[C4] P. Daga, G. Winston, **M. Modat**, M. J. Cardoso, M. White, A. W. McEvoy, J. Thornton, D. Hawkes, J. Duncan, and S. Ourselin, “Improved neuronavigation through integration of intra- operative anatomical and diffusion images in an interventional MRI suite,” in IPCAI, 2011.

[C5] K. K. Leung, J. Barnes, **M. Modat**, G. R. Ridgway, J. Bartlett, N. C. Fox, and S. Ourselin, “Automated brain extraction using multi-atlas propagation and segmentation (MAPS),” in IEEE International Symposium on Biomedical Imaging: From Nano to Macro, 2011.

[C6] L. Han, J. Hipwell, T. Mertzanidou, T. Carter, **M. Modat**, S. Ourselin, and D. Hawkes, “A hybrid FEM-based method for aligning prone and supine images for image guided breast surgery,” in IEEE International Symposium on Biomedical Imaging: From Nano to Macro, 2011.

[C7] M. J. Clarkson, M. J. Cardoso, **M. Modat**, G. R. Ridgway, K. K. Leung, J. D. Rohrer, N. C. Fox, and S. Ourselin, “Cross-sectional analysis using voxel or surface based cortical thickness methods: A comparison study,” in IEEE International Symposium on Biomedical Imaging: From Nano to Macro,

2011.

[C8] P. Daga, G. Winston, **M. Modat**, M. J. Cardoso, J. Stretton, M. Symms, A. W. McEvoy, D. Hawkes, J. Duncan, and S. Ourselin, "Integrating structural and diffusion mr information for optic radiation localisation in focal epilepsy patients," in IEEE International Symposium on Biomedical Imaging: From Nano to Macro, 2011.

[C9] **M. Modat**, G. R. Ridgway, P. Daga, M. J. Cardoso, D. J. Hawkes, J. Ashburner, and S. Ourselin, "Log-Euclidean free-form deformation," *in Proc. of SPIE*, 2011.

[C10] M. J. Cardoso, M. J. Clarkson, G. R. Ridgway, **M. Modat**, H. Talbot, M. Couprie, and S. Ourselin, "Topologically correct cortical segmentation using Khalimsky's cubic complex framework," *in Proc. of SPIE*, 2011.

[C11] **M. Modat**, J. McClelland, and S. Ourselin, "Lung registration using the NiftyReg package," *in MICCAI workshop: Evaluation of Methods for Pulmonary Image Registration*, 2010.

[C12] S. Pedemonte, A. Bousse, K. Erlandsson, **M. Modat**, S. Arridge, B. F. Hutton, and S. Ourselin, "GPU accelerated rotation-based emission tomography reconstruction," *in IEEE Nucl. Sci. Symp. Conf*, 2010.

[C13] H. Roth, J. McClelland, **M. Modat**, D. Boone, M. Hu, S. Ourselin, G. Slabaugh, S. Halligan, and D. J. Hawkes, "Establishing spatial correspondence between the inner colon surface visualized in prone and supine colon CT," *in Proc. MICCAI'10*, 2010.

[C14] M. J. Clarkson, I. Malone, **M. Modat**, K. K. Leung, N. Ryan, D. D. Alexander, N. C. Fox, and S. Ourselin, "A framework for using diffusion weighted imaging to improve cortical parcellation," *in Proc. MICCAI'10*, 2010.

[C15] P. Daga, **M. Modat**, C. Micallef, L. Mancini, M. White, M. J. Cardoso, N. Kitchen, A. W. McEvoy, J. Thornton, T. Yousry, D. Hawkes, and Sébastien Ourselin, "Near real time brain shift estimation for interventional MRI suite," *in High-Performance MICCAI workshop*, 2010.

[C16] **M. Modat**, T. Vercauteren, G. R. Ridgway, D. J. Hawkes, N. C. Fox, and S. Ourselin, "Diffeomorphic demons using normalised mutual information, evaluation on multi-modal brain MR images," *in Proc. of SPIE*, 2010.

[C17] J. McClelland, S. Hughes, **M. Modat**, S. Ahmad, D. Landau, S. Ourselin, and D. J. Hawkes, "Study of inter-fraction variations in respiratory motion using deformable registration based motion models," *in Proc. of the XVIth ICCR*, 2010.

[C18] **M. Modat**, Z. A. Taylor, G. R. Ridgway, J. Barnes, E. Wild, D. J. Hawkes, N. Fox, and S. Ourselin, "Nonlinear elastic spline registration: evaluation with longitudinal Huntington's disease data," *in Workshop on Biomedical Image Registration*, 2010, pp. 128–139.

[C19] M. J. Cardoso, M. J. Clarkson, **M. Modat**, G. R. Ridgway, and S. Ourselin, "Locally weighted Markov random fields for cortical segmentation," *in IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, 2010, pp. 956–959.

[C20] **M. Modat**, G. R. Ridgway, D. J. Hawkes, N. C. Fox, and S. Ourselin, "Nonrigid registration with differential bias correction using normalised mutual information," *in IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, 2010, pp. 356–359.

[C21] M. J. Cardoso, M. J. Clarkson, G. R. Ridgway, **M. Modat**, N. C. Fox, and S. Ourselin, "Improved maximum a posteriori cortical segmentation by iterative relaxation of priors," *in Proc. MICCAI'09*, vol. 12, no. Pt 2, 2009, pp. 441–9.

[C22] **M. Modat**, G. R. Ridgway, Z. Taylor, and D. J. Hawkes, “A parallel-friendly normalized mutual information gradient for free-form registration,” *in Proc. of SPIE*, 2009.

[C23] **M. Modat**, Z. A. Taylor, J. Barnes, D. J. Hawkes, N. C. Fox, and S. Ourselin, “Fast free-form deformation using the normalised mutual information gradient and graphics processing units,” *in High-Performance MICCAI workshop*, 2008.

Peer-reviewed Abstracts

[A1] B. Siow, J. Cleary, N. Greene, P. Daga, **M. Modat**, R. Ordidge, S. Ourselin, D. Alexander, and M. Lythgoe, “Mouse embryo phenotyping with contrast-enhanced micro-diffusion tensor imaging,” in 19th meeting of the International Society for Magnetic Resonance in Medicine, Montreal, 2011.

[A2] G. P. Winston, P. Daga, J. Stretton, **M. Modat**, M. R. Symms, A. W. McEvoy, S. Ourselin, and J. S. Duncan, “Propagation of probabilistic tractography of the optic radiation for neuronavigation in epilepsy surgery,” in Proceedings 19th Scientific Meeting, International Society for Magnetic Resonance in Medicine, Montreal, 2011.

[A3] M. Clarkson, **M. Modat**, L. Han, S. Ourselin, D. Hawkes, D. Alexander, M. Cardoso, H. Roth, S. Yu, and D. Barrat, “The UCL centre for medical image computing (CMIC): An integrated suite of open source software tools to support imaging research in neurosciences, oncology and cardiovascular sciences,” in IMAGINE workshop, European Congress of Radiology, 2011.

[A4] J. O. Cleary, B. Siow, N. D. E. Greene, P. Daga, **M. Modat**, R. J. Ordidge, S. Ourselin, D. Alexander, and M. F. Lythgoe, “Contrast-enhanced micro-diffusion tensor imaging for mouse embryo phenotyping,” *in ISMRM 16th Meeting of the British Chapter*, Nottingham, 2010.

[A5] B. Sinclair, J. O. Cleary, **M. Modat**, F. C. Norris, F. Wiseman, E. Fisher, M. F. Lythgoe, and S. Ourselin, “Automated tensor based morphometry for phenotyping the Tc1 mouse model of Down syndrome,” *in ISMRM 16th Meeting of the British Chapter*, Nottingham, 2010.

[A6] F. C. Norris, **M. Modat**, J. O. Cleary, A. N. Price, B. Sinclair, K. McCue, P. J. Scambler, S. Ourselin, and M. F. Lythgoe, “3D embryo atlas for high-throughput MRI phenotyping using automated segmentation propagation,” *in ISMRM 16th Meeting of the British Chapter*, Nottingham, 2010.

[A7] F. C. Norris, J. O. Cleary, **M. Modat**, A. N. Price, K. McCue, S. Beddow, P. J. Scambler, S. Ourselin, and M. F. Lythgoe, “Propagation-based morphometry in an ex vivo mouse embryo atlas – assessment and validation,” *in Proceedings 18th Scientific Meeting, International Society for Magnetic Resonance in Medicine*, Stockholm, 2010.

[A8] J. O. Cleary, **M. Modat**, A. Price, N. Greene, D. Thomas, P. Scambler, D. J. Hawkes, R. Ordidge, S. Ourselin, and M. Lythgoe, “Phenotyping in the mouse embryo using a microMRI atlas,” *in Proceedings 17th Scientific Meeting, International Society for Magnetic Resonance in Medicine*, Honolulu, 2009.

[A9] J. O. Cleary, A. N. Price, **M. Modat**, D. L. Thomas, P. J. Scambler, V. Kyriakopoulu, R. Ordidge, S. Ourselin, and M. Lythgoe, “Novel phenotyping methods using microMRI,” *Genetics Research*, vol. 91, no. 2, pp. 138–138, 2009.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Medical image registration consists of bringing a group of images into spatial alignment. These images can be acquired using different acquisition techniques such as magnetic resonance imaging (MRI), computed tomography (CT) positron emission tomography (PET) or ultra-sound (US) for example; or from the same modality. The registration cases are referred as multi- and mono-modal respectively. Figure 1.1 illustrates these two cases where a floating image is registered to two different reference images.

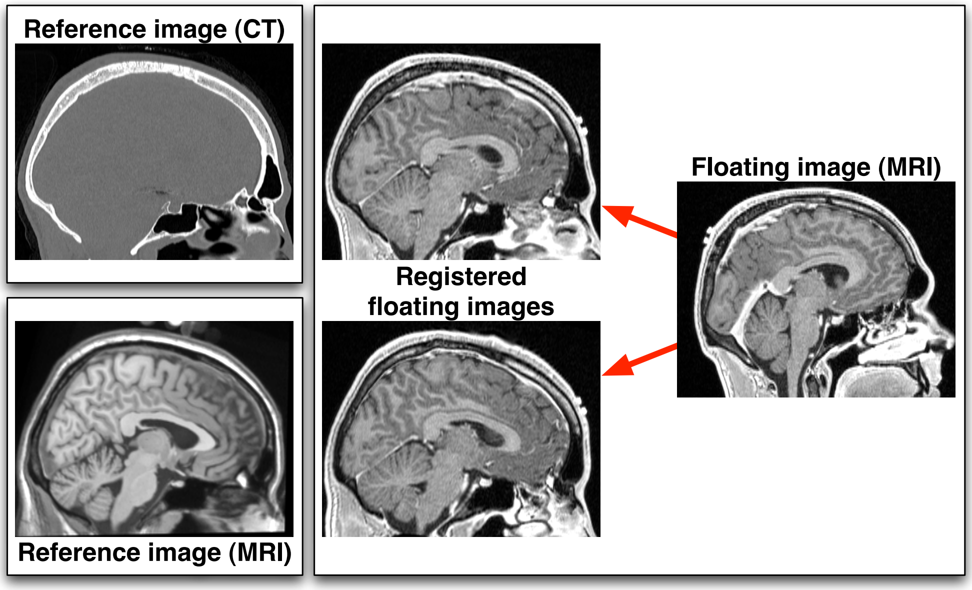


Figure 1.1: Multi- and mono-modal registration. An MRI floating image has been registered to two different reference images: CT and MRI.

1.1 Examples of medical image registration

1.1.1 Fusion

Each modality has various abilities to visualise different aspects of the imaged body. For example, T1-weighted MRI is typically used to image structural information in the brain. PET is, on the other hand, able to image functional information such as a specific molecule concentrations providing images of metabolism (e.g. fluorine-18 fluorodeoxyglucose (FDG)) or molecular pathology (e.g.. amyloid imaging). There is increasing interest in being able to merge or fuse information from several modalities [6]. To fuse images, spatial correspondences between these modalities must be found and this process is referred to as registration. Figure 1.2 illustrates the merging of an MRI with a PET scan, both from the same patient. This specific merging of these images allows the physician to visualise the structural and the functional information in relation to each other.

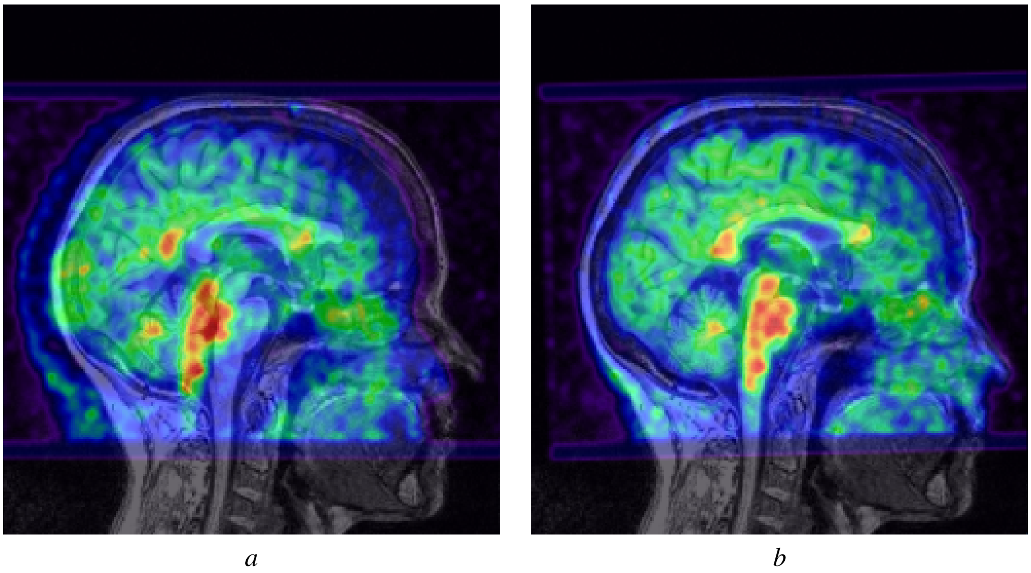


Figure 1.2: Image fusion. Both slices present a saggital view of a human brain acquired using T1-weighted MRI; an FDG-PET scan image of the same patient is overlaid. Slice *(a)* corresponds to unaligned images whereas images have been registered in slice *(b)*. Original images are from the Azheimer's disease neuroimaging initiative (ADNI) database.

1.1.2 Longitudinal changes quantification

Longitudinal acquisitions are of interest as they allow us to quantify changes occurring over time. It is used for example to quantify brain atrophy in degenerative neurological disorders such as Alzheimer's disease. Techniques such as the boundary shift integral [7, 8] (figure 1.3) require alignment of the baseline and follow-up scans to assess the brain volume changes. Registration techniques can also be used to quantify directly the volume change voxelwise, as illustrated by figure 1.10

1.1.3 Morphometric studies

Groupwise registration can be used to spatially normalise a cohort of subjects in a common space in order to statistically quantify global or local differences between groups of subjects. It is of interest for drug trials [9] or phenotyping [10] for example. Figure 1.4 shows an average image after spatial normalisation of 261 MR images including patients diagnosed with Alzheimer's disease and aged-matched controls.

1.1.4 Surgery planning propagation

Surgery planning is performed on images acquired prior to the surgery and is a very time consuming task as it often requires manual intervention. However, while performing a surgery, the organ of interest may undergo changes in shape and position. One must thus be able to update the planning to take into account of these shifts. This can be done using registration, as illustrated by figure 1.5, where a region

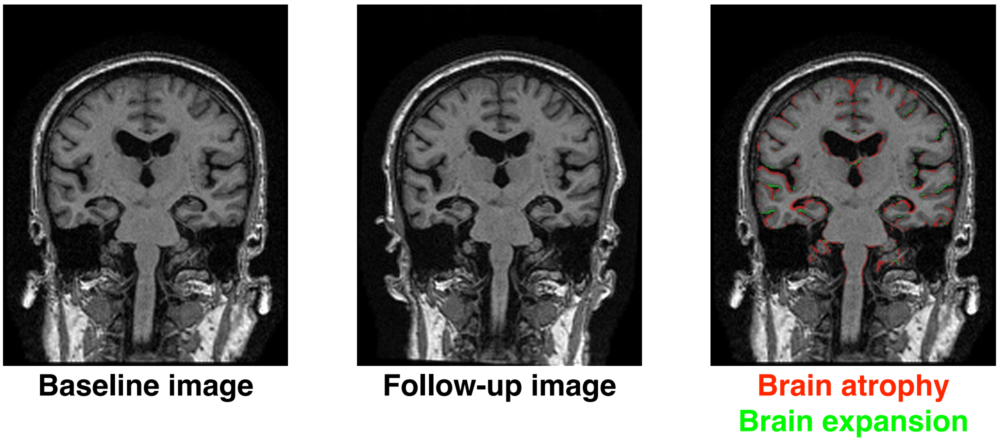


Figure 1.3: Longitudinal changes. The brain volume variations between baseline and follow-up T1-weighted volumetric MR scans have been quantified using the boundary shift integral technique after input images have been spatially normalised. The colour overlay shows in red region of loss.

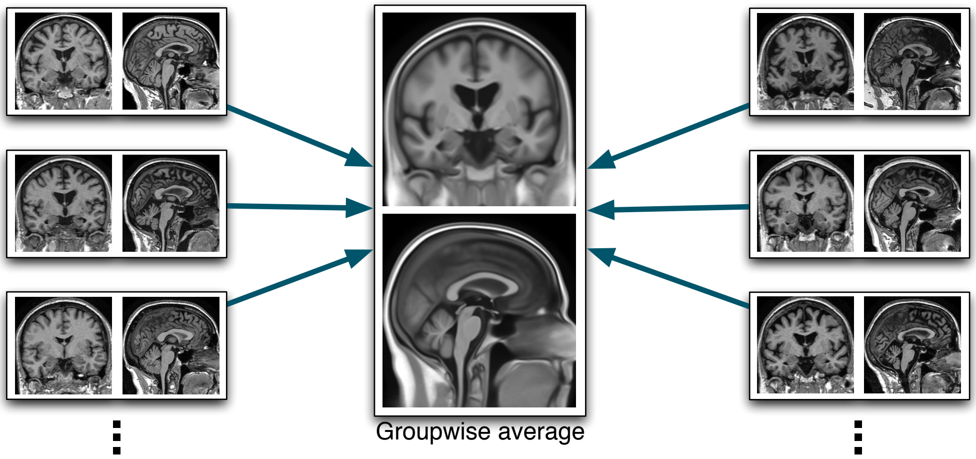


Figure 1.4: Groupwise registration. Two hundred and sixty one input images have been spatially normalised to a common space and averaged. The average image represents the mean shape of the cohort of patients used to perform the groupwise registration.

of interest is propagated using non-rigid registration from a pre-operative scan to a scan acquired at the time of the surgery.

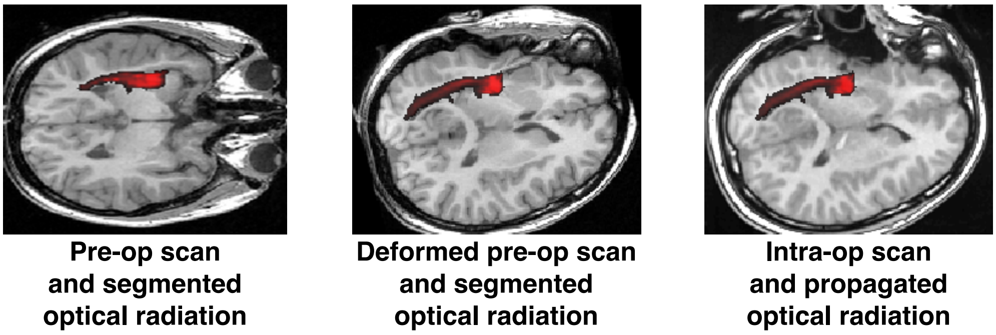

Pre-op scan and segmented optical radiation
Deformed pre-op scan and segmented optical radiation
Intra-op scan and propagated optical radiation

Figure 1.5: Axial T1-weighted MR images: a pre-operative volume is registered to an intra-operative scan. The spatial transformation is then used to propagate the optic radiation segmentation from one space to another.

1.1.5 Region of interest segmentation

Medical image registration can also be used as a pre-processing step for segmentation purposes. Indeed, when using template-based segmentation, several template images may need to be aligned with the image to segment [8]. This is illustrated in figure 1.6 where several manual segmentations of hyppocampi have been propagated to an image to segment. Registration is also used to initialise expectation-maximisation algorithms where a probabilistic atlas has first to be warped into the space of the image to segment [11].

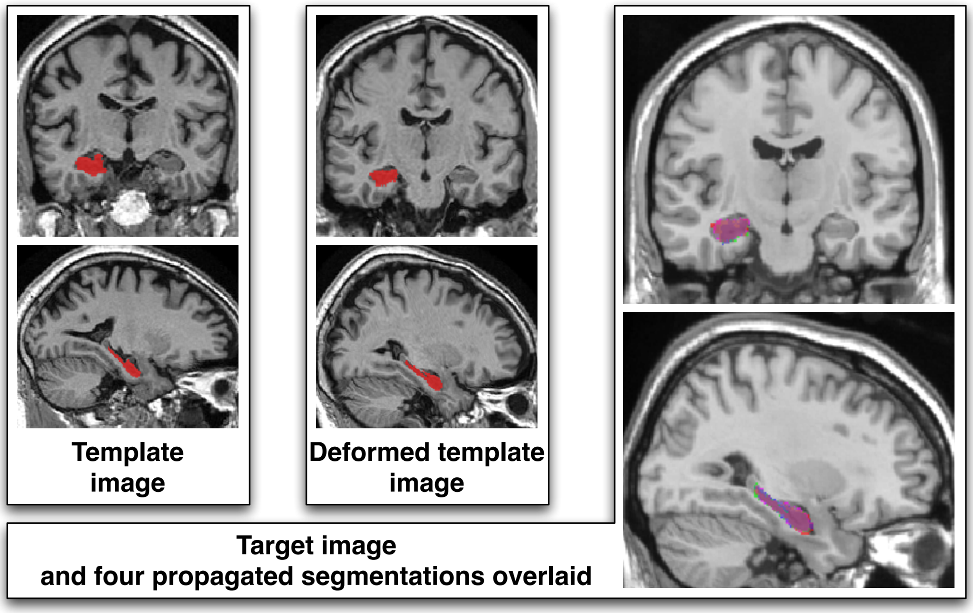


Figure 1.6: Segmentation propagation.

1.2 Medical image registration key challenges

Medical image registration has been and is still an extensive topic of research and a large number of techniques are proposed every year. Registration, however, remains an ill-posed problem [12] and there is no unique algorithm that is suitable for every application. Usually the algorithms are tuned to meet the needs of a specific application. This complexity arises from several registration-specific challenges.

1.2.1 Efficiency

Algorithms have ideally to reach the best accuracy in a minimal amount of time. In practice, one must find a balance suitable for a specific application between accuracy and computation time. For some applications, such as clinical studies, where several patient scans are registered long after the patients have been scanned, registrations can be performed in several hours without dramatically affecting the findings of the studies. However, in other cases, such as image guided-surgery the surgeons require the registrations to be real-time. Indeed it is not feasible to wait hours for registration results while the patient is under general anaesthesia. Figure 1.7 presents an interventional MR case and highlights the differences between an MRI acquired before surgery and another acquired during a brain tumour removal procedure.

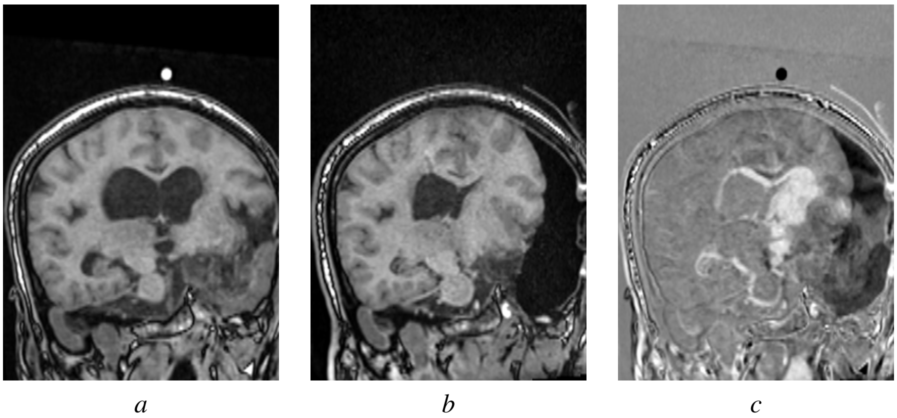


Figure 1.7: Brain surgery. The difference between the pre-operative (a) and post-operative (b) MR images is shown by image (c). Both input images have been globally registered. The difference image highlights the local deformation happening to the brain.

The acquisition techniques are greatly improving nowadays and the resolution of the acquired images increases. More data has thus to be processed and the registration computational requirements are increasing.

In order to improve the quality of the registration, researchers have been developing more complex and thus more computationally expensive techniques.

1.2.2 Topology conservation and invertibility

As explained by Fischer and Modersitzki [12], registration is an ill-posed problem as it aims at finding a warping between two or more images without information about the transformation itself. There is thus no unique solution for a specific case and the penalisation or regularisation of the transformation model is of crucial importance to lead to realistic transformation. However, even if the true solution is unknown, some useful properties can be assumed and can be explicitly taken into account by the registration algorithm. For example, when performing a registration from an image A to an image B, one expects the transformation from B to A to be consistent. Figure 1.8 shows two registration results using the same input images. One registration has been performed by constraining the backward and forward transformations to be consistent; whereas the other has been performed without any constraint. It can be seen that the second registration looks more accurate in terms of visual resemblance; however, the topology has been broken. This is qualified as folding and corresponds to the loss of information.

1.2.3 Different information

Different modalities will highlight different information. Multi-modal registration cases are challenging since images to register do not necessarily share a lot of commonality, as shown by figure 1.2.

1.2.4 Data corruption

Data acquisition techniques are not perfect and thus can contain noise, distortion artefacts or intensity non-uniformity. Figure 1.9 illustrates a non-uniformity bias field with two images acquired the same day on a single patient. These complications may have to be taken into account while performing registration.

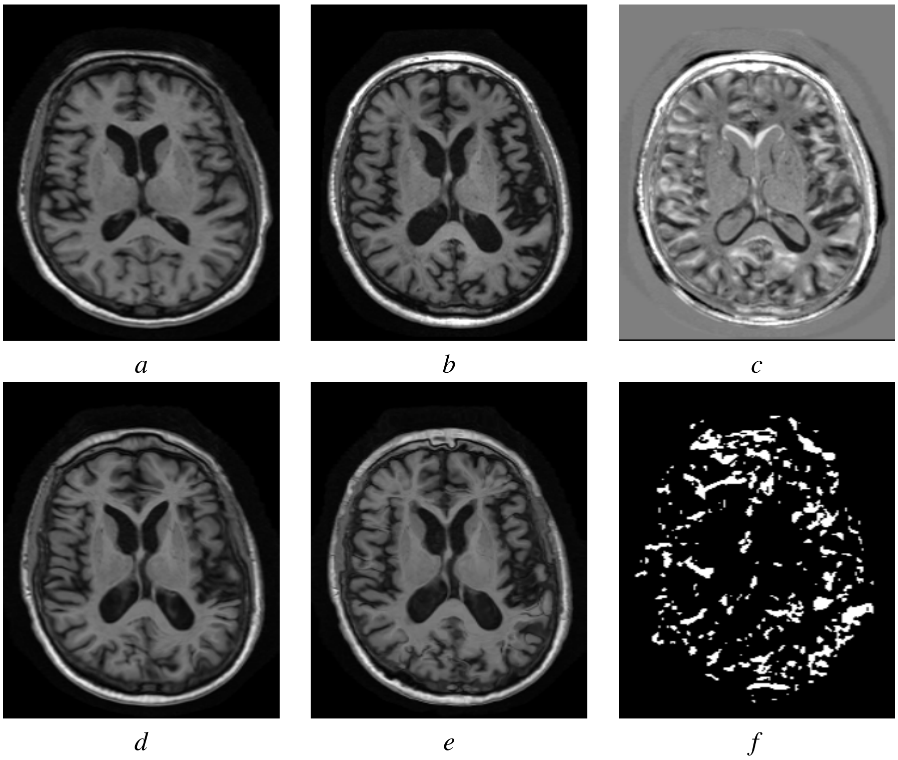


Figure 1.8: Topology conservation. Image *a* has been registered to image *b* using a non-rigid registration framework. Initial differences between the two images are shown by *c*. The registration has been performed twice, once constraining the back- and for-ward transformations to be consistent and once without constraint. Images *d* and *e* show the two result images, with and without constraint respectively. The last image (*f*) shows where the topology has been broken by the second registration.

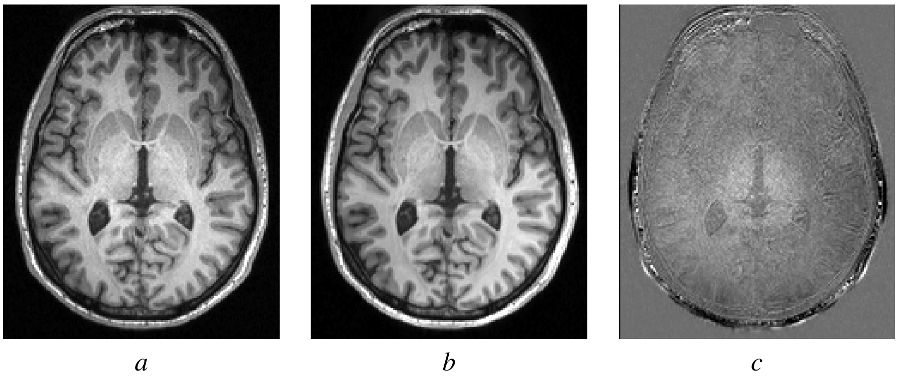


Figure 1.9: Non-uniformity. Images *a* and *b* are T1-weighted MR images from the same patient, acquired the same day but using different headcoils. The difference image (*c*) illustrates mainly the difference of non-uniformity field.

1.2.5 Plausibility

Numerous algorithms for registration have been proposed in the literature [13, 14, 15]. However, they are mostly based on mathematical or physics concepts that do not necessarily reflect the true tissue deformation. All these algorithms recover the deformations differently and thus lead to different warping. As already mentioned, while performing registration, the user has no information about the ground truth deformation. One challenge of intra-patient registration is thus to provide plausible alignment. Figure 1.10 illustrates this by comparing registration results using the same input images but different registration algorithms [1, 2].

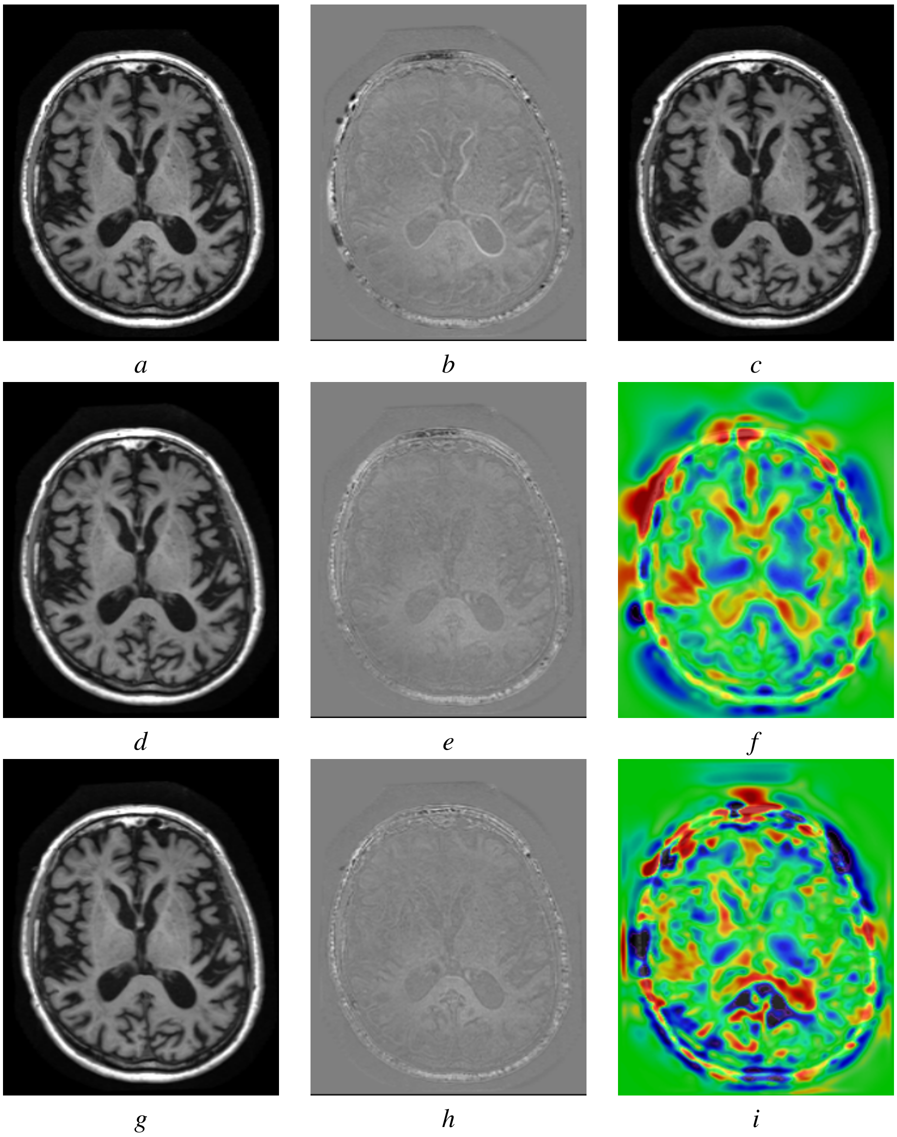


Figure 1.10: A baseline image (*a*) and its followup image (*c*) after one year show some differences (*b*). Two different registration algorithms have been use to align them: FNIRT [1] (images *d*, *e* and *f*) and ANTS [2] (images *g*, *h* and *i*). Result images (*d*,*g*) and difference images (*e*,*h*) are similar; however, the compression/expansion maps (*f*,*i*) testify different deformations. Compression maps are scaled between 50% compression (blue) and 50% expansion (red)

1.2.6 Associativity

When dealing with longitudinal data, one would like the composition of transformations from A to B and B to C to equal the transformation from A to C. For some specific applications, such as cardiac or respiratory motion, the transformation model can be regularised to fit the application. Indeed, in these specific cases, the registration can embed the cyclic nature of the deformation and thus force the registration to be associative. While dealing with other organs, such as the brain in a neurodegenerative context, no cycle can be embedded. This is illustrated in figure 1.11 using three scans from the same patient acquired with a one year gap between each scan.

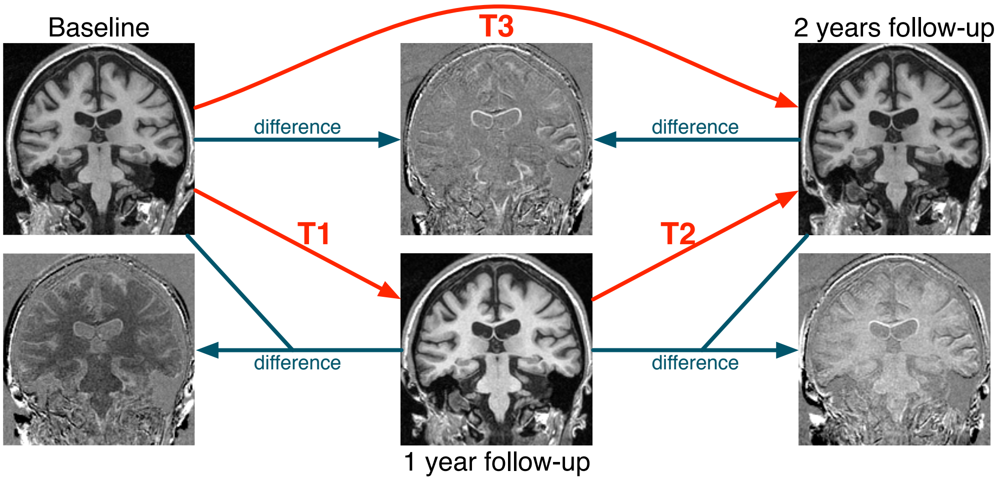


Figure 1.11: Associativity. Three images have been acquired one year apart. Three deformations are represented T1, T2 and T3. One expects the composition of T1 and T2 to be equal to T3.

All these challenges mostly arise in the context of non-rigid registration.

1.3 Thesis contributions

This thesis aims to develop and asses methods to make image registration more efficient and reliable in a variety of clinical applications. The proposed work has mostly been applied to brain MR images [16, 17, 18, 19, 20, 21], but also to mouse embryos for phenotyping purpose [22, 23, 24, 25], lung CT images [26, 27] or colon CT images [28].

As already mentioned, numerous algorithms for non-rigid registration, have been proposed in the last twenty years. Free-form deformation (FFD) [29] is one of them. This algorithm is widely used. Within this thesis is gathered work done to address some of the previously described challenges with respect to the FFD algorithm.

The contributions of this thesis include the following:

- The FFD algorithm has been re-factored in order to lie in a rigorous mathematical framework. This then makes it possible to compute analytical derivative of the objective function. An analytical formulation enabled efficient computation of the objective function partial derivatives thereby improving registration in terms of speed and accuracy.

- Using the proposed framework for the FFD, a volume constraint has been embedded. Combined with a folding correction scheme specifically developed for cubic B-Splines model, topology conservation could then be ensured while preserving efficiency.

- A differential bias field model has been added in the registration framework and is optimised concurrently with the spatial deformation with regards to a common similarity measure. This contribution addresses the problem of non-uniform intensity interfering with registration in MRI.

- In order to enforce a plausible deformation, the FFD's spline deformation model has been combined with a non-linear elastic solver based on a finite element model. This ensured that the recovered deformation is biomechanically plausible for intra-patient longitudinal registration.

- To increase the efficiency of the proposed methods, all the code has been ported to graphics hardware. Taking advantage of graphic processing units (GPU), each method processing time has been decreased by a ratio of 10 to 20. This allows near real-time medical image registration. The developed code has been robustly implemented and has been made available to the community through an open-source distribution.

- The FFD deformation model does not implicitly ensure topology conservation while warping an image. I used a stationary velocity field to parametrize the displacement of each degree of freedom. It allows to ensure diffeomorphism and to obtain the for- and back-ward transformations.

1.4 Thesis organisation

The next chapter, **chapter 2**, is a review of the non-rigid registration methods that have been presented in the literature. This review is focusing on algorithms that use intensity-based similarity measures.

Chapter 3 summarises the registration approaches that have been implemented using graphics hardware.

Chapter 4 describes the formulation and implementation of the fast free-form deformation (F_3D) algorithm. I developed this algorithm [16, 17, 18] in order to address one of the registration challenges: computationally expensive processing. The F_3D method has been validated on different datasets [22, 23, 24, 25, 21]; these validations are also presented in this chapter.

Chapter 5 contains the improvements that have been proposed for F_3D. The first improvement deals with the imperfection of the data to register, specifically in the context of MRI [19] where a different bias field can occur between images to register. The second improvement addresses unrealistic registration by constraining the transformation to be pairwise defined and thus topologically consistent [27].

The methods presented in chapters 4 and 5 have been made freely available and can be downloaded from `http://sourceforge.net/projects/niftyreg`. NiftyReg is a framework for rigid, affine and non-rigid registration I developed as part of work for this thesis.

Chapter 6 presents work that also addresses the problem of unrealistic transformations for longitudinal intra-patient registration. The presented approach [30] takes advantage of a non-linear elastic solver combined with a finite element model to ensure that the obtained transformation is biomechanically plausible.

The last contribution of this thesis is presented in **chapter 7**. It aims at addressing several registration challenges. The fast free-form diffeomorphic deformation (F_3D^2) algorithm [20] is presented. This is a non-parametric non-rigid registration algorithm based on a stationary velocity field. Efficiency and topology conservation are addressed by this implementation but also consistency since the formulation of the algorithm provides both the forward transformation from R to F and the backward transformation from F to R.

Chapter 8 concludes this thesis and outlines the work I would be keen to do in the future.

Chapter 2

Medical Image Non-Rigid Registration, Literature Review

Non-rigid registration consists of locally spatially deforming one or several images to match another image. Prior to any local warping, global transformation parameters have to be evaluated. These global parameters, as seen in chapter 1, consist of translation, rotation, scaling and shearing. They are used to initialise the non-rigid registration. This chapter is a review of the most widely used non-rigid registration algorithms that have been presented. Algorithms have been classified into non-parametric and parametric approaches. In this review, it is assumed that global registration has always been performed and only local warping is addressed.

2.1 Description of a generic algorithm for image registration

Numerous algorithms have been developed to perform registration. They all follow the same principle. First an image is transformed to appear more similar to another, then the similarity between the two images is assessed and lastly the transformation is optimised to maximise the similarity. These three steps are repeated until an optimal transformation is found. All the proposed algorithms can be classified by their transformation model.

2.1.1 Transformation model and resampling

The first transformation model is called rigid transformation. It consists of moving an object in space but never changing its original shape. The object can only be translated or rotated. While dealing with 3 dimensional images, a rigid transformation will be parametrised by 6 degrees of freedom: 3 rotations and 3 translations. These parameters are illustrated by figure 2.1.

The second transformation model is an affine model. The transformation that is applied to the object is global, meaning that every parameter will affect the whole object. An affine transformation consists of translation and rotation parameters, as in rigid registration, but in addition has scaling and shearing parameters. Scaling and shearing parameters are illustrated by figure 2.2.

The third and last transformation model is called non-rigid or local and involves many more degrees of freedom. It consists of applying localised transformation to an object as illustrated in figure 2.3.

Figure 2.4 presents the registration of two MR images. Registrations have been performed using

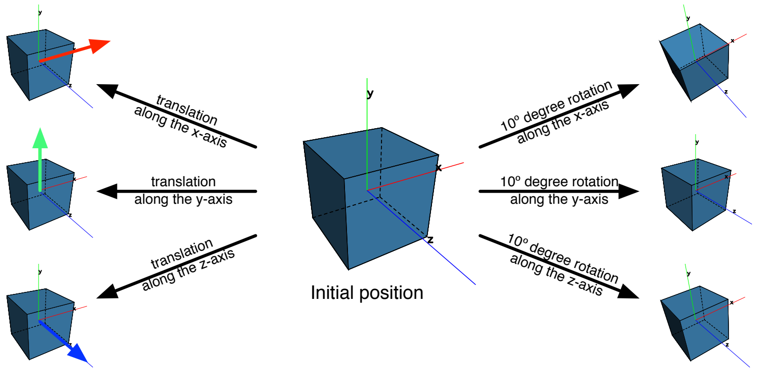

translation along the x-axis
translation along the y-axis
translation along the z-axis
Initial position
10° degree rotation along the x-axis
10° degree rotation along the y-axis
10° degree rotation along the z-axis
x
y
z

Figure 2.1: Rigid transformation parameters applied to a cube.

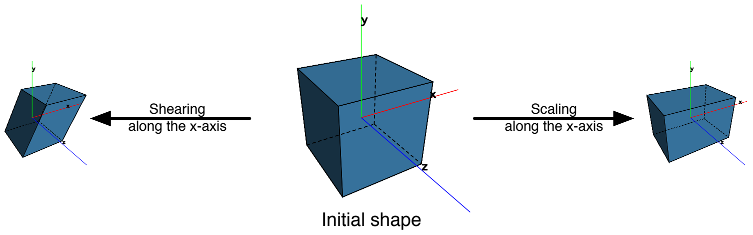


Figure 2.2: Affine transformation parameters applied to a cube. Only the parameters affecting the x-axis are depicted in this figure. Note that the same transformation can be applied along every axis.

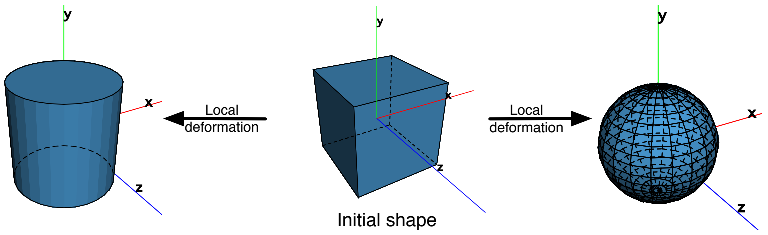


Figure 2.3: Two local, or non-linear, deformations have been applied to the initial shape.

the three previously mentioned transformation models: rigid, affine and non-rigid.

Once a transformation model $\mathbf{T}$ is chosen, it is used to spatially deform a floating image F into a warped image $F(\mathbf{T})$ that is in the space of a reference image R. For each voxel $\vec{x} \in \Omega$, which is the space of the reference image, one is able to know the transformed position $T(\vec{x})$. Each voxel intensity in the $F(\mathbf{T})$ image is resampling from the original F image. This is illustrated in figure 2.5.

2.1.2 Measure of similarity – objective function

The similarity between images R and $F(\mathbf{T})$ is then assessed. The evaluation is performed using a measure of similarity. Different measures have been proposed for image registration. These measures can be classified into two categories: feature- or intensity-based. Feature-based measures required a pre-processing step in order to extract useful features. Features can be for example points, lines, surfaces or volumes. Intensity-based measures do not require any pre-processing step and are computed directly from the voxel intensities.

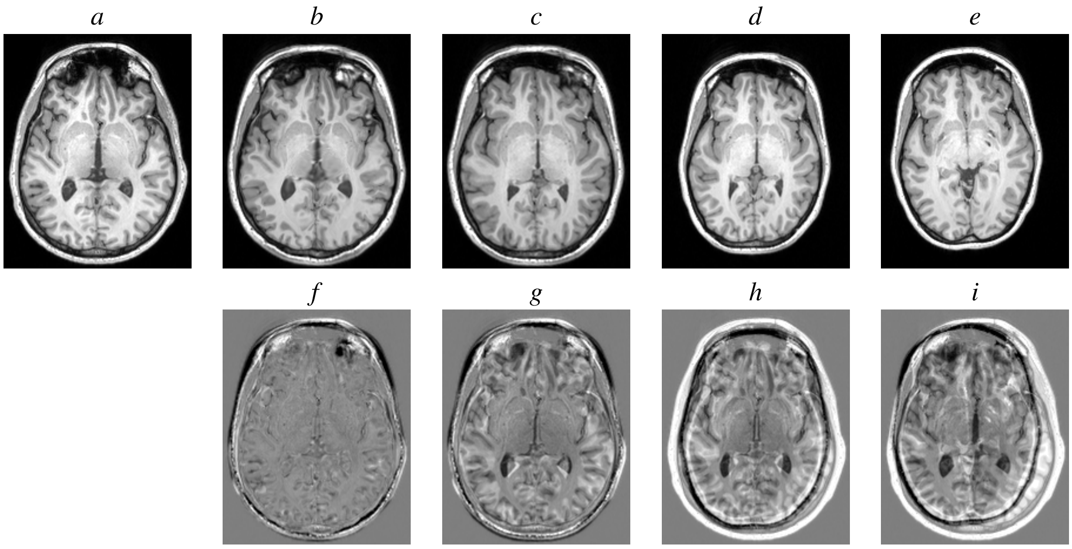


Figure 2.4: Registration example. A floating image (*e*) is registered to a reference image (*a*) using 3 different approaches: rigid (*d*), affine (*c*) and non-rigid (*b*). The bottom row shows the difference images between each image and the reference. Note that only 2 dimensional axial views are presented but the registrations have been performed using volumes.

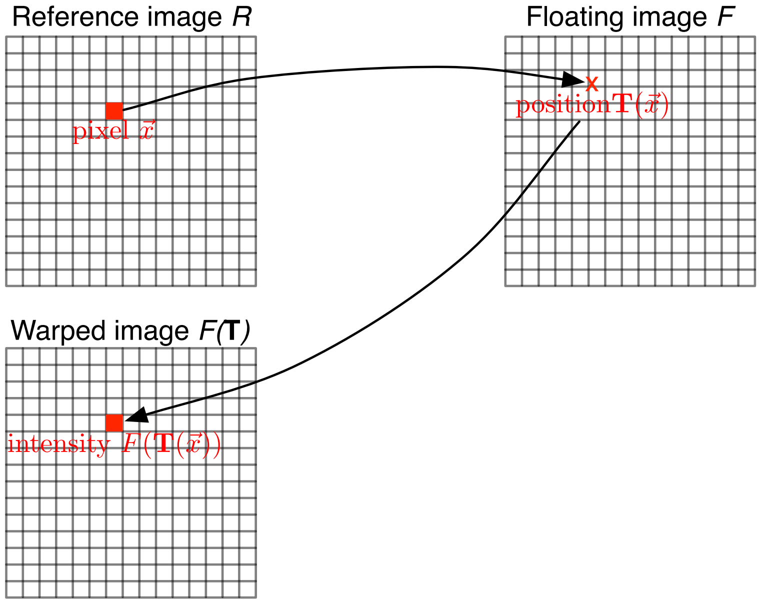

Reference image R
pixel $\vec{x}$
Floating image F
position $\mathbf{T}(\vec{x})$
Warped image $F(\mathbf{T})$
intensity $F(\mathbf{T}(\vec{x}))$

Figure 2.5: Image resampling. The intensities in the floating image are use to compute the warped image intensities.

2.1.3 Optimiser

An optimiser is finally used to maximise the value of the similarity measure by changing the transformation model degrees of freedom. The optimiser schemes that are used in medical image registration are common to other computer science fields. Algorithms can use schemes that do not require any derivative of the objective function such as the simplex algorithm or methods which depend on derivatives of the first or second order.

2.2 Definition

All registration algorithms follow the same approach and are based on three different modules. These modules are: a transformation model, an objective function and an optimisation scheme that can be combined with a regularisation scheme. Their organisation is shown on figure 2.6.

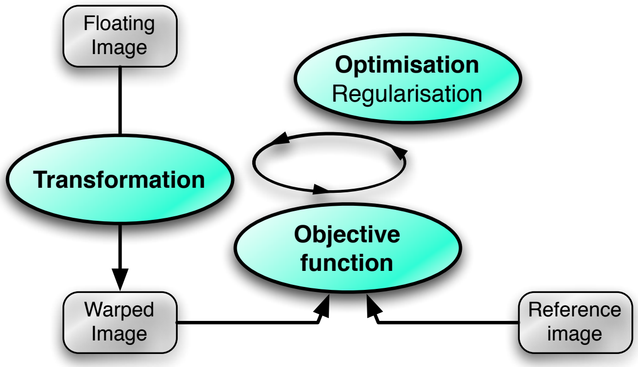


Figure 2.6: Registration module. A transformation model is used to resample a floating image into the space of a reference image. The parameters of the transformation model are optimised in order to maximise an objective function.

2.2.1 Transformation model

The transformation model, noted $\mathbf{T}$, can be either non-parametric or parametric.

Within a non-parametric model, every voxel has three degrees of freedom, which correspond to its translation along the x-, y- and z-axis. The transformation is usually stored as a deformation field, $\mathbf{u}$, where each voxel position has to be optimised: $\vec{x}' = \mathbf{u}(\vec{x})$. While using this model, a regularisation of the transformation is necessary in order to ensure a smooth deformation. Indeed, without regularisation, surrounding voxels have no information about each other and the topology of the object to register is most likely to be broken.

A parametric model takes advantage of another function Fct, a transformation model, to generate a deformation field: $\vec{x}' = \mathbf{u}(\vec{x}) = \text{Fct}(\vec{x})$. The number of degree of freedom is usually lower than the number of voxels when using a non-parametric model. The smoothness of the transformation model promotes an uninterrupted transformation. Nevertheless, penalty terms acting as regulariser are often added to the objective function in order to favour continuous transformation.

In most applications, the transformation has to be bijective meaning that a one-to-one mapping has to exist between both input images. An absence of one-to-one mapping is referred to as folding and is characterised by a loss of material and thus a broken topology.

In many implementations, a pyramidal approach is applied to the input images. This is summarised by figure 2.7. The input images are downscaled in order to: (1) avoid local minima and (2) decrease the registration computation time. It consists of performing the registration on a coarse level first and propagating the deformation to initialise the next finer level.

The transformation parameters are optimised until an optimal value of the objective function is reached.

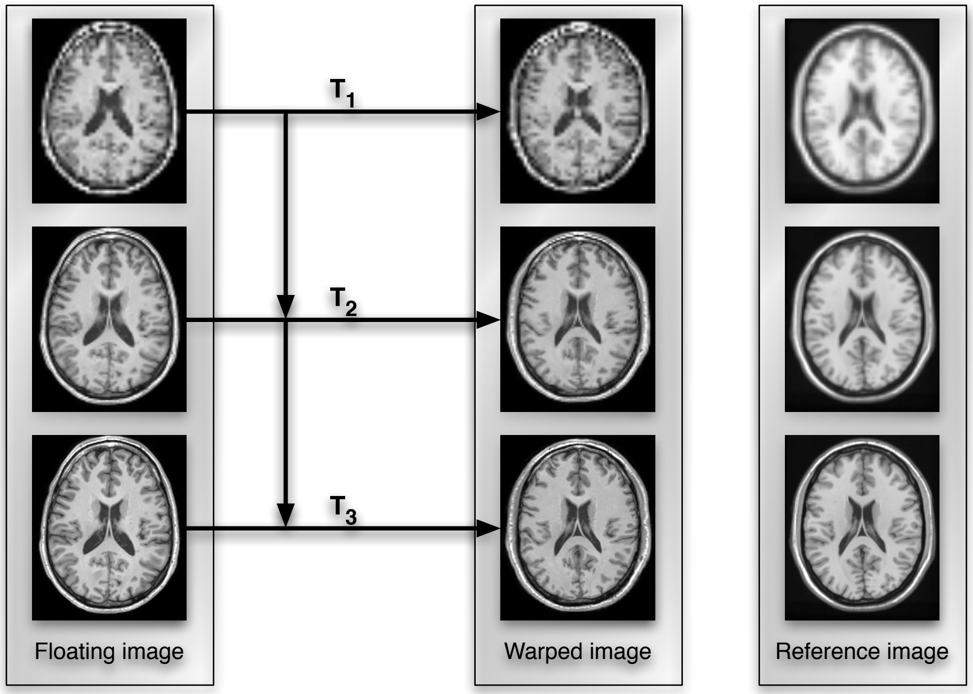


Figure 2.7: Multiresolution illustration. The registration is performed using 3 levels of a pyramidal downsampling.

2.2.2 Measure of similarity

The objective function is composed of a measure of similarity and a penalty term in specific cases.

The measure of similarity is used to assess the quality of the warping between a reference image R and a deformed floating image $F(\mathbf{T})$, where F corresponds to the input floating image. The similarity measure can be based either on features (such as landmarks for example) or on intensities. Intensity-based measures are more widely used since they do not require any pre-processing. I will only review intensity-based methods in this chapter.

The simplest similarity measure consists of searching for a direct relationship between the images' intensities. It can be the sum of squared difference (SSD) or the sum of absolute differences. The SSD between 2 images R and $F(\mathbf{T})$ is computed as:

$$\mathrm{SSD}(R, F(\mathbf{T})) = -\sum_{\vec{x} \in R} |R(\vec{x}) - F(\mathbf{T}(\vec{x}))|^2 , \tag{2.1}$$

where $\vec{x}$ corresponds to a voxel in the reference image domain. Note that a negative sign is used since the SSD is here considered as a measure of similarity and not a measure of dissimilarity.

These measures of similarity require both images to have a one-to-one relationship in terms of intensities. However, while dealing with images from the same object with identical modality but different scanner for example, images can have different intensity distributions. Measures such as normalised cross-correlation (NCC) takes these changes into account. NCC is computed as:

$$\text{NCC}(R, F(\mathbf{T})) = \left| \frac{1}{N-1} \sum_{\vec{x} \in R} \frac{(R(\vec{x}) - \overline{R}) \times (F(\mathbf{T}(\vec{x})) - \overline{F(\mathbf{T})})}{\sigma^R \times \sigma^{F(\mathbf{T})}} \right|, \tag{2.2}$$

where N represents the total number of voxels in R, $\overline{R}$ and $\overline{F(\mathbf{T})}$ represent the average intensity values for the reference and warped images respectively and σ^R and $\sigma^{F(\mathbf{T})}$ represent the variance of their intensity distribution.

SSD and NCC measures are directly based on the intensities of the input images. They are thus not suitable for multi-modal registration when used globally. However, the use of a local NCC using for example small sub-volumes of the images enables the registration of images from different modalities [31, 2].

Some measures based on information theory use the distribution of the voxel intensities and are thus suitable for multi-modal registration. The joint entropy (JE) measure computes the common information between the reference and warped image and maximising this information aims at maximising the alignment [32]. The first requirement for an entropy-based measure is to build a joint histogram $\mathcal{H}$. This joint histogram contains the occurrence of each combination of paired-intensities and is normalised in order to contain paired-distribution probabilities. Figure 2.8 shows 2 joint histograms between various input images. Assuming an image R with intensity $r \in [0\ bin_r[$ and an image $F(\mathbf{T})$ with intensity $f \in [0\ bin_f[$ where bin_r and bin_f are the binning number in each image; $p(r, f)$ represents the probabilities of having an intensity r from one image aligned with an intensity f in the other image. The JE

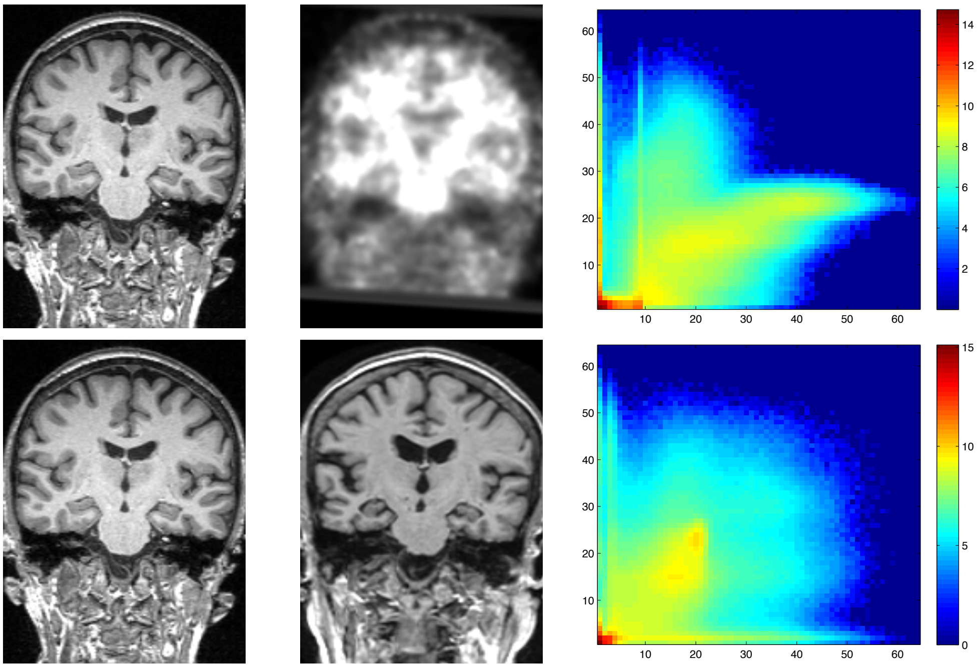


Figure 2.8: Joint histogram. The top row shows the joint histogram of two images from different modalities after global registration. The bottom row shows a histogram between images from the same modality. All image intensities have been normalised between 0 and 63. The joint histograms are displayed using a log-scale.

can then be computed, using Shannon's formula [33] for entropy, as:

$$\mathrm{H}(R, F(\mathbf{T})) = -\sum_{r=0}^{bin_r-1}\sum_{f=0}^{bin_f} p(r,f) \times \log(p(r,f)). \tag{2.3}$$

Several methods have been proposed in order to fill the joint histogram. It can be done for example using a linear interpolation, where the floating image is first resampled and then each occurrence is added as 1 into the histogram. A partial volume approach consists of adding a weight between 0 and 1 in the joint histogram according to the transformation [34]. A Parzen window approach consists of adding weight in the neighbourhood of the voxel intensities [35]. The last step of all these methods is the histogram normalisation in order to obtain probabilities instead of number of occurrences.

The JE suffers from an important drawback when used for medical image registration. Namely, it is possible to optimise the joint entropy by reducing the content of either image. In other words, the measure can be maximised by removing the overlap between both images. To address this issue, Viola and Wells [36] and Maes *et al.* [34] introduced mutual information (MI), which is based on the JE and the marginal entropies. The marginal entropies of the reference (equation 2.4) and warped images (equation 2.5) are computed similarly to the JE, using the joint histogram:

$$\mathrm{H}(R) = -\sum_{r=0}^{bin_r-1} p(r) \times \log(p(r)), \tag{2.4}$$

$$\mathrm{H}(F(\mathbf{T})) = -\sum_{f=0}^{bin_f-1} p(f) \times \log(p(f)), \tag{2.5}$$

where $p(r) = \sum_{f=0}^{bin_f-1} p(r,f)$ and $p(f) = \sum_{r=0}^{bin_r-1} p(r,f)$. Both marginal and the joint entropies are then used to compute the MI:

$$\mathrm{MI}(R, F(\mathbf{T})) = \mathrm{H}(R) + \mathrm{H}(F(\mathbf{T})) - \mathrm{H}(R, F(\mathbf{T})). \tag{2.6}$$

Studholme *et al.* [37] proposed the use of a normalised version of the MI: the normalised mutual information (NMI) that is the ratio of the marginal entropies over the joint entropy.

$$\mathrm{NMI}(R, F(\mathbf{T})) = \frac{\mathrm{H}(R) + \mathrm{H}(F(\mathbf{T}))}{\mathrm{H}(R, F(\mathbf{T}))}. \tag{2.7}$$

They empirically show the NMI to be more robust to overlap than the MI.

None of the entropy-based similarity measures we have discussed so far contain any spatial information about the intensity distribution. Loeckx *et al.* [38] presented a framework using a conditional MI (cMI) where the spatial information is taken into account using a weighting based on the voxel positions. Zhuang *et al.* [39] generalised the framework and proposed the spatial MI (sMI).

Most of the registration algorithms using an entropy-based measure use Shannon's entropy formula; however other formulae can be used and might lead to different results. Different formulae of entropy have been assessed by Cahill [40] for registration purposes.

The list of presented measures of similarity in this section is not exhaustive and I have only covered the most widely-used measures.

2.2.3 Regularisation

As previously mentioned, registration is an ill-posed problem since it aims at finding warping between input images without prior knowledge about the ideal transformation. One may enhance the algorithm in order to generate realistic deformations. A realistic deformation is considered to be a smooth warping, meaning there are no discontinuities in the deformation field and the topology of the deformed object is not broken.

In order to constrain a transformation model to produce realistic deformations, one applies regularisation to limit the search space during the registration process. Transformation regularisation can be referred as explicit and implicit regularisation.

Explicit regularisation can be seen as a direct modulation of the transformation. Some algorithms take advantage of physics-based properties [41, 42], some others use a composition scheme that consists of composition of deformation fields [43] or use more pragmatic methods such as smoothing [44]. Implicit regularisation consists of the addition of a penalty-term to the similarity measure. The penalty term promotes smooth deformation and limits unrealistic deformation. Both formulation, explicit and implicit are however linked and could be interchanged, they mostly depends of the implementation rather than the mathematical formulation of the problem.

2.2.4 Optimisation

Optimisation strategies [45] which are used in medical image registration are not specific to the field. They are commonly used in the field of computer science [46]. Some of the most commonly used are gradient descent [29] or Gauss-Newton like approaches [47]. Due to the high number of parameters to optimise while performing non-rigid registration, first and/or second derivatives of the objective function are required. Indeed, a brute force search strategy would be too time consuming in this setting.

The derivatives of the objective functions can be complicated to evaluate. For example, the derivatives of the SSD are quite simple to compute, whereas the analytical derivatives of the MI are more complicated and time consuming [48, 35]. Some implementations thus take advantage of approximated derivatives [49].

2.3 Intensity-based non-rigid registration algorithms

Researchers have proposed a large amount of approaches for medical image registration. This review focuses on those which are the most well-known and widely used.

2.3.1 Non-parametric registration

As already mentioned, a non-parametric transformation model has its deformation field directly optimised through the registration process. I will here focus on the optical flow, the diffusion-based, the fluid-based methods as well as on the inverse consistent approaches and the methods based on velocity field, stationary and non-stationary.

Optical flow

The optical flow method for image alignment has been presented by Horn and Schunck [50]. In order to optimise the SSD between a reference and a warped image, they used the first derivative of the similarity measure to drive the voxel displacements. However, this approach would lead to a non-smooth deformation since every voxel position could vary independently. A smoothness regulariser is thus added to the SSD in the form of a penalty term: the sum of the Laplacian of the deformation model. The objective function $\mathcal{O}_{OF}$ to optimise is thus:

$$\mathcal{O}_{OF}(R, F(\mathbf{T})) = \alpha \times \text{SSD}(R, F(\mathbf{T})) - \sum_{\vec{x} \in R} \left(\frac{\partial^2 T_x(\vec{x})}{\partial x^2} + \frac{\partial^2 T_y(\vec{x})}{\partial y^2} + \frac{\partial^2 T_z(\vec{x})}{\partial z^2} \right) \tag{2.8}$$

where α is a weighting factor and T_x, T_y and T_z are the three components of the deformation field. The derivative of the SSD, the optical flow, is computed as:

$$\triangledown \text{SSD}(R, F(\mathbf{T})) = 2 \times (R - F(\mathbf{T})) \times \triangledown F(\mathbf{T}) \tag{2.9}$$

The optical flow method has the advantage of being very efficient. However, the method is not very robust and it is heavily dependent on the trade-off parameters.

Demons algorithm

Thirion [44] proposed considering the registration process as a diffusion process [51]. The name of the demons algorithm has been chosen as an analogy to Maxwell's demon. It is based on a diffusion process to regularise the deformation. In this application the demons are forces which transform the floating image. These forces are based on an optical flow between the reference and floating images.

In his original work Thirion [44] proposed different variations of the demons algorithm. Either every voxel is considered as being a demon, or only those on the contours of segmented images. In this report I consider the demons in a non-parametric approach where a force is applied to each voxel.

The demons algorithm is based on the SSD and a normalised optical flow between the two images:

$$\mathbf{b}(\vec{x}) = \frac{(R(\vec{x}) - F(\mathbf{T}(\vec{x}))) \times \nabla R(\mathbf{T}(\vec{x}))}{||\nabla R(\mathbf{T}(\vec{x}))||^2 + \alpha \times (R(\vec{x}) - F(\mathbf{T}(\vec{x})))^2} \tag{2.10}$$

This normalisation prevents instability for small values.

The registration is performed using a gradient descent where the deformation field is updated using the computed force as $\mathbf{u}(\vec{x}) \leftarrow \mathbf{u}(\vec{x}) + \mathbf{b}(\vec{x})$. This process iterates until the stopping criteria is reached. Thirion used two Gaussian smoothing approaches in order to regularise the transformation. The forces are smoothed before the deformation field is updated. This step is beneficial for large deformations but is inappropriate for the independent displacements of single voxels. This smoothing step corresponds to the diffusion process. The second smoothing step consists of, after each update, smoothing the deformation field itself. This corresponds to an elastic regularisation.

In order to guarantee a one-to-one mapping, Vercauteren *et al.* [47] presented a diffeomorphic approach of the demons algorithm. Their implementation is based on the Lie group theory where the deformation field is optimised over a diffeomorphic space. Using the property that the composition of two diffeomorphic transformations is a diffeomorphic transformation, a composition starting from a blank grid will lead to a diffeomorphic transformation if each update is ensured to be a diffeomorphism. Each update is thus derived from the exponentiation of the update field [52]. Composition, usually represented by the $\circ$ operator, consists of applying a deformation field to a second deformation and results in a new field which is applied to the floating image. Figure 2.9 illustrates the difference between an additive update and a composition update step. Moreover Vercauteren *et al.* used a fast computation of vector field exponentials based on a scaling-and-squaring approach [43]. This allows for their method to be very efficient when compared to other techniques.

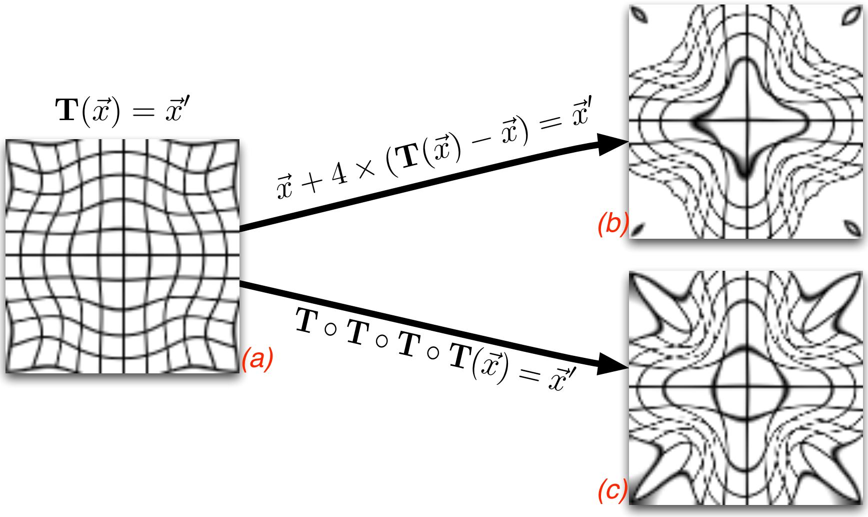

$\mathbf{T}(\vec{x}) = \vec{x}'$
$\vec{x} + 4 \times (\mathbf{T}(\vec{x}) - \vec{x}) = \vec{x}'$
$\mathbf{T} \circ \mathbf{T} \circ \mathbf{T} \circ \mathbf{T}(\vec{x}) = \vec{x}'$
(a)
(b)
(c)

Figure 2.9: Additive and composition update step. An initial transformation is applied to a blank grid *(a)*. The same transformation is then added *(b)* or composed *(c)* to itself four times. It can be seen than the topology of the grid is conserved using a composition step but broken using the additive scheme.

The demons algorithm is efficient compared to other techniques, and the case remains even when the diffeomorphic approach is included. Thanks to the work of Vercauteren *et al.*, the demons algorithm now leads to a one-to-one mapping between both images. This work has also been extended by Yeo *et al.* [53] to perform spherical registration which is of interest for example to align brain cortical layers. The log-demons framework, proposed by Vercauteren *et al.* [54], further expanded the demons algorithm framework to include the optimisation of a single stationary velocity field. The concept of a stationary velocity field is described in the following sections of this chapter. The demons algorithm is based on the SSD and is therefore more sensitive to noise than approaches using, for example, NMI. For the same reason, this method is not suitable for multimodal registration.

Fluid-based algorithms

Fluid-based algorithms use a physical process to regularise forces and convert them into displacements. Several algorithms can be referred to as fluid, the first algorithm for non-rigid registration was proposed by Christensen *et al.* [41, 42]. This algorithm is based on equations to describe the motion of fluids. It models the behaviour of a fluid when stresses are applied. In the case of registration, the volume to deform is considered as a homogeneous fluid to which external forces are applied.

The Navier-Stoke equations are partial derivative equations (PDE) used in fluids mechanics to describe fluid motion. The approximation used for registration is defined as:

$$\mu\vec{\nabla}^2\mathbf{v} + (\mu+\lambda)\,\vec{\nabla}\left(\vec{\nabla}^T\mathbf{v}\right) + \mathbf{b}\left(\mathbf{u}\right) = \vec{0} \tag{2.11}$$

where $\mathbf{v}$ is the velocity field of the deformation field $\mathbf{u}$, ∇ is the gradient operator and ∇^2 its Laplacian. μ and λ are the viscosity coefficients. The body force $\mathbf{b}$ corresponds to the gradient used to drive the deformation. Christensen *et al.* [41, 42] aimed to maximise the SSD between the reference and floating image. As a consequence, the gradient of this similarity measure can be easily computed as in equation 2.9.

Once the similarity measure gradient is known for every voxel, equation 2.11 has to be solved in order to obtain the deformation field $\mathbf{u}$. This step is extremely time consuming. Several approaches have been proposed to solve this equation. In their original work Christensen *et al.* used the Successive Over-relaxation (SOR) method to solve the PDE. Freeborough *et al.* [55] later proposed using the full multigrid [46] method. This method is based on a coarse-to-fine approach and reduces the computation time of the solver. More recently, Cahill [56] took advantage of a Fourier transform to efficiently solve the regularisation step.

In order to guarantee a smooth transformation without any folding in the deformation, a regridding step has been added. It consists of using a blank deformation field to be composed with the current one once the compression at a voxel position is too high. This approach enables to maintain reasonable numerical accuracy while approaching large deformations. The regridding step is added when the compression ratio of at least one voxel drops below a threshold defined by the user, usually between 0.3 and 0.5. The volume change of one voxel can be computed using the determinant of the Jacobian matrix:

$$\det_{Jac}(\vec{x}) = |Jac(\vec{x})| = \begin{vmatrix} \frac{\partial\mathbf{T}_x(\vec{x})}{\partial x} & \frac{\partial\mathbf{T}_x(\vec{x})}{\partial y} & \frac{\partial\mathbf{T}_x(\vec{x})}{\partial z} \\ \frac{\partial\mathbf{T}_y(\vec{x})}{\partial x} & \frac{\partial\mathbf{T}_y(\vec{x})}{\partial y} & \frac{\partial\mathbf{T}_y(\vec{x})}{\partial z} \\ \frac{\partial\mathbf{T}_z(\vec{x})}{\partial x} & \frac{\partial\mathbf{T}_z(\vec{x})}{\partial y} & \frac{\partial\mathbf{T}_z(\vec{x})}{\partial z} \end{vmatrix} \tag{2.12}$$

A Jacobian determinant with values less than 1 corresponding to a compression, a value higher than 1 to an expansion and a value equal to 1 to a volume preservation.

Algorithm 1 and figure 2.10 summarise the fluid algorithm.

The fluid algorithm leads to smooth deformation where the composition steps theoretically guarantee a one-to-one mapping. Similarly to the diffeomorphic demons, it appears that in some cases it still produces a negative Jacobian determinant value. Indeed, as explained by Vercauteren *et al.* [47], the unfolded transformation is only guaranteed if all the steps are kept and composed. However, Christensen *et al.* only records the final transformation.

The fluid approach is computationally expensive. Indeed solving the PDE, 2.11, is very time consuming despite the improvement proposed by Freeborough *et al.* or Cahill. The core of the algorithm is

Algorithm 1 Fluid algorithm scheme

```
if Maximal iteration number is not reached then
    Compute the body forces
    Solve the Navier-Stoke equation
    Compute the new deformation field
    Calculate the Jacobian determinant
    if Jacobian determinant values are below a user-defined threshold then
        Regrid and compose the deformation fields
    end if
end if
Return last deformation field and exit
```

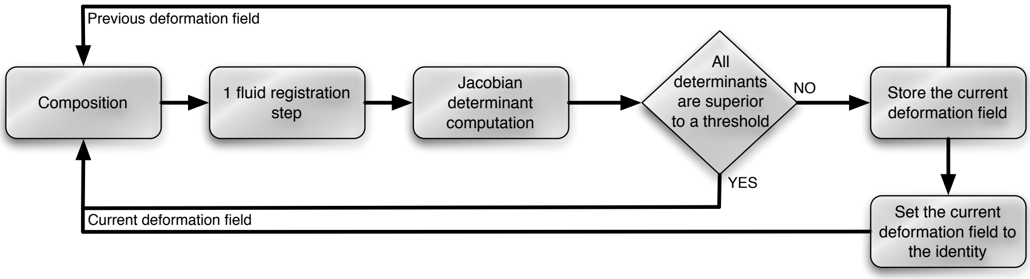

Previous deformation field
Composition
1 fluid registration step
Jacobian determinant computation
All determinants are superior to a threshold
NO
Store the current deformation field
YES
Current deformation field
Set the current deformation field to the identity

Figure 2.10: Fluid algorithm scheme

based on the Navier-Stokes equation to regularise the deformation. As a consequence it is not possible to add an additional regularisation term if the method needs to be modified for a specific application. Viscosity coefficients λ and μ in equation 2.11 largely influence the registration and are user defined. The user would have to be experienced enough not to use too high a value that would lead to poor matching accuracy, and not too small a value that would lead to unconstrained deformations and require more re-gridding steps.

Inverse consistent approaches

In order to ensure a one-to-one mapping between the reference and floating images, Christensen and Johnson [57] proposed an inverse consistent approach. This concept has been initially introduced by Cachier and Ray [58]. The forward transformation $\mathbf{T}$ is optimised concurrently with the backward transformation $\mathbf{T}^{-1}$. The optimisation process takes into account two measures of similarity and an inverse consistency constraint (ICC):

$$\mathcal{O} = \text{SSD}(R, F(\mathbf{T})) + \text{SSD}(R(\mathbf{T}^{-1}), F) + \text{ICC} \tag{2.13}$$

where, in Christensen and Johnson, the ICC is defined using a symmetric form as:

$$\text{ICC} = \sum_{\vec{x}} \|\mathbf{u}(\vec{x}) - \tilde{\mathbf{w}}(\vec{x})\|^2 + \|\mathbf{w}(\vec{x}) - \tilde{\mathbf{u}}(\vec{x})\|^2 \tag{2.14}$$

with $\mathbf{T}(\mathbf{x}) = \vec{x} + \mathbf{u}(\vec{x})$, $\mathbf{T^{-1}}(\mathbf{x}) = \vec{x} + \mathbf{w}(\vec{x})$ and $\tilde{\mathbf{u}}$ and $\tilde{\mathbf{w}}$ are the approximated inverse displacement fields of $\mathbf{w}$ and $\mathbf{u}$ respectively.

Avants *et al.* [2], in the symmetric normalisation algorithm (SyN), used a similar approach where they estimated a middle point between the reference and the floating image. Instead of concurrently registering the reference image to the floating space and the floating image to the reference space, they registered both images to a middle space such that $F(\mathbf{T}(0.5)\circ\mathbf{G}^{-1}(0.5))(\vec{x}) = R(\vec{x})$ where $\mathbf{T}$ represents the forward transformation and $\mathbf{G}$ the backward. Figure 2.11 summarises the different transformations.

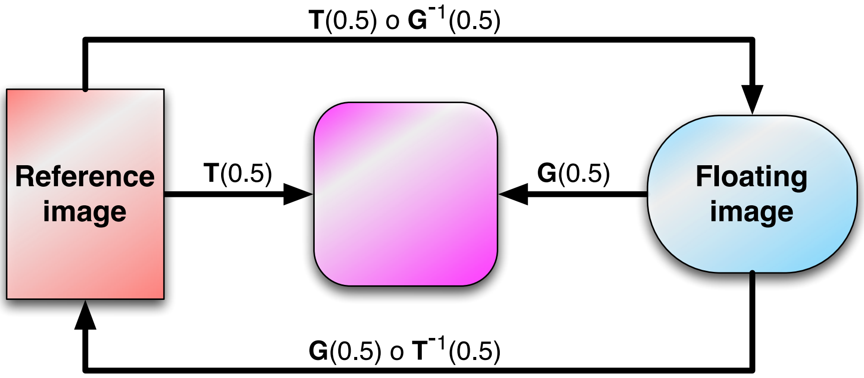


Figure 2.11: Inverse consistency scheme as defined by Avants *et al.*. A middle image is generated from the reference and source images.

Non-stationary velocity field

Beg *et al.* [59] proposed integrating a velocity field over time to generate a deformation field. The framework they proposed is entitled large deformations diffeomorphic metric mapping (LDDMM). In practice the time-varying velocity field is discretised into *n* steps that are then composed to generate the final forward deformation field as well as the backward transformation deformation field. Figure 2.12 illustrates the framework using 5 steps.

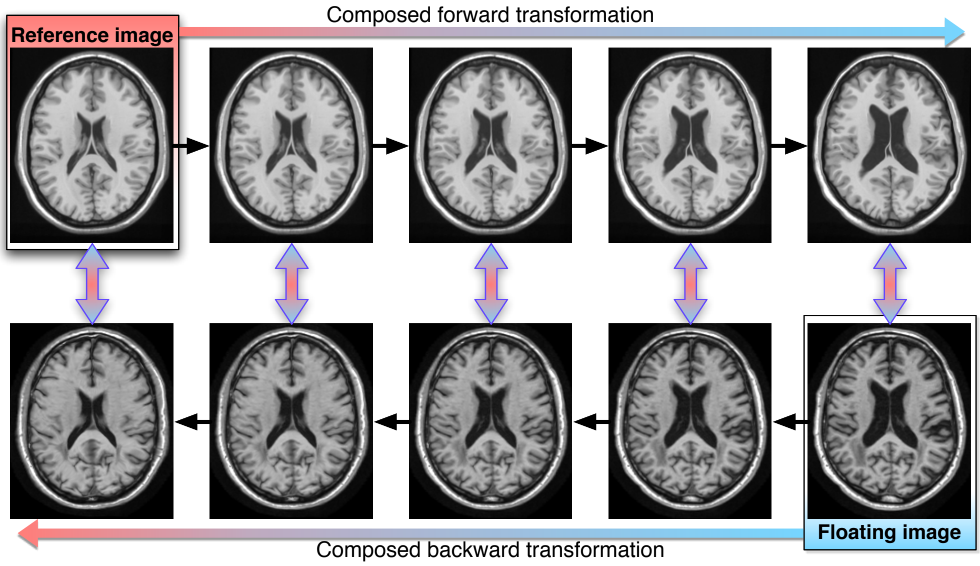

Composed forward transformation
Reference image
Floating image
Composed backward transformation

Figure 2.12: A non-stationary velocity field is integrated over 5 steps to generate a geodesic flow of diffeomorphism.

The deformation regularisation is performed in a Hilbert space using kernels of different sizes. The wider the kernel, the coarser the registration is. Risser *et al.* [60] proposed to use a coarse-to-fine approach on the kernel sizes in order to obtain more accurate registration and remove the heavy dependence on a single kernel size.

Stationary velocity field

Ashburner [3] and Hernandez *et al.* [61, 62] simultaneously proposed to take advantage of a single stationary velocity field to generate a deformation field. This concept has been initially proposed by Arsigny *et al.* Arsigny2006 but without its use in a registration algorithm.

The Diffeomorphic Anatomical Registration Through Exponentiated Lie Algebra, or DARTEL, algorithm was proposed by Ashburner in 2007 [3]. As the diffeomorphic version of the demons, this approach is based on the Lie Group theory to guarantee diffeomorphism. DARTEL also has the added benefit of providing a constant inverse transformation. Figure 2.13 shows that the composition of one deformation field by its inverse results in a blank grid.

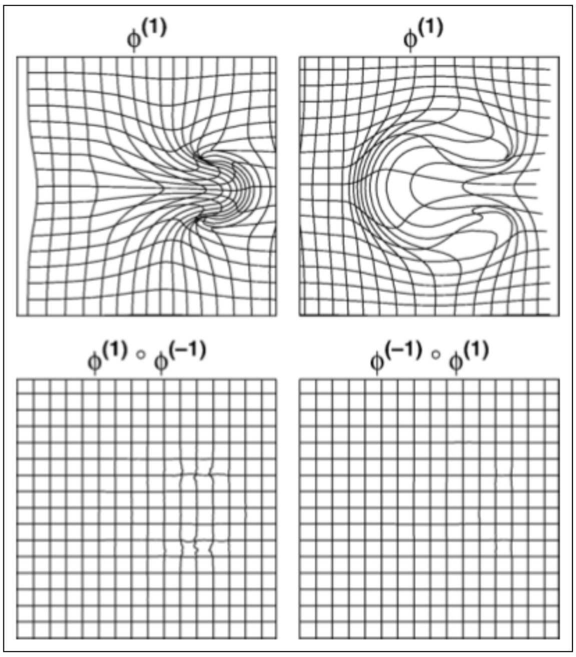

$\phi^{(1)}$
$\phi^{(1)}$
$\phi^{(1)} \circ \phi^{(-1)}$
$\phi^{(-1)} \circ \phi^{(1)}$

Figure 2.13: Inversion and composition of a diffeomorphic deformation field using DARTEL. This figure is extracted from [3]

The deformation starts from an identity deformation field which is integrated over time using the velocity field. The flow remains constant through time. The integration is done using an Euler approach $x_{t+1} = x_t + h\frac{\partial x_t}{\partial t}$ where the velocity field is divided into n identical steps. For example, if $n = 8$, the final deformation field $\vec{u}^{(8/8)}$ is computed using:

$$\begin{aligned}
\vec{u}^{(1/8)} &= \text{Identity} + \vec{v}/8 \\
\vec{u}^{(2/8)} &= \vec{u}^{(1/8)} \circ \vec{u}^{(1/8)} \\
\vec{u}^{(3/8)} &= \vec{u}^{(1/8)} \circ \vec{u}^{(2/8)} \\
\ldots &= \ldots \\
\vec{u}^{(8/8)} &= \vec{u}^{(1/8)} \circ \vec{u}^{(7/8)}
\end{aligned}$$

The small time step produces an accurate solution where the voxel trajectory, from the floating to the reference image, follows a manifold.

More than 8 steps are usually used to guarantee a smooth deformation. To improve the computation time it has been shown that if the number of steps is a power of two, a scaling squaring approach [43] can be employed.

As illustrated by figure 2.13, it is possible to use the same principle to determine the consistent inverse transformation which warps the reference image in the floating referential. Again using 8 steps and the scaling squaring method, the inverse transformation is computed as:

$$\begin{aligned}
\vec{u}^{(-1/8)} &= \text{Identity} - \vec{v}/8 \\
\vec{u}^{(-2/8)} &= \vec{u}^{(-1/8)} \circ \vec{u}^{(-1/8)} \\
\vec{u}^{(-4/8)} &= \vec{u}^{(-1/4)} \circ \vec{u}^{(-1/4)} \\
\vec{u}^{(-1)} &= \vec{u}^{(-1/2)} \circ \vec{u}^{(-1/2)}
\end{aligned}$$

The velocity field $\vec{v}$ is solved using the Levenberg-Marquardt algorithm [46]. This method iteratively minimises a cost function using the first and second derivatives of the function. It also includes the inversion of large matrices where a full multigrid approach [46] is used in order to decrease the computation time. The cost function includes a similarity term, the SSD and a penalty term. There are several penalty terms which have been implemented within DARTEL (membrane energy, bending energy or linear elastic energy).

The use of DARTEL in SPM[1] (Matlab[2] toolbox) is not based on the SSD but on a particular metric. This metric is based on a grey and white matter probabilistic segmentation of the brain. The advantage of this approach is that it is less sensitive to noise, such as bias, however it is dependent on the quality of segmentation.

The Hernandez *et al.* [61, 62] approach uses the same basic concept and only differs in the objective function. Their cost function optimises the backward as well as the forward transformation, using a symmetric approach.

2.3.2 Parametric registration

Parametric registration algorithms see their spatial transformation parametrised using a deformation model. Several models have been proposed. We will here present three different methods using three different models. The first model takes advantage of a discrete cosine function to model low-frequency transformation. The second algorithm uses an elastic model based on a Gaussian function and the third model uses an interpolation based on a cubic B-spline to define a C^2 continuous transformation.

Spatial Normalisation Using Basis Functions

The SPM toolbox is a Matlab package purposely designed for brain image analysis. SPM currently includes different registration algorithms including Spatial Normalisation using Basis Functions. The

[1] http://www.fil.ion.ucl.ac.uk/spm

[2] http://www.mathworks.com

basis functions that are used are discrete cosine transforms (DCT).

Spatial Normalisation using Basis functions [63] uses a linear combination of low-frequency periodic basis functions such as those presented in figure 2.14. The algorithm aims to decrease the SSD between the reference and warped images.

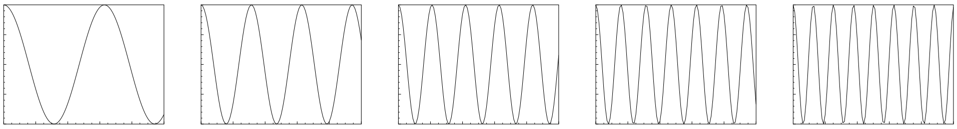


Figure 2.14: Discrete cosine transform. Different frequencies can capture different deformations.

Only low frequency periodic basis functions are considered in this approach and consequently only large deformations are tracked. However, folding in the warping can still happen by generating unnecessary displacements which increase the similarity measure. To avoid this issue, penalisation terms have been added within the framework: membrane energy, bending energy or linear-elastic energy.

The membrane energy and the linear-elastic energy are computed using the first derivatives of the transformation at every voxel position.

The bending energy (BE), based on the second derivatives of the transformation, is computed as

$$\begin{aligned}\text{BE} \quad = \quad & \sum_{\vec{x}\in R}\left(\left(\frac{\partial^2\mathbf{T}(\vec{x})}{\partial x^2}\right)^2+\left(\frac{\partial^2\mathbf{T}(\vec{x})}{\partial y^2}\right)^2+\left(\frac{\partial^2\mathbf{T}(\vec{x})}{\partial z^2}\right)^2+\right. \qquad (2.15)\\ & \left. 2\times\left[\left(\frac{\partial^2\mathbf{T}(\vec{x})}{\partial xy}\right)^2+\left(\frac{\partial^2\mathbf{T}(\vec{x})}{\partial yz}\right)^2+\left(\frac{\partial^2\mathbf{T}(\vec{x})}{\partial xz}\right)^2\right]\right).\end{aligned}$$

The penalty terms are weighted against the similarity measure using some trade-off constants. If their values are too large the regularisation becomes too significant and the deformation will not be captured by the algorithm. On the contrary, factor values which are too small will produce a deformation field which over-transforms the floating image in order to optimise the similarity measure.

The computation time of the algorithm is quite reasonable as there are only a few hundred parameters to optimise. The non-rigid registration of a 3D MRI brain image takes approximatively five minutes. Low frequency basis functions are however unable to track high frequency deformations and, as a consequence, this approach is not as accurate as others which include more degrees of freedom.

The Hierarchical Attribute Matching Mechanism for Elastic Registration algorithm

The HAMMER algorithm was proposed by Shen and Davatzikos [64]. This approach has been developed to add anatomical information to the registration process and is based on an elastic deformation model.

Attribute vectors are the principal novelty in this algorithm. For each voxel an attribute vector can be estimated. This vector contains information about the underlying tissues, the voxel information and geometric moment invariant (GMI). The GMI provides information about the spherical neighbourhood of a voxel, this information is invariant to rotation and depends on the radius of the sphere taken into account. To obtain information about the tissue properties, both images have to be segmented into grey matter (GM), white matter (WM) and cerebro-spinal fluid (CSF). A set of vectors is then chosen to drive the deformation. In order to improve the speed of the algorithm, not all of the voxels are considered as driving voxels. The preferred selection of voxels are boundary voxels between tissues.

The objective function to optimise in this framework is based on two similarity measures and one penalty term: the BE (equation 2.15). The first measure is based on the transformation from the reference to the floating image whereas the second is obtained from the inverse transformation. This is similar to the approach based on inverse-consistency proposed by Christensen and Johnson [57].

Once all the vectors have been calculated, for each driven voxel in the floating image, the most similar attribute vector in its neighbourhood is detected. The displacement applied to a driven voxel goes from its current position to the position of its most similar voxel in the second image. The same approach is applied to obtain the inverse driven movement going from the reference image to the floating image. To propagate the displacement of the driven voxel to others a smoothing approach is used after which the deformation field is refined using the bending energy term. This loop iterates until convergence or until it reaches the maximal number of iterations.

The HAMMER approach uses anatomical knowledge to warp one image to another. This is done using a tissue segmentation, from which the extra information provided is added to the voxel intensities.

The use of segmentation can introduce a bias into the method as the method itself is largely dependent on the quality of this pre-step. Additionally, the number of parameters which have to be manually set is significant and can have a great impact on the registration quality.

Free-Form Deformation

The free-form deformation (FFD) algorithm was presented by Rueckert *et al.* [29]. The deformation model is based on a cubic B-Spline interpolation which guarantees C^2 continuous deformation. The objective function to optimise is a balance between the NMI and a smoothness penalty term. Originally a steepest gradient ascent was used as an optimiser.

The cubic B-Spline deformation is a powerful tool which is used, for example, to model 3-D objects and was also used in the medical field for tracking and motion analysis in cardiac images [65] before Rueckert *et al.* in 1999.

An underlying mesh of control points is applied to an image and when a control point of this lattice moves all the voxels in the neighbourhood move as well. The distance between two control points is called the spacing, noted as δ. The number of control points is defined by the image size and the spacing used along each axis. The new position of each voxel is computed by using the position of the closest control points.

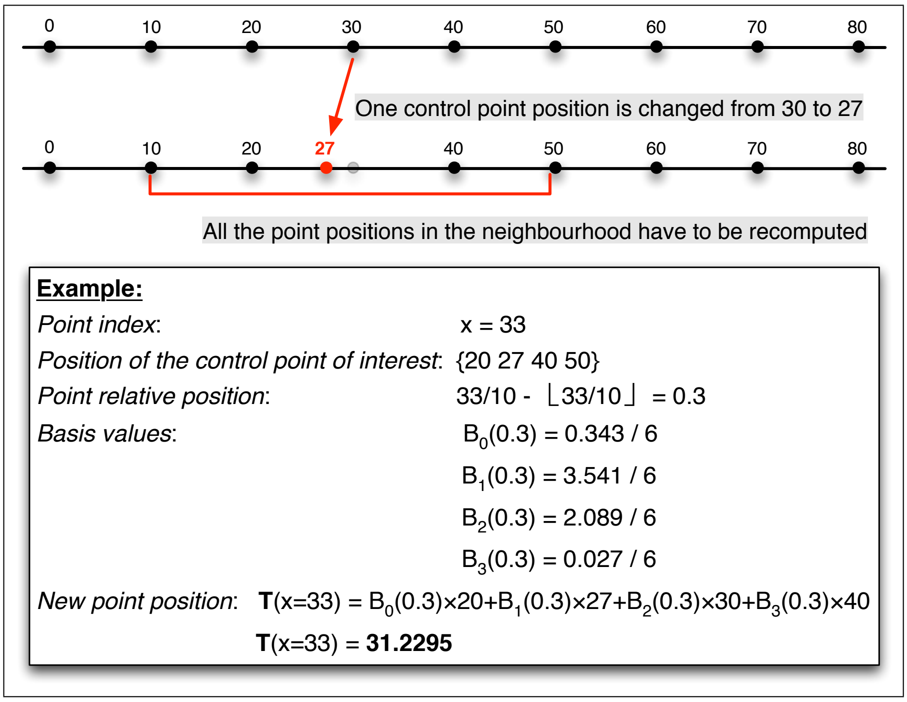
In one dimension the formula to compute the new coordinate of one point is:

$$\mathbf{T}\left(\vec{x}\right) = \sum_{l=0}^{3} B_l \left(\frac{\vec{x}}{\delta} - \lfloor \frac{\vec{x}}{\delta} \rfloor \right) \mu_{i+l} \tag{2.16}$$

μ_i is the first control point position taken into account. To compute a point position in one dimension four control points are used, two before the indexed point and two after it. The functions B_0 to B_3 are the approximated third-order spline polynomials, also called basis functions.

$$\begin{aligned} B_0(u) &= (1-u)^3/6 \\ B_1(u) &= \left(3u^3-6u^2+4\right)/6 \\ B_2(u) &= \left(-3u^3+3u^2+3u+4\right)/6 \\ B_3(u) &= u^3/6 \end{aligned} \tag{2.17}$$

Figure 2.15 shows an example of B-Spline interpolation in one dimension.


0
10
20
30
40
50
60
70
80
One control point position is changed from 30 to 27
0
10
20
27
40
50
60
70
80
All the point positions in the neighbourhood have to be recomputed

Example:

Point index: x = 33

Position of the control point of interest: {20 27 40 50}

Point relative position: $33/10 - \lfloor 33/10 \rfloor = 0.3$

Basis values: $B_0(0.3) = 0.343 / 6$

$B_1(0.3) = 3.541 / 6$

$B_2(0.3) = 2.089 / 6$

$B_3(0.3) = 0.027 / 6$

New point position: $\mathbf{T}(x{=}33) = B_0(0.3)\times 20 + B_1(0.3)\times 27 + B_2(0.3)\times 30 + B_3(0.3)\times 40$

$\mathbf{T}(x{=}33) = \mathbf{31.2295}$

Figure 2.15: Cubic B-Spline parametrisation in one dimension

In three dimensions the FFD can be written as the 3-D tensor product of the 1-D cubic B-Spline (equation 2.16):

$$\mathbf{T}(\vec{x}) = \sum_{l=0}^{3}\sum_{m=0}^{3}\sum_{n=0}^{3} B_l(u)\, B_m(v)\, B_n(w)\, \mu_{i+l,j+m,k+n} \tag{2.18}$$

where u, v and w are the relative positions of the index point along each axis. i, j and k are the indices of the first control point taken into account along the x, y and z -axis respectively. In that case the new coordinates are computed from the $4 \times 4 \times 4$ surrounding control points.

Cubic B-Splines have local influence; if one control point is moved only the voxels in its surrounding area will be affected, the area of which is $(4\delta_x)(4\delta_y)(4\delta_z)$ voxels size centred on that control point.

The time required to compute all new coordinates is high. We have seen previously that the position of sixty-four control points are needed to compute only one coordinate. Each voxel has three coordinates, one along each axis, therefore for one voxel 192 degrees of freedom have to be consulted. If the spacing is about twenty voxels wide along each axis the number of voxels affected by one control point displacement is $(4 \times 20)^3$, i.e. 512,000 voxels. Therefore, when one control point is moved, $512,000 \times 192 = 98,304,000$ voxel positions have to be calculated. The number of parameters required to complete a full transformation of one image is a function of the number of control points. For example for a 256^3 image with a 20 millimetres uniform spacing, the number of control points is 2197 and as there are 3 coordinates for each control point the number of parameters to optimise according to the cost function is 6591.

As stated previously the number of control points depends on the spacing and as a consequence the registration quality is better when the number of control points is higher. However, starting with a very small spacing would be computationally too expensive. Indeed it could be faster to move each point close to its optimal position with a smaller number of parameters, made possible by using a multi-resolution approach. This technique consists of starting with a large spacing, for example 20 millimetres, and then decreasing the spacing length once or several times. The B-Spline properties allow for this approach, indeed an arbitrary cubic B-Splines deformation can be refined to an identical deformation by adding a new control point at each half-spacing length [66]. Using this technique the spacing can be divided by two along each axis and the number of control points is increased by a factor of eight (2^3). This procedure allows control points to be close to their optimal position and therefore less iterations are required during the final stage. Moreover, it reduces the chances of finding a local minimum that would not lead to an accurate registration.

To enforce the deformation to be smooth, Rueckert *et al.* used a smoothness penalty by adding a biharmonic model penalty term. This is based on the bending energy (2.15) of a thin plate of metal that is subject to bending deformations. It is composed of second-order derivatives of the B-Spline deformation which can be computed analytically from the B-Spline basis functions (2.17) due to the C^2 continuity of the model [67]. For example the second-order derivatives of the deformation along the x axis (2.19) and along the x and y axis (2.20) are:

$$\frac{\partial^2 T}{\delta x^2} = \frac{1}{\delta_x^2} \sum_{l=0}^{3} \sum_{m=0}^{3} \sum_{n=0}^{3} \frac{d^2 B_l(u)}{du^2} B_m(v) B_n(w) \mu_{i+l,j+m,k+n} \tag{2.19}$$

$$\frac{\partial^2 T}{\delta xy} = \frac{1}{\delta_x \delta_y} \sum_{l=0}^{3} \sum_{m=0}^{3} \sum_{n=0}^{3} \frac{dB_l(u)}{du} \frac{dB_m(v)}{dv} B_n(w) \mu_{i+l,j+m,k+n} \tag{2.20}$$

The first (2.21) and second order (2.22) derivatives from the B-Spline basis functions are respectively:

$$\begin{aligned} dB_0(u)/du &= \left(-u^2 + 2u - 1\right)/2 \\ dB_1(u)/du &= \left(3u^2 - 4u\right)/2 \\ dB_2(u)/du &= \left(-3u^2 + 2u + 1\right)/2 \\ dB_3(u)/du &= u^2/2 \end{aligned} \tag{2.21}$$

$$
\begin{aligned}
d^2B_0(u)/du^2 &= -u+1 \\
d^2B_1(u)/du^2 &= 3u-2 \\
d^2B_2(u)/du^2 &= -3u+1 \\
d^2B_3(u)/du^2 &= u
\end{aligned}
\tag{2.22}
$$

Rueckert *et al.* found experimentally that a weighting of 1% for the penalty term is a good compromise between the optimisation of the NMI and the penalty term. His findings are based on registration of contrast-enhanced breast MR images.

$$
\mathcal{O}(R, F(\mathbf{T})) = 99\% \times \text{NMI} - 1\% \times \text{BE} \tag{2.23}
$$

The smoothness penalty term used here does not strictly guarantee a one-to-one mapping between the reference and floating image. Other approaches have been developed to emphasise the penalty term influence. For example, a constraint on a strictly positive determinant of the Jacobian can be embedded into the optimisation scheme [68]. Rohlfing *et al.* [69] used a penalty term based on the Jacobian determinant scalar at every voxel to ensure volume preservation while performing registration.

More recent work by Rueckert *et al.* [70] guaranteed a one-to-one mapping using a composition scheme. The approach is similar to those described for non-parametric models; however, instead of composing several deformation fields, lattices of control points are considered. In order to ensure one-to-one mapping within a single lattice of control points, Rueckert *et al.* used a hard constraint on the grid displacement. Choi and Lee [71] have shown that if all the control point displacements are within the range of $[-0.4\delta; 0.4\delta]$ the lattice will not generate any folding. Rueckert *et al.* used a new lattice of control points each time the displacement values exceed the cited range. Figure 2.16 summarises this approach. Similarly, De Craene *et al.* [72, 73] employed several lattices of control points corresponding to different time points of a cardiac cycle. These lattices are then composed to obtain the complete transformation.

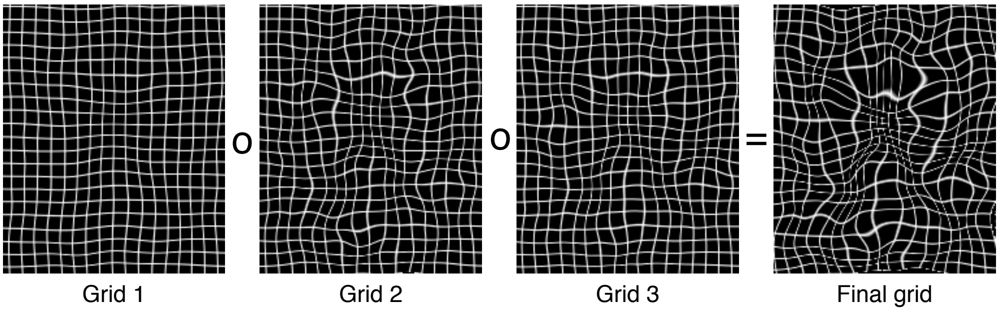


Figure 2.16: Different grids of control points at various resolutions can be composed to generate a final deformation field.

The cubic B-spline interpolation as well as the NMI optimisation are computationally expensive tasks. For this reason, Schnabel *et al.* [74] proposed to use a non-uniform grid of control points where only control points of interest are considered as active and their positions optimised.

2.4 Conclusion

Many algorithms have been proposed for medical image non-rigid registration. However, there is no algorithm that ideally suits every application. All algorithms have their advantages and drawbacks. Some algorithms, such as the fluid, SyN or LDDM embed nice properties enforcing diffeomorphism or transformation following geodesic paths. This is done at the expense of increased computation time that make the algorithms unsuitable for near-real time requirement. On the contrary, algorithms such as the optical flow, the demons of the DCT perform registration in the order of minutes but either with lower accuracy or by breaking the input image topologies. Such algorithms are thus not suitable for applications that require sub-voxel accuracy or a one-to-one mapping.

Within this chapter, I only described algorithms using intensity-based similarity measure. There are numerous other algorithms that use landmarks such as points [75], lines [76] or surfaces [77]. Those algorithms were initially more popular than intensity-based registration but are nowadays less commonly used.

Chapter 3

Medical image registration using Graphics Processing Units (with CUDA)

Medical image registration is a computationally expensive task. It could thus hugely benefit from multi-core architectures. Indeed, assuming an algorithm is parallel, its computation can be spread over a large number of processors. The effect is a computation time decrease while the algorithm complexity is maintained. Graphics processing units (GPUs) have a large number of core that can be use to speed-up computationally expensive algorithms. Graphical cards were originally design for efficient visual rendering for applications such as games. With their important computational power to cost ratio, they became popular for general purpose computation. Five years ago, general purpose GPU- (GPGPU-) implementations were performed through the graphical pipeline using for example OpenGL [1], DirectX [2] or Cg [3]. Vendors such as ATI [4] (now AMD) or NVidia [5] created application programming interface (API) specifically for GPGPU. NVidia released, in late 2006, an API called compute unified device architecture, CUDA [4], that enables programmers to utilise GPUs for general purpose computation. It only requires basic knowledge of the C language and little understanding of the hardware. Other GPGPU API have been released such as Brook [6] or CTM (Close-to-Metal) [7] but we will in this chapter focus on CUDA as it appears to be the more popular in the medical imaging community.

Medical image registration algorithms are suitable for GPU-based implementation as long as their computation is parallel friendly. The current chapter is a brief presentation of GPU-based implementation for non-rigid registration. I refer the reader to review papers [78, 79] that include global registration for further details.

3.1 CUDA presentation

CUDA, compute unified device architecture, is an extension of the C language which facilitates the integration of CPU- and GPU-based code. CUDA enables data transfer and communication between

[1] http://www.opengl.org/

[2] http://msdn.microsoft.com/en-us/directx

[3] http://developer.nvidia.com/page/cg_main.html

[4] http://ati.amd.com

[5] http://www.nvidia.com

[6] http://graphics.stanford.edu/projects/brookgpu

[7] http://sourceforge.net/projects/amdctm

CPU				GPU			
Name	Model	Date	GFlops	Name	Model	Date	GFLOPS
Intel core 2 duo	E8600	2006	26	GeForce	8800GTX	2006	518*
Intel i7	965	2008	51	GeForce	9800GTX	2008	648*
Intel i7	860	2009	46	GeForce	285GTX	2009	1062*
Intel i7	980X	2010	107	Tesla	C2050	2010	1288*/515

Table 3.1: Example of CPU and GPU hardware released since 2006. GFLOPS (FLoating OPerations per Second) entries correspond to double precision operations except where indicated otherwise with a * (single precision).

the CPU and the GPU, namely the host and the device. A function that is run on the device, is called a kernel. Each execution of a kernel is called a thread. Parallelism is achieved by running several threads concurrently. GPU implementation falls under the Single Instruction Multiple Threads (SIMT) derived from the Single Instruction, Multiple Data (SIMD) class of parallel computing. Each concurrent thread exploit data level parallelism by performing the same operation on independent data simultaneously. In order to spread the computation of the threads amongst the multi-processors, CUDA uses a grid and block architecture. A grid contains several blocks that are indexed using blockID and each block contains several threads indexed using ThreadID. BlockID and threadID allow to segment the data to be processed using single indexing. This indexing could be done using 1D, 2D or 3D indexing which makes it practical for 2D or 3D image registration. A block runs on a single multi-processor making possible the sharing of information between threads within the same block.

3.1.1 GPU trend

Since 2005 the maximal CPU clock rate stayed around 3.4 GHz. One way to increase the overall computational power has thus been to increase the number of CPU cores. Modern desktop computers have between 2 and 8 CPU cores. Graphical cards have processors that ran slower than current CPUs but have however significantly more cores. A GeForce 9400, which is a laptop chipset has for example 16 cores and a Tesla C2070 card has 448 cores. Their processor clock rates are 1.10 GHz and 1.15 GHz respectively.

Table 3.1 presents the rate of FLoating OPeration per Seconds (FLOPS) for different hardware, CPU and GPU, released between 2006 and 2010. Figure 3.1 shows the CPU and GPU trend from 2001 to 2011.

3.1.2 Hardware

A graphical card has different types of memory. They are illustrated on figure 3.2. They all have advantages and drawbacks and one must consider their different characteristics while implementing a GPU-based algorithm. Indeed, efficiency of a GPU-based implementation is highly dependent of the memory accesses within a thread.

- The *global memory* can be both read and written from the CPU and be accessed by any GPU core for computation purposes. Since both the CPU and the GPU can access this memory, the global memory is used for data transfer. It is the largest memory on the graphical card, up to 6 GB on the most advanced card and generally around 1 GB on a day-to-day use card. The drawback of the

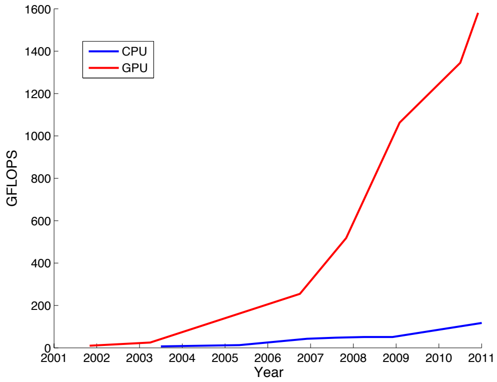


Figure 3.1: Floating operations per seconds for different CPU and GPU hardware between 2001 and 2011. Figure derived from http://www.wikipedia.com

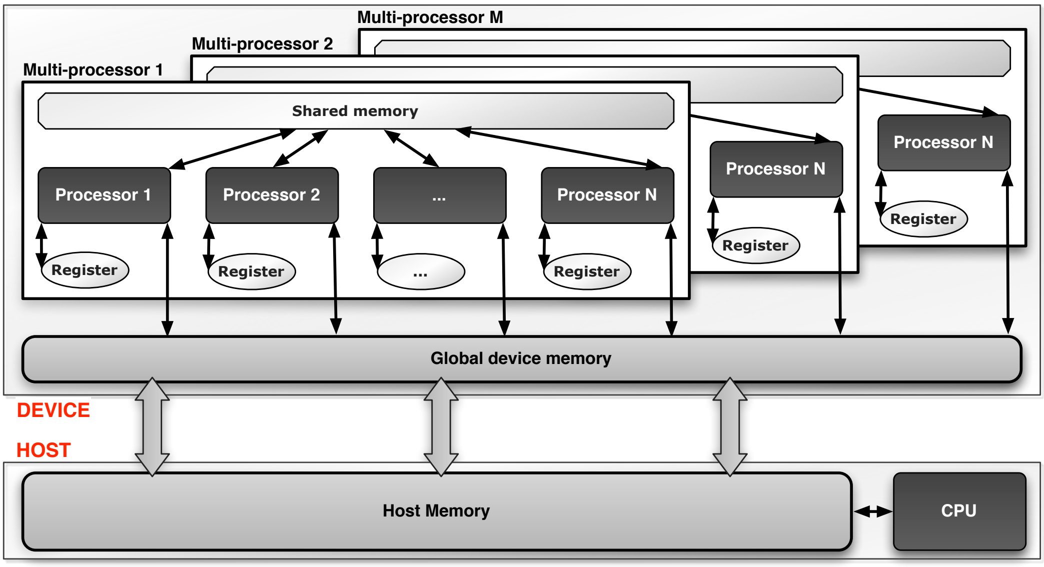


Figure 3.2: Communication between device and host is operated through the device global memory. The global memory can be read and written from every core whereas the shared memory can only be accessed from processors in the same multi-processor. Each processor has its own register memory. This diagram is inspired by [4].

memory type is its speed access, from 300 to 600 cycles per read. This latency can however be hidden using coalesced memory access. In other words, if all the threads in a block access adjacent memory addresses it yields to greater efficiency of the kernels.

- The *texture memory* corresponds to global memory that has caching facilities. Texture memory is read-only but its reading speed can be faster than the global memory. It can indeed provide large performance gains when threads read areas that are spatially close. Moreover, texture enables efficient hardware linear interpolation using 1D, 2D or 3D texture.

- The *shared memory* is limited in size, 16 kB, and in accessibility. Each multi-processor has its own shared memory meaning that only threads that are run on the processor (same CUDA block) can access, read and write. The memory access is however faster (2 cycles per read) than global memory but it can only be accessed from the GPU. In order to be used, the data has first to be transferred from the global memory during a thread execution. The shared memory is divided into banks that correspond to memory blocks of 4 B and one has to avoid bank conflicts. Bank conflicts consist of several threads accessing the same bank; the access becomes serialised and the efficiency decreased.

- The *register memory* is read and write access. This memory is only accessible from a single thread and its amount per thread corresponds to 8 kB divided by the number of concurrent threads running on a multi-processor. The number of threads that can be within a block is defined by the amount of register memory required for the execution of a single thread. If a thread require more memory than available, the global memory will be used instead and the efficiency of the implementation will be significantly decreased.
- The *constant memory* consists of 64kB for the whole graphical card. This memory is read-only from the GPU and is cached during its first access (4 cycles).

3.1.3 Generic GPU-based registration implementation

A basic registration algorithm consists of the following steps:

- The input images are loaded onto the host memory and then transferred to the device global memory.
- The processing is performed on the device while trying to minimise the transfer between host and device.
- The results, transformation parameters and warped image, are transferred back to the host memory and saved.

Since the transfer of data between host and device is a time consuming task, one must minimise the amount of transferred data.

3.2 GPU-based non-rigid registration review

This section tailors the three main parts of any registration algorithm: the transformation model, the measure of similarity and the optimisation scheme. Each subsection presents implementations that have been proposed in the literature. Most of the recent works use CUDA but this section is not limited only to the NVidia API.

3.2.1 Transformation

Optical flow and derivates

Optical flow algorithms were the first to be implemented using graphic hardware. This is due to their suitability to parallel implementation. Strzodka *et al.* implemented a 2D [80] optical flow using Direct X 9 API. They were able to register 2D images in a few seconds using a gradient descent to minimise the SSD between images. Using OpenGL, Vetter *et al.* [81] and Fan *et al.* [82] implemented an optical flow algorithm that optimised the mutual information and a Kullback-Leibler distance between the intensity distributions. Sharp *et al.* [83] and Courty *et al.* [84] implemented the demons algorithm using Cg and Brook respectively. The speed-up of the later was mainly coming from the implementation of recursive filtering and the use of a flat 2D texture, where a 3D volume was mapped into a 2D texture. Samant *et al.* [85] ported the demons algorithm to the CUDA API. Later, Gu *et al.* [86] implemented several demons algorithm variations; these variations were mostly related to the measure of similarity and the gradient forces which are used. They were all modifications of the original optical flow.

Diffusion- and fluid-based algorithms

Diffusion- and fluid-based algorithms are more complex to implement on graphical hardware due to the difficulty of solving partial derivative equation in a parallel-friendly fashion. Noe *et al.* [87, 88] implemented a viscous-fluid algorithm for non-rigid registration using CUDA.

Parametric algorithms

Relatively little work has been proposed using the cubic B-Spline as a transformation model. However, Soza *et al.* [89] and Li *et al.* [90] used meshes and linear interpolation or Bezier curves to parametrise the deformation field. To optimise their transformation parameters they applied Powell's method and Levenberg-Marquardt scheme respectively. Plishker *et al.* [91] implemented a spline algorithm driven by the SSD using multiple platform including single and multiple CPU as well as single and multiple GPU. Teßmann *et al.* [92] presented an implementation of the free-form deformation implemented using OpenGL and optimised through a Powell's method. Ruitjers *et al.* [93] implemented the cubic B-Spline interpolation scheme using CUDA and combined it with the SSD or the cross-correlation as a similarity measure. They applied a quasi-Newton method for optimisation.

Lapeer *et al.* [94] implemented a non-rigid registration algorithm based on radial basis function and normalised mutual information. They reported speed-up ratio against serial CPU implementation between 2 and 10 depending of the number of degrees of freedom (up to 44).

Joldes *et al.* [95] used a finite element model in order to recover brain shift in a surgical context. The finite element model was solved using a non-linear elastic model. Their approach did not follow the classical registration optimisation approach since boundary conditions were extracted from ultra-sound images and then applied to the model.

3.2.2 Measure of similarity

The SSD has been used in most of the proposed GPU-based implementation. Its computation and its gradient computation are easily parallelised since there is no shared information between different spatial locations. A classic implementation of the SSD consists of two steps: the first is the computation, for every voxel of the squared intensity difference (each computation usually corresponds to one thread); the second step is a summation step of all the squared differences typically using a reduction approach.

Entropy-based measures such as the mutual information are more challenging to implement in parallel since a joint histogram has to be generated. Whilst filling the histogram, several pairs of voxels could have the same intensities and thus several threads could try to write within the same memory address. In order to compute the joint histogram using GPU-based implementation, a refactoring of the algorithm is requested to make it parallel-friendly. Several approaches have been developed to implement parallel-friendly approaches on the GPU. Fluck *et al* [96] and Scheuermann *et al.* [97] presented implementation of histogram generation. They scattered the computation and then gathered several independent sub-histograms. This approach is suitable for limited number of bins but is less efficient in the case of a joint histogram when the number of bins is significantly higher (typically 64^2). Shams *et al.* [98], using CUDA, used a sort-and-count approach to build their joint histogram. This approach has the advantage of being write-conflict free and handles histograms with large bin number.

All the proposed implementations on GPUs use a linear interpolation scheme to fill their joint-histogram. This approach is not optimal for medical image registration since it does not allow for computation of the analytical derivatives of the similarity measure. The optimisation task is thus more time consuming.

3.2.3 Optimisation

As stated by Shams *et al.* [78] and Fluck *et al.* [79], the optimisation scheme is not a computational burden in medical image processing. There are two main approaches that can be used: gradient-free method (eg. simplex, brute force) and gradient-dependant method (eg. gradient descent, quasi-Newton). Due to the large number of degrees of freedom, most GPU-based implementations use gradient-dependant method. The optimisation part is only limited after the objective function derivatives have been computed.

3.3 Discussion

In this chapter, I discussed a number of GPU-based implementations summarised in table 3.2; nevertheless, other platforms have been used for high-performance medical image registration. Jiang *et al.* [99] took advantage of FPGA (field-programmable gate array) chips and applied it to the free-form deformation algorithm. Rohlfing *et al.* [100] also implemented the free-form deformation algorithm but using a shared memory multiprocessor environment. Rohrer *et al.* [101] used a Cell (Cell broadband engine architecture) platform to efficiently compute deformation field based on the cubic B-Spline scheme.

GPUs have however some advantages over these three architectures. FPGA chips have the advantage of being cheap, although they are often restricted to a single application since they have to be programmed for one specific task. Shared memory multi-processor computers are not available in every environment due to their prohibitive price. Moreover, they would require large volume of data transfer to perform efficient registration since the shared memory multi-processor computer cannot be stored ”on-site”, eg. in the operative room. Cell platform is also not very popular since they are considered as challenging for software development.

Graphical cards are already in everyday computer. They are relatively cheap compared to other architectures. They can be used ”on-site” and no data transfer is required. Their popularity increased with the release of the CUDA API and a simple interface for software development. However, GPUs have also their drawbacks but they tend to disappear nowadays. NVidia GPUs combined with CUDA

	Year	Transformation model	Similarity of measure	Reported speed-up	Language
Soza *et al.* [89]	2002	Bezier's function	MI	$*^1$	OpenGL
Strzodka *et al.* [80]	2003	Optical flow	SSD	$*^2$	DirectX 9
Vetter *et al.* [81]	2007	Optical flow	MI / KL$*^3$	×5	OpenGL
Sharp *et al.* [83]	2007	Demon	SSD	×70	Brook
Fan *et al.* [82]	2008	Optical flow	MI / KL$*^3$	×18	DirectX 10
Courty *et al.* [84]	2008	Demons	SSD	×10	Cg
Samant *et al.* [85]	2008	Demons	SSD	×55	CUDA
Noe *et al.* [87, 88]	2008	Viscous-fluid	SSD	×25	OpenGL
Li *et al.* [90]	2008	Bezier's function	SSD	×5	OpenGL
Plishker *et al.* [91]	2008	Cubic B-Splines	SSD	×20	CUDA
Teßmann *et al.* [92]	2008	Cubic B-Splines	NMI	×10	OpenGL
Gu *et al.* [86]	2010	Demons	SSD	×100	CUDA
Lapeer *et al.* [94]	2010	RBF$*^4$	NMI	×10	–
Ruijters *et al.* [93]	2011	Cubic B-Splines	SSD/CC	×50	CUDA

Table 3.2: Summary of the GPU-based non-rigid registration algorithms. All the reported figure for speed-up against CPU-based implementation are extracted from the author's publications. They have thus been evaluated at different times, on different hardware and by different programmers.
$*^1$ 6/7 minutes to register brain images of $256 \times 256 \times 112$ voxels with $9 \times 9 \times 9$ degrees of freedom and using an NVidia GeForce3 64MB.
$*^2$ 2/4 seconds to register a 256^2 pixels image using an NVidia GeForceFX 5800.
$*^3$ Kullback-Leibler distance.
$*^4$ Radial Basis Function

did not allow computation using double precision, the newly release graphical card have a proportion of double precision enabled chips. The CUDA API is only suitable for NVidia products, as a result it is impossible to use other vendor, such as AMD hardware. Once again, things are changing with the apparition of OpenCL [8] managed by the technology consortium Khronos Group. OpenCL has the advantage of being vendor independent but it also supports CPUs.

Medical image registration is a time consuming task that requires high performance computing to maintain efficiency. In the last few years, GPU-based implementations have shown to be suitable for this task. Indeed, they ally efficiency, portability and relatively low cost.

[8] www.khronos.org/opencl

Chapter 4

Fast Free-Form Deformation: F$_3$D, the basics

While a huge amount of research has been devoted to the methodological development [13, 14] for non-rigid registration, very little research has focused on the computational burden of the proposed algorithms. One of the most widely used algorithms, Free-Form Deformation [29] (FFD), has not reached its full clinical utility as a result; FFD's computation time on a single data set can extend to several hours. If such constraints could be removed, or alleviated a new range of clinical applications, which require real-time or near real-time computation could be attempted. Such applications arise, for instance, in the context of real-time image-guided surgery: new patient information acquired during surgery, such as ultra-sound images, could be used efficiently to update a previously developed surgical plan.

The bottleneck of the FFD algorithm is the cubic B-Spline computation, and consequently work has been done to speed up this part using various architectures. Jiang *et al.* [99] used a FPGA-based implementation which lead to a speed-up of 3.2 times compared to a 2.666 GHz CPU execution. Rohlfing *et al.* [69] reduced computation time by more than 50 times using 64 CPUs of a shared-memory supercomputer. More recently, Rohrer *et al.* [101] presented a multicore implementation of the B-Spline computation based on a Cell Broadband Engine^{TM} (Cell/B.E.) platform. Their architecture performed 40% faster than serial execution on a standard computer.

These techniques provide considerable computation time improvements, however they require either hardware-specific technical knowledge or hardware with prices that inhibits wide adoption. I proposed the use of graphics processing units (GPUs) as a cost effective high performance solution. Moreover I advocate use of NVidia Corporation's CUDA API, which is an extension of the C-language and requires minimal hardware specific knowledge. A further advantage of GPU execution is the simplification of data confidentiality issues compared with cluster-based computation, since data may be processed entirely on a user's own machine. Anonymisation procedures are thus avoided.

In this chapter I present a parallel-friendly formulation of the algorithm suitable for Graphics Processing Unit (GPU) execution [18]. The first section of this chapter presents the method as well as it's implementation on GPU. In the second section, some of the validations and applications that have been performing with the method are presented.

4.1 Method and implementation

The main requirement for an algorithm to benefit from GPU execution is data parallelism. As described in the previous chapter, the FFD algorithm comprises three components, which may be considered independently: transformation of the floating image using uniform cubic B-splines and an interpolation function; evaluation of an objective function; and optimisation against this function. Individually, these components may be formulated in a data parallel manner as they mainly consist of voxel-wise computations. However difficulties associated with GPU memory constraints entail that certain aspects of the algorithm needs redesign for an efficient implementation on the GPU.

4.1.1 Deformation model

The FFD algorithm consists of locally deforming an image volume using cubic B-Splines. This technique has the desirable feature of guaranteeing a C^2 continuous deformation (see Fig. 4.1). The cubic B-Splines framework has been detailed in chapter 2 and is omitted here. However note that a particularly favourable property of the framework is that any deformation produced with a grid of density n can be exactly produced on a grid of density $2n-1$. This property has been used in a pyramidal approach in the implementation.

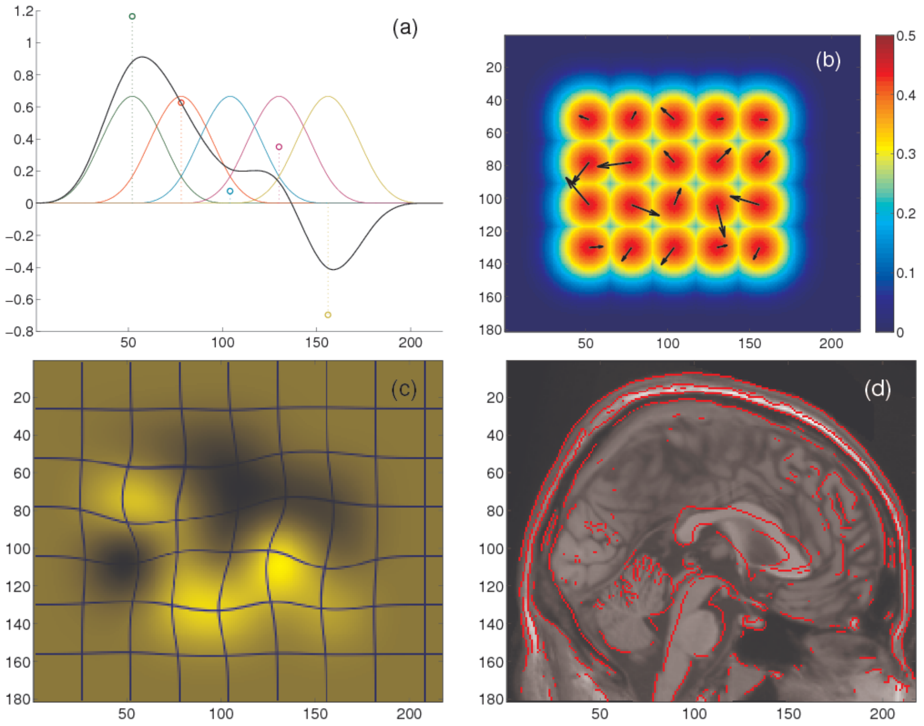


Figure 4.1: From splines to image warps. (a) A weighted sum of uniformly spaced cubic B-spline basis functions used to construct a C^2 continuous curve in one dimension. (b) The previous five basis functions are combined with another four to generate a two-dimensional tensor product; two weighted sums of these 2D basis functions are used to model the x and y components of a displacement vector field. (c) The x displacement field in yellow has been used to deform a regular grid, overlaid in blue. (d) The same transformation illustrated using a brain image: the red edges from the original MRI are overlaid on a grayscale image of the warped result.

The cubic B-Spline method is the bottleneck of the algorithm. For this reason in the *classical* approach [29] only one control point was considered at a time to compute the objective function derivative, which means the whole image does not have to be fully interpolated at each step. The computation of each voxel's position and their new intensities are fully independent and thus their computation is suitable for parallel implementation. Since GPU-based computation is more efficient when processing large amounts of data concurrently, I optimise all control points and interpolate the whole image at each step.

The deformation $\mathbf{T}$ which optimises an objective function between the deformed floating image $F(\mathbf{T})$ and the reference R is sought.

4.1.2 Objective function

The Normalised Mutual Information (NMI) is a voxel intensity-based information-theoretic similarity measure based on the paired-intensity distribution in R and $F(\mathbf{T})$. A larger NMI value reflects a greater level of shared information between the two images. It is computed from

$$\text{NMI} = \frac{H(R) + H(F(\mathbf{T}))}{H(R, F(\mathbf{T}))}, \tag{4.1}$$

where $H(R)$, $H(F(\mathbf{T}))$ and $H(R, F(\mathbf{T}))$ are respectively the two marginal entropies and the joint entropy. Its computation thus requires a joint histogram which, in my implementation, was filled using a Parzen Window (PW) approach [48, 35].

In order to promote smooth deformation, a penalty term $\mathcal{P}_{BE}$ has been added to the NMI value. The objective function $\mathcal{O}(R, F; \{\vec{\mu}\})$ to be optimised is a balance between the NMI similarity measure and the deformation penalty:

$$\mathcal{O}(R, F; \{\vec{\mu}\}) = \alpha \times \text{NMI} - \beta \times \mathcal{P}_{BE}, \tag{4.2}$$

where α and β are set to sum to one. The penalty-term I describe here, the bending-energy, was used for non-rigid registration by Rueckert *et al.* [29]. It is defined as

$$\begin{aligned} \mathcal{P}_{BE} &= \frac{1}{N} \sum_{\vec{x} \in \Omega} \left(\frac{\partial^2 \mathbf{T}(\vec{x})}{\partial x^2} \right)^2 + \left(\frac{\partial^2 \mathbf{T}(\vec{x})}{\partial y^2} \right)^2 + \left(\frac{\partial^2 \mathbf{T}(\vec{x})}{\partial z^2} \right)^2 \\ &+ 2 \times \left[\left(\frac{\partial^2 \mathbf{T}(\vec{x})}{\partial xy} \right)^2 + \left(\frac{\partial^2 \mathbf{T}(\vec{x})}{\partial yz} \right)^2 + \left(\frac{\partial^2 \mathbf{T}(\vec{x})}{\partial xz} \right)^2 \right], \end{aligned} \tag{4.3}$$

where N is the voxel number in Ω, the domain of R. I approximated this penalty term by computing the bending-energy values at the control point positions only, which reduced the number of computations. Furthermore, as explained in Rohlfing *et al.* [69], this approach allowed precomputation of the cubic B-spline basis values for each node, thus easing the calculation further.

4.1.3 Optimization

I used a conjugate gradient ascent scheme to optimise the control point positions. This approach is more efficient than a simpler steepest ascent optimisation, and is less memory intensive than Newton type algorithms. Moreover, it has the advantage to be parallel-friendly. It thus required the derivative

$\partial\mathcal{O}(R, F; \{\vec{\mu}\})/\partial\mu^{\xi}_{ijk}$ of the objective function:

$$\frac{\partial\mathcal{O}(R, F; \{\vec{\mu}\})}{\partial\mu^{\xi}_{ijk}} = \alpha \times \frac{\partial \text{NMI}}{\partial\mu^{\xi}_{ijk}} - \beta \times \frac{\partial\mathcal{P}_{BE}}{\partial\mu^{\xi}_{ijk}}, \tag{4.4}$$

where ξ are the x, y and z components of the control point μ_{ijk}.

The gradient of the NMI is calculated as:

$$\frac{\partial \text{NMI}}{\partial\mu^{\xi}_{ijk}} = \frac{\frac{\partial H(R)}{\partial\mu^{\xi}_{ijk}} + \frac{\partial H(F(\mathbf{T}))}{\partial\mu^{\xi}_{ijk}} - \text{NMI} \times \frac{\partial H(R,F(\mathbf{T}))}{\partial\mu^{\xi}_{ijk}}}{H(R, F(\mathbf{T}))}, \tag{4.5}$$

which requires computation of the derivative of the marginal and joint entropies. The Shannon entropy formulation is used and the derivatives can be computed as:

$$\frac{\partial H(R)}{\partial\mu^{\xi}_{ijk}} = -\sum_{r=0}^{bin-1} \frac{\partial p(r)}{\partial\mu^{\xi}_{ijk}} \times (\log(p(r)) + 1) \tag{4.6}$$

$$\frac{\partial H(F(\mathbf{T}))}{\partial\mu^{\xi}_{ijk}} = -\sum_{f=0}^{bin-1} \frac{\partial p(f)}{\partial\mu^{\xi}_{ijk}} \times (\log(p(f)) + 1) \tag{4.7}$$

$$\frac{\partial H(R,F)}{\partial\mu^{\xi}_{ijk}} = -\sum_{r=0}^{bin-1} \sum_{f=0}^{bin-1} \frac{\partial p(r,f)}{\partial\mu^{\xi}_{ijk}} \times (\log(p(r,f)) + 1) \tag{4.8}$$

These computations require the derivative of the intensity distribution, which requires the derivative of the joint histogram $\mathbf{H}$ [48]:

$$\frac{\partial\mathbf{H}(r,f)}{\partial\mu^{\xi}_{ijk}} = \sum_{\vec{x}\in\Omega} \beta^3_r\left(R(\vec{x}); r\right) \left.\frac{\partial\beta^3_f\left(i, f\right)}{\partial i}\right|_{i=F(\mathbf{T}(\vec{x}))} \left.\frac{\partial F\left(p\right)}{\partial p}\right|_{p=\mathbf{T}(\vec{x})} \frac{\partial\mathbf{T}(\vec{x})}{\partial\mu^{\xi}_{ijk}} \tag{4.9}$$

where β^3 is a cubic B-spline kernel which is used in the Parzen window scheme for joint histogram filling. The kernel size along each dimension of the joint histogram can be set to different values and are noted as β^3_r and β^3_f along the reference and floating image intensities respectively.

This approach provides the mathematical value of the gradient but involves significant computational redundancy, since each voxel is included in the neighborhood of several control points. Moreover it is memory intensive as each node requires one joint histogram per degree of freedom. In order to decrease this redundancy and the memory requirement, I propose a voxel-centric approach to evaluate the node-centric gradient [17]. I first compute the gradient value for every voxel, then gather the information from all voxels to obtain the nodal gradient values.

I computed the voxel-centric gradient values $\frac{\partial\mathbf{H}(r,f)}{\partial\mathbf{u}^{\xi}_{\mathbf{z}}}$ using the formulas in equation 4.9, with $\frac{\partial\mathbf{T}(\mathbf{x})}{\partial\mu^{\xi}_{ijk}}$ replaced by $\frac{\partial\mathbf{T}(\mathbf{x})}{\partial\mathbf{u}^{\xi}_{\mathbf{z}}}$, where $\frac{\partial\mathbf{T}(\mathbf{x})}{\partial\mathbf{u}^{\xi}_{\mathbf{z}}} = \mathbf{I}$ if $\mathbf{z} = \mathbf{x}$ as $\mathbf{T}(\mathbf{x}) = \mathbf{x} + \mathbf{u}(\mathbf{x})$.

From the voxel-centric gradient values, I extracted the analytical node-centric derivative of the similarity measure. I first applied a convolution window to the gradient field where the convolution window was a cubic B-Spline curve which matched the basis functions in the deformation model in terms of node spacing; it was equivalent to $\frac{\partial\mathbf{T}(\mathbf{x})}{\partial\mu_{ijk}}$ in equation 4.9. In a second time, I extracted the gradient value, using linear interpolation, from the smoothed image at the node position. This conversion step is illustrated by figure 4.2.

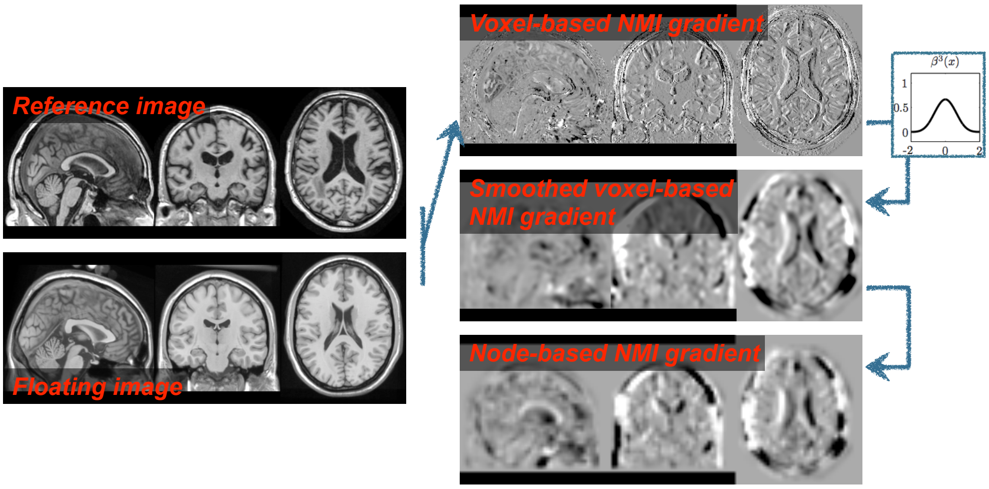


Figure 4.2: From voxel-based to node-based NMI gradient. The voxel-based NMI gradient is first computed, then convolved with a spline kernel and lastly the node-based gradient is extracted using a linear interpolation.

As seen in equation 4.4, the gradient of the bending energy is also required. Abbreviating equation 4.3 as:

$$\mathcal{P}_{BE} = \frac{1}{N}\sum_{\vec{x}\in\Omega} A^2 + B^2 + C^2 + 2D^2 + 2E^2 + 2F^2, \tag{4.10}$$

the derivative of the penalty term involves a sum of derivatives each of which can be obtained using the chain rule, e.g.:

$$\frac{\partial\left(A^2\right)}{\partial\mu^{\xi}_{ijk}} = \frac{\partial\left(A^2\right)}{\partial A}.\frac{\partial A}{\partial\mu^{\xi}_{ijk}} = 2A\frac{\partial A}{\partial\mu^{\xi}_{ijk}}. \tag{4.11}$$

Similarly to BE evaluation this gradient was computed at the control point positions only for performance reason.

4.1.4 GPU implementation

The F$_3$D implementation was achieved using CUDA [102] which is an Application Programming Interface developed by NVidia to simplify general purpose programming on graphics card as well as the interface between CPU (host) and GPU (device). The framework comprises four steps, organised as in Fig. 4.3.

The first step performs image interpolation via cubic B-Splines and trilinear interpolation to define the new voxel position and intensity. As already stated, the computation of each voxel's displacement and intensity interpolation is independent and their parallel hardware implementation is therefore straightforward. However the calculations are demanding in terms of dynamic memory resources, requiring allocation of around 22 registers per computational thread. As GPU memory is limited, a higher register requirement per thread dictates that fewer threads may be executed concurrently, resulting in sub-optimal use of the device's computational resources. The ratio of active threads to maximum allowed (hardware dependent) is referred to as occupancy [102], and an efficient implementation should maximise this. A

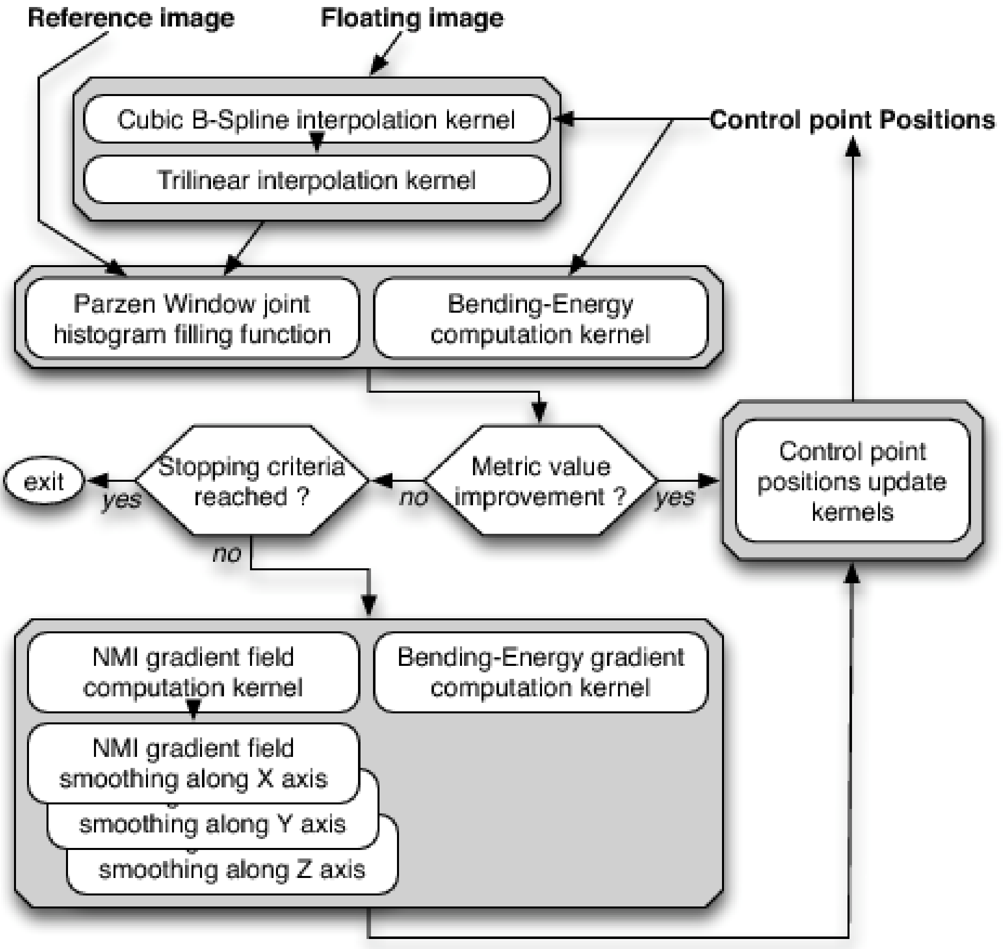

Reference image
Floating image
Cubic B-Spline interpolation kernel
Trilinear interpolation kernel
Control point Positions
Parzen Window joint histogram filling function
Bending-Energy computation kernel
exit
yes
Stopping criteria reached ?
no
Metric value improvement ?
yes
Control point positions update kernels
no
NMI gradient field computation kernel
Bending-Energy gradient computation kernel
NMI gradient field smoothing along X axis
smoothing along Y axis
smoothing along Z axis

Figure 4.3: Organization of the proposed implementation

single kernel requiring 22 registers leads to an occupancy of 42%. For this reason this step has been split into two kernels, the first dealing with the B-Splines interpolation only and the second with trilinear interpolation. Register requirements then fall to 16 and 12 respectively, and occupancies increase to 67% and 83%. Such a technique allows a computation time improvement of 36.8% in the present case.

The second step involves filling the whole joint histogram and computing the different entropy values. A GPU implementation of this step did not show a significant reduction in computation time compared with serial implementation. Furthermore this step occupies only around 2.2% of the entire computation time. Moreover a GPU implementation necessitates use of single precision which, for this step, proves detrimental to accuracy[1]. For these reasons this step is executed on CPU rather than on GPU. This choice does not affect the computation time even with the data transfer between device and host.

In the third step the gradient value is computed for each voxel and the convolution windows are applied. As for the first step, I distributed the computation across several kernels to improve occupancy. The first kernel computed the gradient values. The gradient was then smoothed using three different

[1] Newer devices do offer double precision accurary, but at a significant lower performance [102]

kernels, each dealing with one axis. For these kernels it appeared that computing the cubic spline curve "on the fly" was faster than precomputing and fetching them from memory.

The last step normalises the gradient and updates the control point positions using a conjugate gradient optimisation. A first kernel is used to extract the maximal gradient value from the whole field. The field is split into several parts from which a maximal value is extracted. Subsequently, the largest value from the extracted maxima is kept. A last kernel updates the control point positions based on the normalised gradient value.

A final feature of this approach is the use of a convergence criterion. Whereas time constraints dictated that earlier implementations [29] performed a set (and small) number of iterations, F_3D iterates until convergence, aiming to ensure better registration. The convergence criteria is met if no increase of the objective function can be achieved using a step size of at least 0.01 times the smallest voxel dimension.

4.1.5 Code Profile and Benchmark

Table 4.1 show the benchmark results of the registration functions on various hardware. The CPU version, which has been implemented in C++, was run on a 3Ghz processor with 4 GB of RAM running 64-bit Linux operating system. GTX 8800 is a NVIDIA GPU with 768 MB of graphics memory and 128 CUDA cores. GTX285 is a NVIDIA GPU with 1 GB of graphics memory and 240 CUDA cores. The benchmarks were run with data of 100^3, 150^3, 200^3 voxels and the control point grid spacing of 5 voxels. The benchmarks were run multiple times and the speed-up achieved by using Nvidia GPUs is reported. This is highlighted in table 4.1 for the different modules.

	GTX8800 speedup			GTX285 speedup		
Module names	100^3	150^3	200^3	100^3	150^3	200^3
B-spline deformation	12.2	12.55	11.66	20.3	23.0	22.27
Trilinear interpolation	12.3	33.0	48.50	18.5	55.0	97.0
Affine deformation	0.97	2.70	4.67	1.88	5.2	9.35
Spatial gradient	7.82	13.71	17.25	17.2	27.3	34.5
Voxel-based NMI gradient	42.33	40.79	40.72	63.5	72.0	67.86
Node-based NMI gradient	12	20.25	20.85	18.0	27.0	29.2
Bending energy	0.43	1.2	3.66	0.75	3.0	5.5
Bending energy gradient	0.54	1.20	1.84	0.86	1.7	2.03

Table 4.1: Speed-up ratio for various registration modules on a CPU and various GPUs on 100^3, 150^3, 200^3 voxels data

A non-rigid registration using the F_3D approach (see figure 4.4) was performed in 42 sec on standard T1 weighted MR brain images.

4.2 Experiments

The previously described algorithm has been assessed in three different contexts. The first experiment required accuracy while performing segmentation propagation [18]. The second experiment demanded efficient implementation to perform brain shift correction in a surgical setting [21]. The third experiment assessed the quality of the implementation while dealing with very large dataset to create a mouse embryo

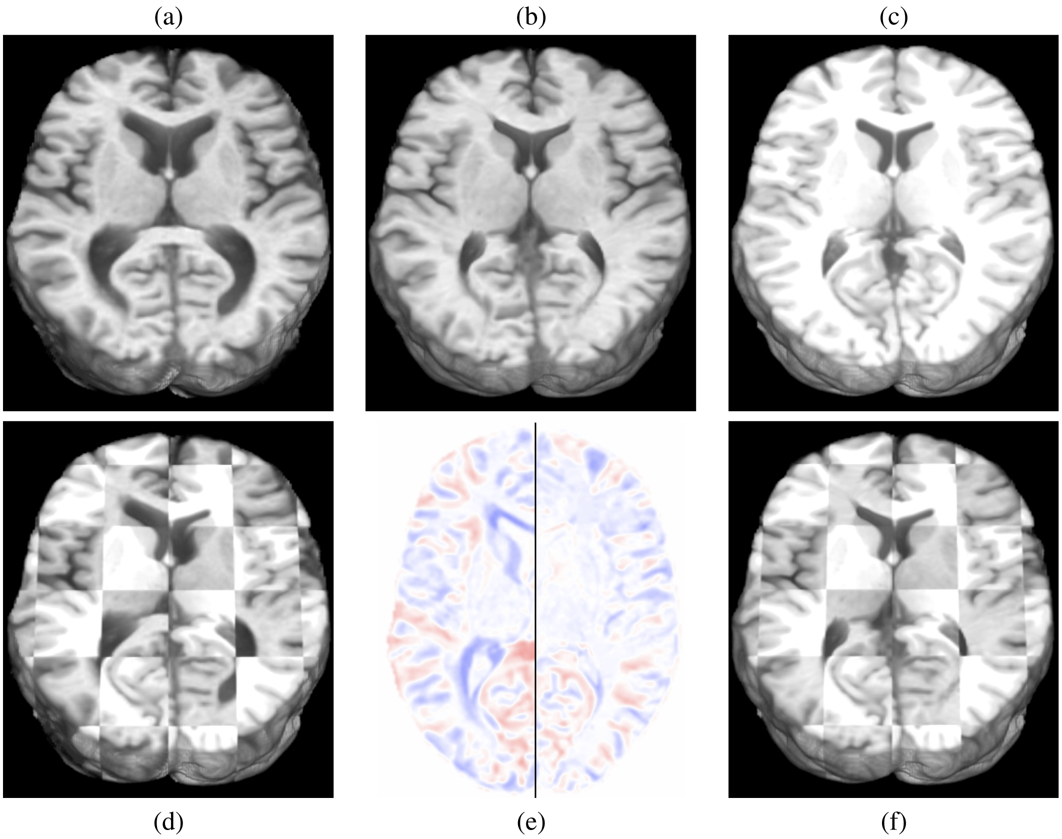


Figure 4.4: 3D image registration. By optimising a measure of the similarity of two images (NMI) as a function of the spline weights, a floating image (a) can be automatically brought into alignment (b) with a reference image (c). The initial misalignment is illustrated by alternating between the two images (d) and as a difference image (e-left). The equivalent results after registration are shown in (f) and (e-right). Optimisation of the 40-by-44-by-40-by-3 = 211,200 weights is computationally challenging.

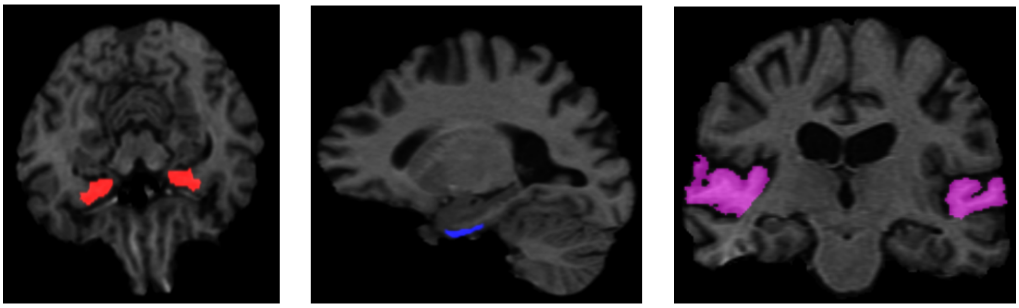
average image from high-field MR images [24].

4.2.1 Segmentation propagation

In order to assess the accuracy of the proposed implementation I performed segmentation propagations and compared the results with those obtained from a *classical* FFD implementation[2]. The dataset consisted in 20 T1-weighted brain MR images, of which 10 scans were of clinically-diagnosed Alzheimer's disease (AD) patients and 10 age-matched control subjects. The data acquisition protocol as well as the subject characteristics have been described by Chan *et al.* [103]. The size of the images used in the registration was $180 \times 180 \times 124$ voxels, with a voxel spatial resolution of $0.9375 \times 0.9375 \times 1.5\text{mm}^3$. For each brain image, different manual segmentations have been performed. The regions of interest are listed in table 4.2 and a few are illustrated in figure 4.5.

Using the F$_3$D and the *classical* serial FFD, I performed 380 (20×19) registrations in which each scan was registered to all others. As scans of both diagnosed AD patients and controls were used, I expected significantly differing brain shapes and correspondingly significant deformations to be recovered by the algorithms. Prior to the non-rigid registration, an affine registration has been performed using FLIRT [104]. All the non-rigid registrations were performed with a pyramidal approach with 3 levels.

[2] A FFD algorithm executable can be downloaded from Daniel Rueckert's webpage: http://www.doc.ic.ac.uk/~dr



(a) (b) (c)

Figure 4.5: Examples of manually segmented masks. Segmentation of the amygdala areas are presented on the axial view (a), the blue area on the sagittal view (b) corresponds to the entorhinal cortex and the coronal view (c) shows the superior temporal gyri.

The finer lattice of control points had a spacing of 5 mm along each axis. Both algorithms employed a conjugate gradient optimisation, and a bending energy weight of $\alpha = 1\%$ (Eq. 4.2). As a preprocessing step, each T1w MR image was skull stripped using BET [105] and a dilation was applied on the obtained mask. The resulting set of deformation fields were then used to propagate the manually segmented masks between images. I computed the Dice Score coefficient (DS), as in equation 4.12, between each manual segmentation (M_m) and the corresponding propagated (M_p) region of interest.

$$\text{DS} = 2 \times \frac{||M_m \cap M_p||}{||M_m|| + ||M_p||} \tag{4.12}$$

The DS rates the overlap of two masks between 0 and 1, where 1 indicates a perfect overlap and 0 none. Table 4.2 summarizes the obtained results using both implementations. For comparison, the DS was computed using only an affine transformation also.

Mask area	Affine only	*classical* FFD	Fast-FFD
left amygdala	0.531 (0.163)	0.759 (0.089)	0.776 (0.066)
left entorhinal cortex	0.203 (0.189)	0.296 (0.164)	0.372(0.155)
left fusiform gyrus	0.398 (0.103)	0.483 (0.096)	0.499(0.098)
left hippocampus	0.429 (0.157)	0.658 (0.093)	0.686(0.075)
left medial-inferior temporal gyrus	0.626 (0.070)	0.699 (0.061)	0.709(0.064)
left parahippocampal gyru	0.399 (0.146)	0.527 (0.094)	0.637(0.070)
left superior temporal gyrus	0.607 (0.069)	0.742 (0.057)	0.737(0.048)
left temporal lobe	0.748 (0.052)	0.832 (0.046)	0.827(0.041)
right amygdala	0.571 (0.139)	0.779 (0.072)	0.787 (0.058)
right entorhinal cortex	0.170 (0.177)	0.266 (0.169)	0.334 (0.162)
right fusiform gyrus	0.450 (0.111)	0.542 (0.119)	0.534 (0.113)
right hippocampus	0.479 (0.162)	0.631 (0.120)	0.710 (0.086)
right medial-inferior temporal gyrus	0.662 (0.062)	0.763 (0.059)	0.760 (0.053)
right parahippocampal gyru	0.276 (0.208)	0.323 (0.189)	0.340 (0.275)
right superior temporal gyrus	0.624 (0.055)	0.780 (0.048)	0.775 (0.040)
right temporal lobe	0.733 (0.119)	0.811 (0.128)	0.813 (0.125)

Table 4.2: Average (standard deviation) results of the segmentation propagation. For each propagation, the Dice similarity value between the manual and the propagated segmentations has been computed.

For these data the mean registration time was around 5 hours per image using the *classical* FFD algorithm, but less than 20 seconds using the GPU-based implementation. For comparison, the proposed implementation had a mean computation time of 3 minutes 18 seconds when running on the same CPU.

The comparison of the CPU and GPU implementations of the proposed algorithm showed a speed-up of approximately 10 times using the latter. I conclude that the algorithm maps well to parallel architectures, and consequently is well-suited to GPU execution. However, for the segmentation propagation examples dramatically higher performance was shown by the F_3D formulation (and implementation) compared with the *classical* algorithm. Thus the majority of the speed improvement arises from the improved formulation, rather than the GPU implementation itself. Two features of the formulation are likely to be responsible: (1) computation of the gradient for all control points concurrently, rather than serially, and (2) use of an analytical objective function gradient, rather than a symmetric difference estimate. The latter, in particular, is significant: a symmetric difference evaluation is time consuming as it requires resampling of the floating image and evaluation of the objective function value six times per control point. Moreover, the use of the analytical metric gradient may lead to a faster conjugate gradient convergence. The DS evaluation showed that both the *classical* FFD and the proposed implementation improved the overlap between regions of interest, compared to a single affine registration. Moreover the F_3D method appears to perform better in most cases; the higher values are statistically significant for the left and right entorhinal cortex and the left parahippocampal gyru when performing a paired t-test ($p < 0.01$). The improvements can be attributed to the use of a stopping criteria based on the objective function value (and the consequent increase in the number of iterations performed) in the Fast-FFD method. To limit computation time I used a maximum of 10 iterations in the *classical* FFD.

4.2.2 Brain shift estimation in the OR

The presented algorithm has been evaluated according to two criteria: registration accuracy and efficiency. The validation was done using pre-operative and intra-operative MR images from six subjects. Two of them went through tumour resection and the other three went through temporal lobe resection for epilepsy treatment. The images were acquired using a 3D FLASH sequence with TR = 5.25ms, TE = 2.5ms and flip angle = $15°$. The images have a spatial resolution of $1.1 \times 1.1 \times 1.3$mm and are 180^3 voxels in size after cropping of the brain.

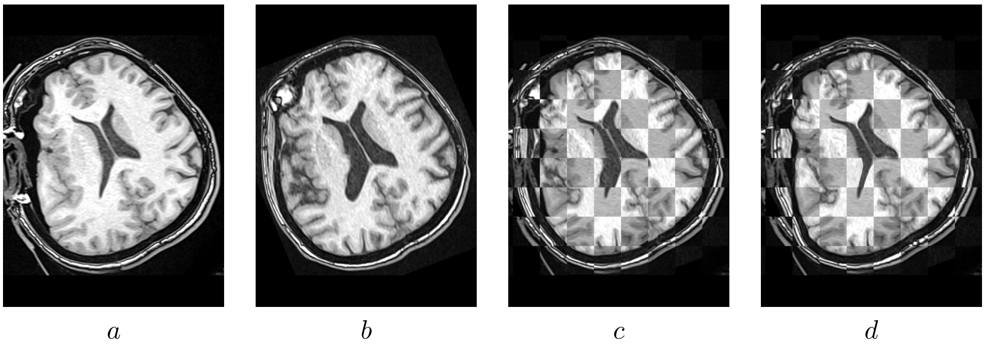


Figure 4.6: Registration result example. (a) Intra-operative image, (b) affinely registered pre-operative image and a checkerboard showing the differences after affine (c) and non-rigid (d) registration.

Registration Accuracy

For quantitative analysis, a senior radiologist was asked to rate the registration accuracy in several regions of interest. I registered three pre-operative images to the corresponding intra-operative MR images. They were presented with the intra-operative image, the affine and the non-rigidly registered pre-operative image. The control point spacing was set to 5 voxel width along each axis for the cubic B-spline parametrisation. They categorised registered landmarks into three categories: good, acceptable and poor. Results are summarised in table 4.3. For a quantitative analysis overlap measurement, Dice Similarity coefficient (DS), have been computed between manual segmented ventricles in the intra-operative images and the pre-operative propagated segmentation. For the 6 subjects, the DS after affine registration was 63.2% (8.8%) and increased to 82.7% (6.5%) after non-rigid alignment.

		Patient 1	Patient 2	Patient 3
Affine registration	Good	8	11	11
	Acceptable	2	2	5
	Poor	9	6	3
Non-rigid registration	Good	17	18	16
	Acceptable	2	1	3
	Poor	0	0	0

Table 4.3: Distribution of the registration accuracy based on a radiologist's visual assessement.

Implementation Efficiency

The mean (std) computation time for the affine registration was 23.3 (2.3) seconds using the CPU implementation and 14.6 (1.4) seconds when running on the GPU (GTX285). It took 249.5 (40.34) seconds and 36.5 (5.4) to perform the non-rigid registration using respectively the CPU- and the GPU-based implementation.

The proposed GPU-based implementation is suitable for near-real time surgery in a operative room settings.

4.2.3 Atlas building using large size dataset

The fast free-form deformation algorithm has been used to create a group-wise registration of 19 mouse embryos [24]. The embryo age was 15.5 days post coitum. The dataset consists of two groups of mouse with different background: 6 CD-1 and 13 C57BL/6 from which 6 were wild-types and 5 were Chd7$^{+/-}$ knockout. Images have been acquired using a Varian VNMRS 9.4 Tesla MRI system (Varian Inc., Palo Alta CA, USA) using a $33mm$ quadrature birdcage volume coil (RAPID Biomedical GmbH, Würzburg, Germany). A gradient-echo sequence with TR=$20ms$, 7 averages and a flip angle=60°was used. The field of view was equal to $27 \times 27 \times 27mm$, the matrix size=512^3 which has been zero-filled to 1024^3 on the console to a voxel-size of $26 \times 26 \times 26\mu m$. Echo-time (TE) was set to $9ms$. The images have been acquired by Jon O. Cleary from the Centre for Advanced Biomedical Imaging[3], UCL (CABI).

The groupwise registration was performed similarly to Rohlfing *et al.* [106] where an iterative process was used. A reference image was selected and all 19 images were rigidly aligned to this reference. A block-matching technique developed by Ourselin *et al.*[31] has been used to performed this initial

[3] http://www.ucl.ac.uk/cabi

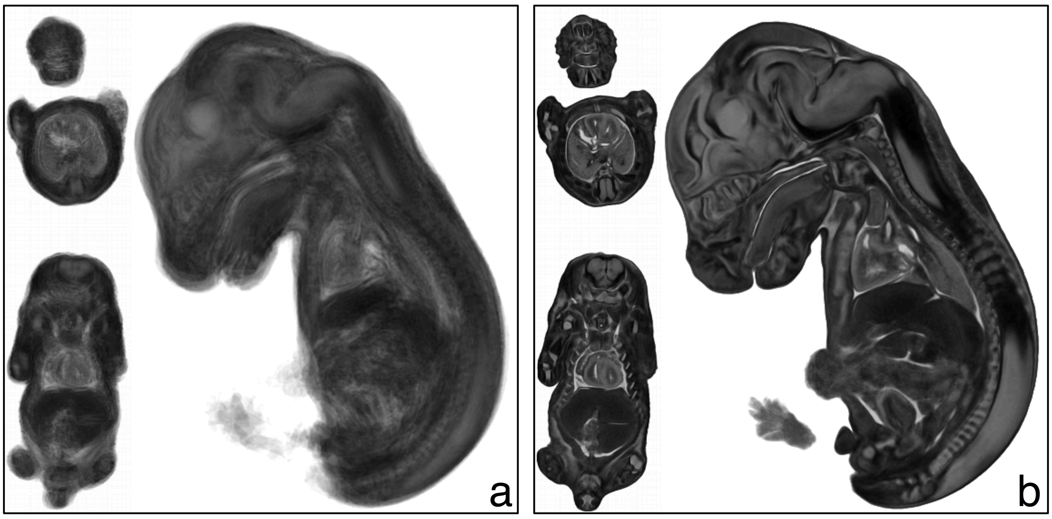


Figure 4.7: Groupwise average image. Average image after five affine steps (a) and 18 non-rigid steps (b). An increase in sharpness can be noticed which corresponds to an alignment improvement.

registration step. After an average image of the 19 input images have been created, it was used as a new reference for affine registration. Each registration step was concluded by an update of the average image. I performed 5 affine steps followed by 18 non-rigid step using F_3D. I used a coarse-to-fine approach where the final control point spacing was set to 5 voxels, approximatively $130\mu m$. Figure 4.7 shows the average image after affine and non-rigid registration.

The final average image has been manually segmented by Francesca C. Norris from CABI in five regions of interest: the whole brain, the olfactory bulb, the mesencephalic vesicle, the pituitary gland and the heart. The segmented regions of interest are shown in figure 4.8. I inverted all obtained deformation fields from the groupwise registration using the method proposed by Ashburner *et al.* [107]. Briefly local affine transformations are extracted from a deformation field to enable straightforward inversion of the deformation field. The manually segmented regions of interest have then been propagated in the embryo's native spaces.

The back propagation of the manual segmentations enables morphometric study using each region volumes. Despite the limited amount of subject, significant differences could be seen between the different mouse line. Results are presented in figure 4.9. Note that only graph with significant differences are shown. The phenotyping process using mice embryos has limited literature and there is no consensus yet on an accepted mode of data presentation. Volume comparisons have thus been performed using both normalised and unnormalised data with respect to the whole body volume size.

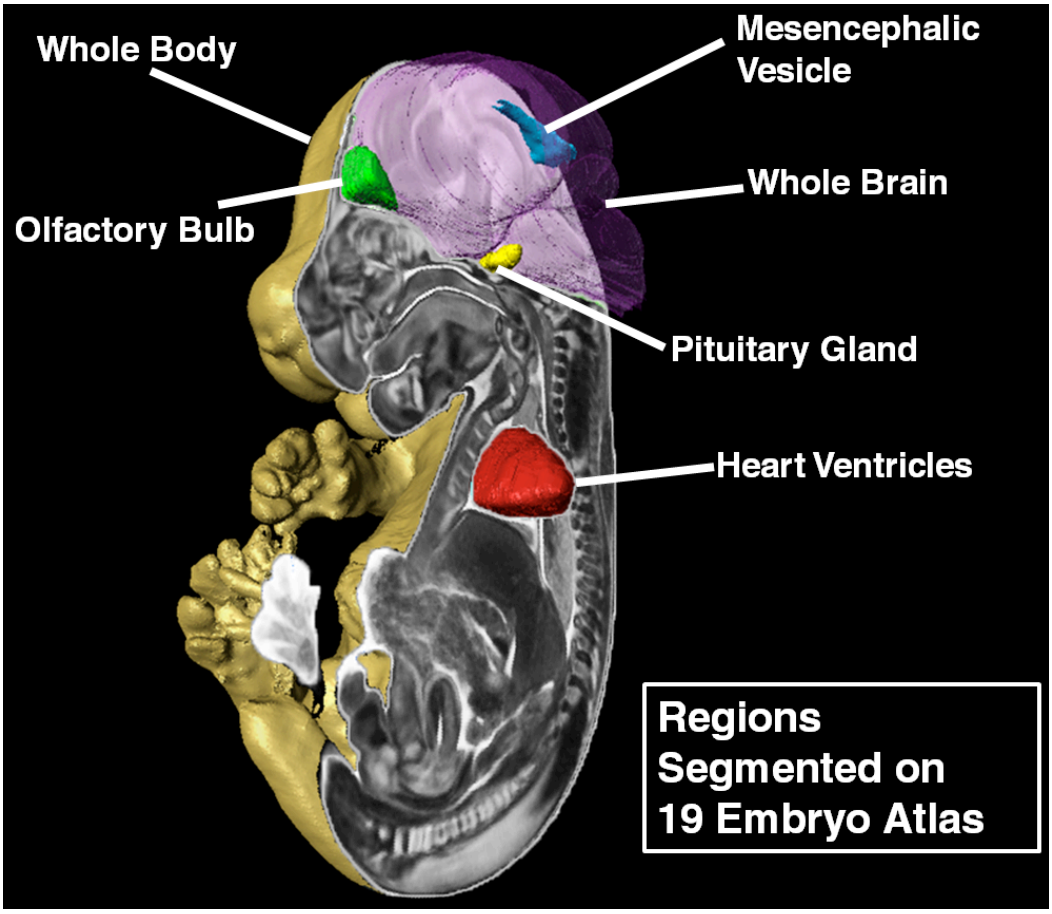


Figure 4.8: Segmented regions of interest. Five regions of interest have been manually segmented using the groupwise average image.

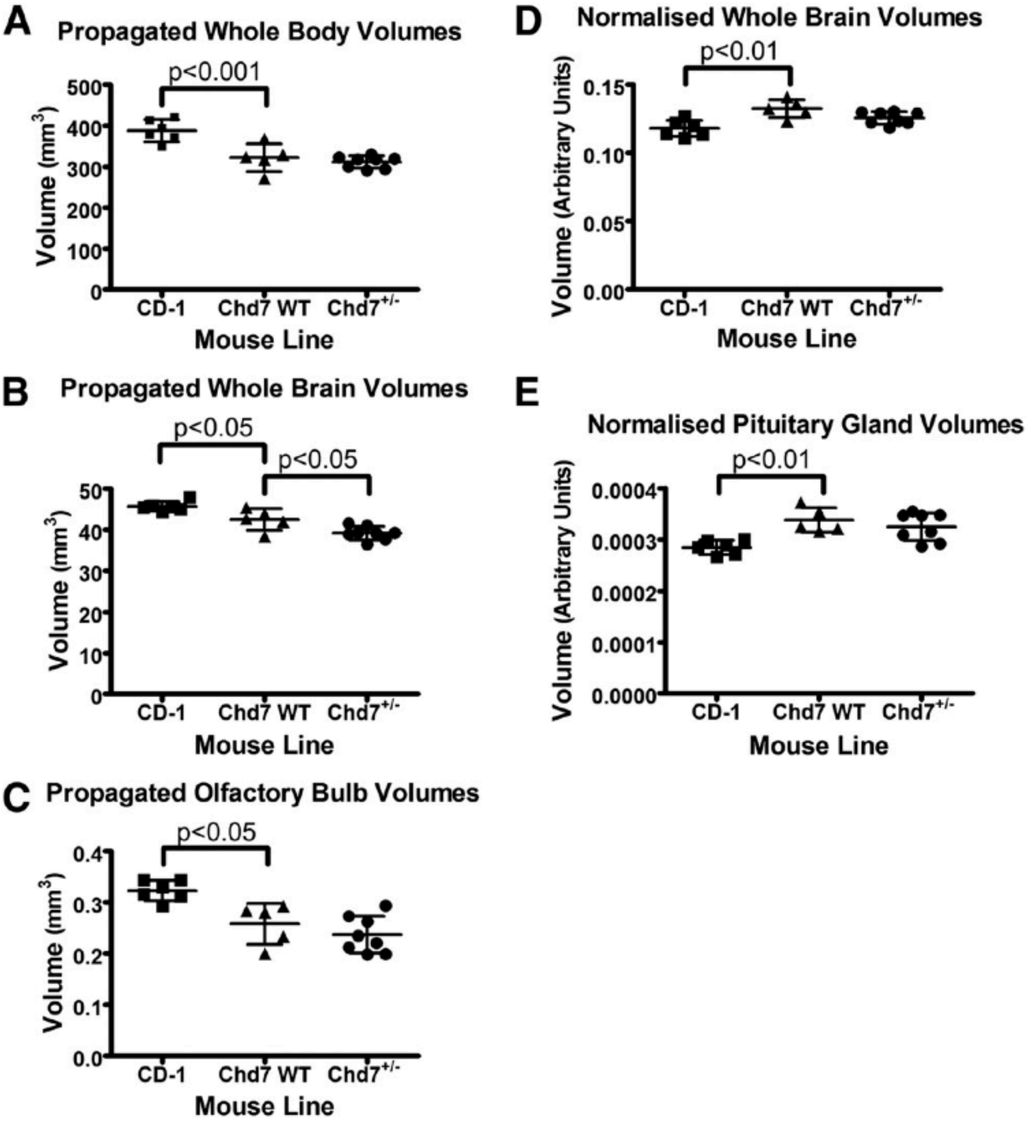


Figure 4.9: Regions of interest volume differences.

Chapter 5

Fast Free-Form Deformation: F$_3$D, advanced

Based on the work of the previous chapter, I extended the F$_3$D framework to tackle two more challenges of medical image registration: topology conservation and data corruption.

5.1 Unfolded transformation

Using the Free-Form Deformation framework, there is no guarantee that the obtained deformation will conserve topology. Indeed, the bending energy penalty term enforces smoothness but does not forbid folding. Approaches have been proposed to ensure un-folded transformation using a B-spline parametrisation.

Andersson *et al.* [1] use a two-step approach in the FNIRT software[1]. The first step consists of locally unfolding a deformation field using a method proposed by Karaçali and Davatzikos [108]. This method required a discretised deformation field in the space of the reference image and is not directly suitable for a parametric approach. For this reason, Andersson *et al.* are required, as a second step, to fit a cubic B-spline parametrisation that best approximates the corrected dense deformation field.

Tanner *et al.* [109] and Rohlfing *et al.* [69] proposed to use the Jacobian determinant as a penalty term in order to ensure positive determinant. Their approach relied on symmetric difference computation for the objective function optimisation and as a consequence is not suitable for efficient implementation.

Rueckert *et al.* [70] employed an approach similar to the re-gridding scheme proposed by Christensen *et al.* [42] for the fluid algorithm. A composition scheme was used each time the algorithm was reaching user-defined displacement. Implementation of this method is memory expensive as it requires the storage of every lattice of control points.

Sdika [68] used the Jacobian determinant as a hard constraint in the FFD optimisation process.

All these approaches are time consuming and do not fit the F$_3$D algorithm requirement as it has been designed for fast registration. In order to guarantee folding-free transformation, we extended the work of Rohlfing *et al.* by using an analytical formulation of the new objective function. This work has been successfully used in a Grand Challenge context to register pairs of lung images [27].

[1] `http://www.fmrib.ox.ac.uk/fsl/fnirt/index.html`

5.1.1 Jacobian determinant as a penalty term and its derivative

As seen in the previous chapters, while using the free-form deformation framework, the positions of the control points $\vec{\mu}$ are displaced until optimisation of an objective function. This function is, in the classical free-form deformation scheme, composed of the normalised mutual information (NMI) as a metric and the bending energy (BE) as a penalty term. In order to ensure topology preservation, we added a second penalty term based on the Jacobian determinant at every voxel. The objective function $\mathcal{O}(R, F; \{\vec{\mu}\})$ is then defined as:

$$\begin{aligned}
\mathcal{O}(R, F; \{\vec{\mu}\}) &= \alpha \times \frac{(H(R) + H(F(\mathbf{T}))}{H(R, F(\mathbf{T}))} \\
&- \beta \times \sum_{\vec{x}\in R} \left(\frac{\partial^2\mathbf{T}(\vec{x})}{\partial x^2}\right)^2 + \left(\frac{\partial^2\mathbf{T}(\vec{x})}{\partial y^2}\right)^2 + \left(\frac{\partial^2\mathbf{T}(\vec{x})}{\partial z^2}\right)^2 \\
&\quad +2 \times \left(\left(\frac{\partial^2\mathbf{T}(\vec{x})}{\partial xy}\right)^2 + \left(\frac{\partial^2\mathbf{T}(\vec{x})}{\partial yz}\right)^2 + \left(\frac{\partial^2\mathbf{T}(\vec{x})}{\partial xz}\right)^2\right) \\
&- \gamma \times \sum_{\vec{x}\in R} \log\left(|\text{Jac}(\mathbf{T}(\vec{x}))|\right)^2
\end{aligned} \tag{5.1}$$

where α, β and γ are user-defined weights and are set to sum to 1, $H(R)$ and $H(F(\mathbf{T}))$ are the marginal entropies and $H(R, F(\mathbf{T}))$ the joint entropy.

In order to optimise $\mathcal{O}(R, F; \{\vec{\mu}\})$, we used a conjugate gradient ascent scheme. The gradient is computed analytically for each component of the function. The gradient computation of the NMI and the BE have been describe in the previous chapter. I will thus focus here on the computation of the Jacobian-based penalty term. In contrast to Rolhfing *et al.* [69], who used an approximated gradient of their objective function, we compute the gradient of the Jacobian-based penalty term analytically. It can be done as:

$$\begin{aligned}
\frac{\partial\left(\sum_{\vec{x}\in R}\log\left(|\text{Jac}(\mathbf{T}(\vec{x}))|\right)^2\right)}{\partial\mu_{i,j,k}} &= 2 \times \log\left(|\text{Jac}(\mathbf{T})|\right) \\
&\quad \times\text{Tr}\left(\text{Jac}^{-1}(\mathbf{T})\frac{\partial\text{Jac}(\mathbf{T})}{\partial\mu_{i,j,k}}\right),
\end{aligned} \tag{5.2}$$

where i, j and k correspond to the control point indices respectively along the x-, y- and z-axis and Tr(.) the trace operator. For efficiency reasons, one can approximate the Jacobian-based penalty term at the control point positions. It corresponds to the same approach used for the bending energy and described in the previous chapter.

The Jacobian-based penalty term ensure strictly positive Jacobian determinant at every voxel but folding may still occur in the line ascent process. Indeed, not only the Jacobian-based determinant is considered for optimisation but also the NMI and BE gradient.

5.1.2 Folding correction applied to parametric model

In parallel to the Jacobian-based penalty term, I implemented a folding correction scheme that takes advantage of the analytical formulation of the Jacobian matrix using the cubic B-Spline parametrisation.

For each voxel in the reference space whose Jacobian determinant value is not strictly positive, we compute the gradient of the Jacobian determinant of the neighbouring control points and displace them until the Jacobian determinant value is strictly positive.

Figure 5.1 shows an example of registration performed using either the F3D implementation described in the previous chapter or F3D with the Jacobian-based penalty term and the proposed folding correction scheme. It can be noticed that result are visually similar but no folding has been generated by the later.

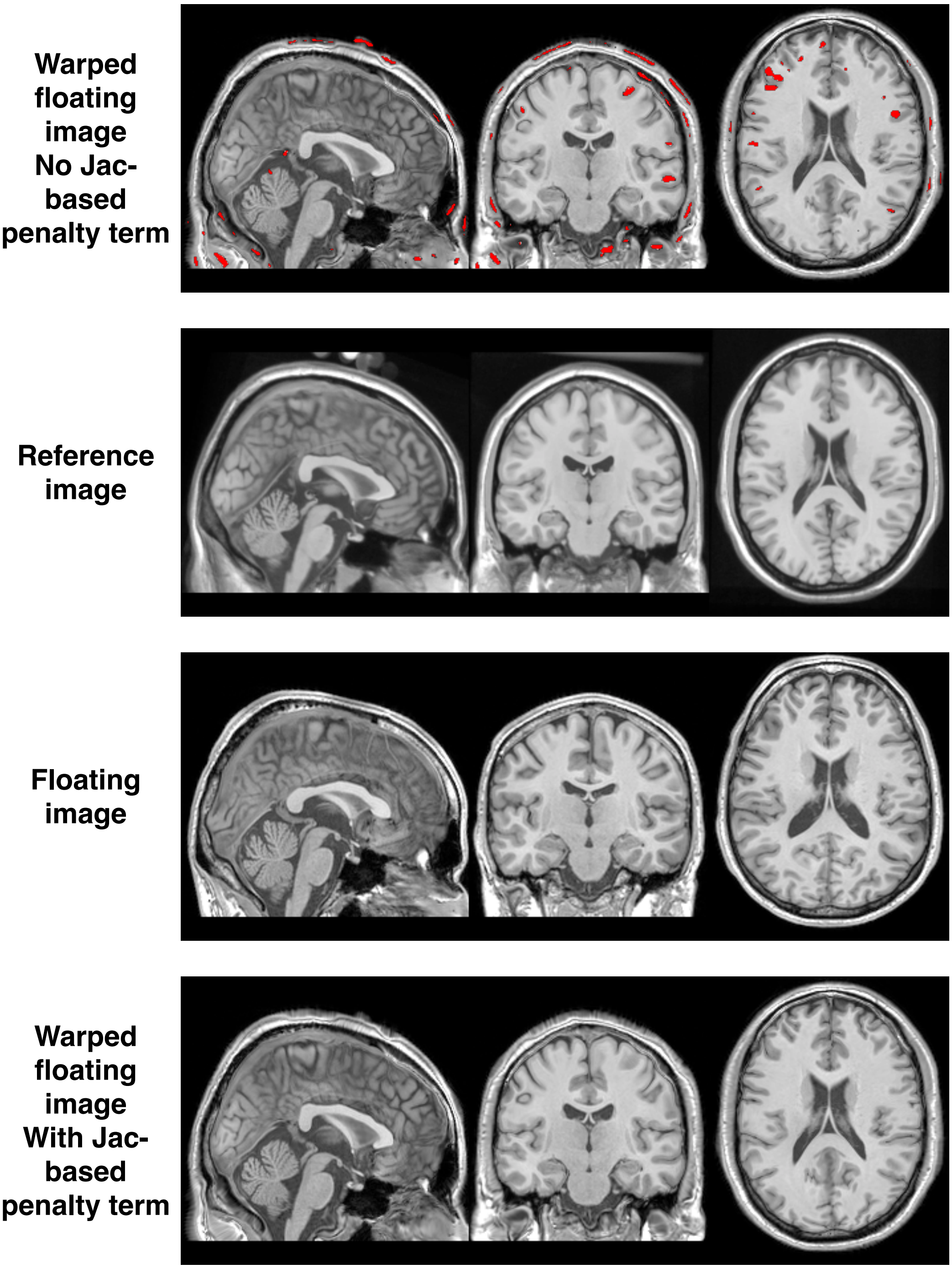

Warped floating image No Jac-based penalty term
Reference image
Floating image
Warped floating image With Jac-based penalty term

Figure 5.1: A floating image has been registered to a reference image using F3D with and without the Jacobian-based penalty term. Folded voxels (zero or negative Jacobian determinant) are shown in red.

5.1.3 Lung image registration example

This algorithm has been applied in the context of the Evaluation of Methods for Pulmonary Image Registration (EMPIRE) 2010 Grand Challenge [2]. The challenge consisted in the registration of pair of lung images. To perform the initial affine registration, I used a block-matching algorithm [31].

Together with Jamie McClelland from the Centre for Medical Image Computing, UCL, we designed a pipeline to perform the registration. We ensured that the total registration time for all stages was reasonable as there was a time constraint of 3 hours in which to perform another 10 registrations at the workshop. Furthermore, we ensure that all stages of the registration can fit onto the GPU memory. The pipeline consists of 1 global registration stage and 3 consecutive local registration stages. Figure 5.2 shows the result of the different stages applied to the first pair of scans. For each stage, the result of the previous stage was used to initialise the transformation.

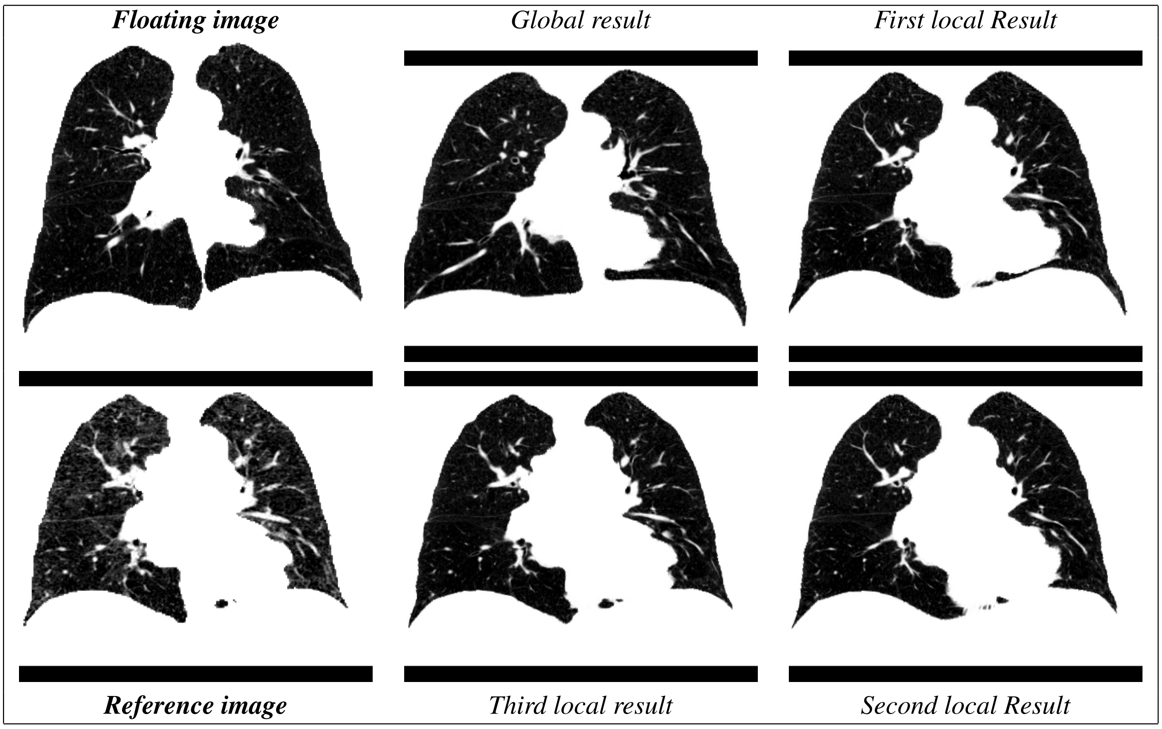


Figure 5.2: Illustration of the different registration steps for case 01.

Global registration stage

A block matching approach was preferred since it allows feature alignment in the lung rather than global alignment of external lung volume boundaries. We obtained a better initialisation for the local registration than using a global scheme.

The size of the block was set to 4^3 voxels and the neighbourhood area of a block has been defined such that the block in the reference image and the block in the floating image always overlap by at least one voxel. As a measure of similarity we used cross-correlation. This has the advantage of being suitable for local intensity difference when used locally. In order to decrease the computation time

[2] http://empire10.isi.uu.nl/

and increase the robustness, only the blocks in the reference image with the highest intensity variance were considered. They correspond to contrast areas and hence structural information. Only half of the possible blocks were then considered. Moreover, we used the provided reference lung mask to ignore any correspondences outside of the region of interest. Due to the large number of voxels, we only performed the two first levels of a three-level pyramidal approach. Using a trimmed least square optimisation scheme, 50% of the blocks were considered as outliers and the transformation was extracted from the remaining 50% blocks.

First local registration stage

The previously obtained global registration parameters were used to initialise the control point positions. The aim of the initial local registration was to quickly register the main structures in the lung.

This has been done performing only registration on the 2 coarsest levels of a 4-level pyramidal approach. The control point spacing has been set to 6 voxel-width. The maximal number of iterations per level was set to 500 and the weight of the bending-energy penalty term to 0.01%. After computation of the NMI gradient, it is smoothed using a Gaussian kernel with standard deviation set to half the control point spacing size (3 voxels width). Moreover a reference image mask was used to ignore every voxel outside of the mask from the similarity measure computation, thus allowing the algorithm to align the lung features rather than the lung boundaries. No penalty term based on the Jacobian determinant has been used at this stage.

Second local registration stage

The second local registration stage was performed on twice down-sampled input images. In contrast to the previous stage, no mask is used, which ensure that the algorithm aims at aligning the border of the lungs. Structures outside the lungs can move differently than the lungs during respiration. All voxels outside the lung mask are set to 0 Hu (approximately soft tissue) to ensure that they do not affect the registration. The NMI gradient is still smoothed with the same kernel size, the bending-energy weight is set the 0.1% and the Jacobian-based penalty term is introduced with the same weight. The maximal number of iteration is set to 500. The aim of this stage was to quickly align the border of the lung.

Third local registration stage

Whereas the previous two stages aimed to coarsely align the internal structures and the lung borders, the goal of the final stage was to established a detailed alignment of the entire lung. This last stage was performed using three levels in the pyramidal approach. No mask was used, the penalty term weights were both set to 0.1% and no smoothing was performed on the NMI gradient field. The maximal number of iterations was set to 300 for each of the three levels. Since the registration was never performed at full image resolution due to the contest time constraint; we ensured that the produced transformation did not generate any folding. This was done by running our folding correction scheme at full resolution as a post-registration step.

Table 5.1 shows the registration accuracy evaluation for the 20 pairs of images provided for the EMPIRE challenge. Evaluation has been performed by the EMPIRE 2010 (Evaluation of Methods for

Pulmonary Image Registration 2010) board. Different criteria have been assessed such as lung boundary, fissure and landmark alignment. The singularities of the deformation field have also been evaluated using the Jacobian maps of the deformation.

	Lung Boundaries		**Fissures**		**Landmarks**		**Singularities**	
Scan Pair	**Score**	**Rank**	**Score**	**Rank**	**Score**	**Rank**	**Score**	**Rank**
01	0.00	2.50	0.00	4.00	1.23	3.00	0.00	11.50
02	0.00	11.00	0.00	15.00	0.45	14.00	0.00	12.50
03	0.00	5.50	0.00	12.50	0.45	16.00	0.00	12.00
04	0.00	9.00	0.00	16.50	0.94	9.00	0.00	14.00
05	0.00	13.00	0.00	16.00	0.03	17.00	0.00	13.50
06	0.00	16.00	0.00	7.00	0.26	4.00	0.00	14.00
07	0.00	1.50	1.69	20.00	1.38	3.00	0.00	10.00
08	0.00	4.00	0.00	9.00	0.71	5.00	0.00	12.50
09	0.00	9.00	0.00	6.50	0.56	10.00	0.00	13.00
10	0.00	6.00	0.00	15.00	1.11	6.00	0.00	13.00
11	0.00	4.00	0.01	9.00	0.84	10.00	0.00	11.50
12	0.00	20.00	0.00	13.50	0.05	10.00	0.00	14.50
13	0.00	3.00	0.09	16.00	0.74	1.00	0.00	13.00
14	0.00	1.00	2.86	10.00	1.23	3.00	0.00	9.50
15	0.00	8.00	0.00	23.00	0.63	7.00	0.00	12.50
16	0.00	3.50	0.00	5.00	0.93	6.00	0.00	13.50
17	0.00	14.00	0.04	12.00	0.59	1.00	0.00	14.00
18	0.00	3.00	0.04	2.00	1.22	2.00	0.00	10.50
19	0.00	14.00	0.00	30.00	0.54	16.00	0.00	14.50
20	0.00	3.50	0.74	4.00	1.06	2.00	0.00	10.50
Avg	0.00	7.57	0.27	12.30	0.75	7.25	0.00	12.50
Average Ranking Overall								9.90
Final Placement								2

Table 5.1: Results for each scan pair, per category and overall. Rankings and final placement are from a total of 34 competing algorithms.

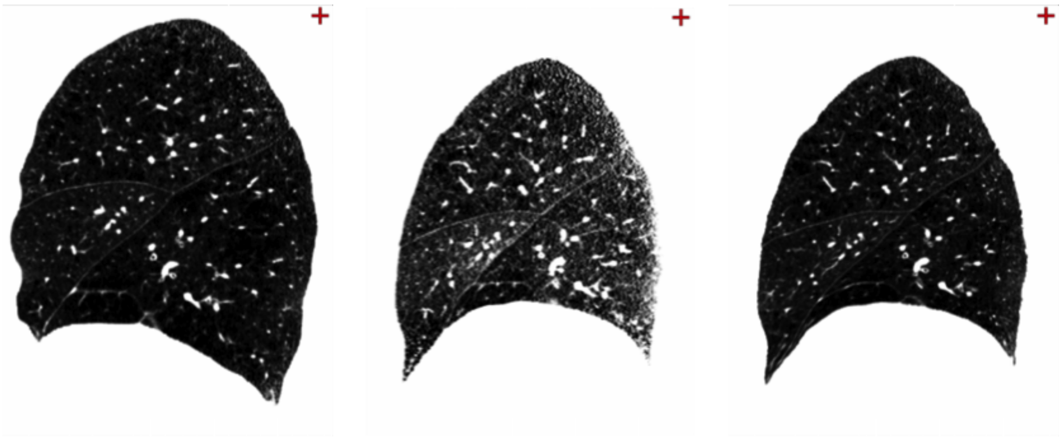
Each registration was performed using our CPU-based single-threaded implementation. A computer cluster was used to assess the deformations, we are thus unable to report a specific architecture, but the range of processor clocks was 1.8 to 3 GHz. For comparison, the registrations were also performed using our GPU-based implementation. An NVidia Quadro FX 2800m on a laptop was used and we then used the same configuration during the workshop. Note that the global registration and the folding correction step required too much memory to be performed on the GPU with this card. Table 5.2 reports the mean computation times for each architecture.

Computation time	CPU-based implementation	GPU-based implementation
Global registration	1.40 (0.57)	NA
Local registration 1	1.28 (0.47)	1.06 (0.41)
Local registration 2	1.32 (0.84)	0.83 (0.36)
Local registration 3	12.25 (5.13)	1.56 (0.27)
Folding correction	1.16 (0.57)	NA

Table 5.2: Mean computation time in minutes (and standard deviation) of the different registration stages.

The results in table 5.1 indicate that our registrations performed well in all the datasets. The method ensured that there were no folding in any of the results. Our method was successful at aligning the lung boundaries, with less than 0.01 % error for any dataset. Although the proposed pipeline also performed reasonably well on the fissure and landmark evaluation, there were still some problematic cases. From visual inspection of the datasets before and after registration, we were able to draw some conclusions about the misalignment. All registrations appeared to align most of the lungs with no errors larger than the voxel size. However, some results contained particular regions, often near the back or base of the lung or near the fissures between different lobes that were misaligned, and leading to worse scores.

The amount of deformation changed greatly between datasets. Examination of the global results showed that for some cases the deformation could be reasonably well approximated by an affine transformation, whereas for others the deformation varied locally from one region to another, and could not be well approximated by an affine transformation. Datasets with the largest local variation tend to have the poorer results; in particular datasets 7 and 14, but also to a smaller degree datasets 1, 18, and 20. In these datasets some regions of the lungs expanded/contracted considerably (Fig. 5.3). These deformations induced changes in the density of the lung tissue and hence changes in intensity in the CT images. Visual examination of the results indicate that when there were large intensity changes in one region but not in another region close by (e.g. adjacent lobes) the registrations failed to recover the correct deformation (Fig. 5.3). This appeared to be the main cause of most of the errors that were observed. Other factors that appeared to contribute to the registration errors in some datasets were some relatively homogenous regions of the images which had few features to guide the registrations and sliding between adjacent lobes (Fig. 5.3).



Floating image *Reference image* *Result image*

Figure 5.3: Comparing the floating and reference images it can be seen that the lower lobe is deforming much more than the upper lobe, some regions contract more than others (such as near the back of the lung and just above the fissure) causing a larger increase in CT intensity in these regions, and that the two lobes are sliding past each other along the fissure.

As already mentioned the submitted results were performed using the CPU-based implementation. Table 5.2 shows that the CPU registration times are acceptable for the on-site registrations, with all 20 registrations taking just over 6 hours. Table 5.2 shows that the GPU-based implementation achieved a speed up for all the stages of the registration where it was used, especially for stage 3, which was the most computationally demanding stage, and where a speed up of greater than 7.8 times was achieved. The average total registration time was just under 6 minutes for the GPU-based implementation, so it was possible to register the 10 datasets on-site in approximately 1 hour, well within the 3 hour time limit. It should be noted that one of the datasets (dataset 2) was too large for final local registration stage to be run on the GPU. This could be detected prior to running the registration, and when the memory requirements was too great for the GPU-based implementation, the CPU-based implementation was used instead.

Better registration accuracy might be possible by modulation of the image intensities according the volume change, as suggested by Yin *et al.* [110]. The use of a locally weighted metric [38] or a local similarity measure [2] might also improve the registration result. To account for sliding between different lobes, the lobes could be segmented and registered separately [111]. We intend to investigate each of these approaches to improve the registrations in the future.

5.2 Differential bias field correction

Magnetic resonance (MR) images are corrupted by smoothly varying intensity nonuniformity (INU) or 'bias' caused by: inhomogeneity of the RF excitation field, spatially nonuniform receiver coil sensitivity profiles, induced currents and 'standing wave effects' or field-focussing [112]. INU has little influence on human interpretation of images, but can severely impact quantitative analysis. The bias is typically modelled as a multiplicative gain-field, with low-order polynomials, low-frequency sinusoidal basis functions, or widely spaced B-splines. However, more severe (and often less spatially smooth) INU occurs with modern acquisitions, because (i) array coils worsen reception uniformity, and (ii) the RF-focussing problem is more severe at field strengths above 1.5 T [113]. Image uniformity can be improved by prospective or retrospective bias correction [114]. Prospective methods relying on images of uniform phantoms cannot compensate for object-specific sources of bias such as field-focusing. Retrospective approaches are hampered by noise and other imaging artefacts, meaning that perfect bias correction is unobtainable.

Serial MR imaging is a powerful means to study longitudinal brain changes due to development, disease or drug treatment [115]. However, residual, uncorrected INU is easily mistaken for biological change by algorithms including the boundary shift integral (BSI) [116] and nonrigid registration based quantification of change [117].

Differential bias correction (DBC) was introduced by Lewis and Fox [118] specifically to remove residual biases not shared by serial image pairs. DBC differs from INU as it aims at creating similar bias field on both images. DBC was shown to reduce misidentified BSI volume change in same-day scan pairs where no true change had occured. However, a small reduction in atrophy in Alzheimer's disease (AD) patients was also observed, suggesting that DBC had also removed some true intensity variation due to atrophy [118]. This is understandable, since their algorithm estimated differential bias from rigidly registered scans, which clearly also contain intensity differences due to atrophic tissue shifts. Naturally, one would expect DBC to perform better after non-rigid registration — an approach similar to Studholme *et al.*'s single-image bias correction using non-rigid registration to an approximately INU-free template image [119].

However, the two problems are inter-related: high degree-of-freedom registration can itself be confounded by the presence of differential bias. Anderson et al. [1][3] developed a novel simultaneous optimisation of a spatial transformation and an INU correction in terms of a sum-squared differences (SSD) cost function. While SSD is suitable for registration of consistently acquired MR images, even minor differences in scanner or pulse sequence could make this criterion sub-optimal.

Studholme et al. [120] and Loeckx et al. [38] independently propose regional modifications of the popular mutual information (MI) similarity measure. Conditioning on intensities within a local spatial Parzen window reduces the problem of INU, but also reduces the number of samples from which to estimate joint histograms and entropies, hence reducing the reliability of the measure. For this reason, Zhuang et al. [121] proposed combining global and local estimates of joint probability densities. In contrast, here we suggest that by explicitly modelling the differential bias field, standard global MI or normalised mutual information (NMI) can be used for registration.

Because INU blurs an image's intensity histogram [112], several bias correction algorithms have been developed that use entropy to quantify the desired histogram re-sharpening [114].

Lewis and Fox [118] observed that their DBC algorithm might have potential to use 'joint intensity histogram sharpening', and Vovk et al. [114] also suggested that 'simultaneous information theoretic registration and inhomogeneity correction are well worth further exploration', but we are unaware of any other work that follows these suggestions.

We therefore propose simultaneous modelling of spatial deformation and differential INU, with a unified model fitted by optimising a single NMI objective function.

5.2.1 Iterative process based on the NMI

In this work, we model the differential bias between the reference and deformed floating image by multiplying the reference image by the exponential of a scalar field $B(\vec{x})$. The exponential form is used in order to enforce strictly positive values without requiring a more complex constrained-optimisation algorithm. The field is parametrised using a second regular lattice of control points $\{b_{lmn}\}$ overlaid on the reference image, where l, m and n are the control point indexes respectively along the x-, y- and z-axis. For each voxel $\vec{x}$ in the space of the reference image R, we can compute:

$$B(\vec{x}) = \sum_{l,m,n} \beta^3\left(\frac{x}{\delta_x^b} - l\right)\beta^3\left(\frac{y}{\delta_y^b} - m\right)\beta^3\left(\frac{z}{\delta_z^b} - n\right) b_{lmn}, \tag{5.3}$$

where β^3 is a cubic B-spline basis function and the spacings between control points are δ_x^b, δ_y^b and δ_z^b.

In order to align the input images and simultaneously model the bias field, the deformation model parameters $\{\vec{\mu}_{ijk}\}$ and bias parameters $\{b_{lmn}\}$ are adjusted to optimise an objective function. The objective function is a balance between the Normalised Mutual Information (NMI) (Eq. 5.4) as a similarity measure and the bending-energy as a penalty term. The penalty term is applied to both spline fields, favoring smooth spatial and intensity transformations. The NMI, which is based on the marginal and joint

[3] See also http://fsl.fmrib.ox.ac.uk/fsl/fnirt

entropies of the images, aims to maximise the amount of information that one image has about another.

$$\text{NMI} = \frac{H(e^B \times R) + H(F(\mathbf{T}))}{H(e^B \times R, F(\mathbf{T}))} \tag{5.4}$$

In order to compute the joint and marginal entropies, a joint histogram $\mathbf{H}$ (Eq. 5.5) is computed using a Parzen Window approach [35]:

$$\begin{aligned} \mathbf{H}(r,f) &= \sum_{\vec{x} \in R} \beta^3 \left(e^{B(\vec{x})} \times R(\vec{x}) - r \right) \\ &\times \quad \beta^3 \left(F\left(\mathbf{T}(\vec{x})\right) - f \right) \end{aligned} \tag{5.5}$$

where r and f are bin intensities for the reference and floating image and the Parzen window function is the cubic B-spline function β^3.

The objective function is optimised using a conjugate gradient descent. To perform such optimisation scheme, the derivative of the objective function has to be computed with respect to both the degrees of freedom of the spatial transformation and of the bias field, as in equation 5.6:

$$\frac{\partial \text{NMI}}{\partial b_{ijk}} = \frac{\frac{\partial H(e^B R)}{\partial b_{ijk}} + \frac{\partial H(F(\mathbf{T}))}{\partial b_{ijk}} - \text{NMI} \times \frac{\partial H(e^B R, F(\mathbf{T}))}{\partial b_{ijk}}}{H(e^B R, F(\mathbf{T}))} \tag{5.6}$$

The gradient of the bending-energy has been detailed in the previous chapter, we will thus only focus on the NMI. In order to compute the similarity measure derivatives, the gradients of the joint and marginal entropies have to be evaluated. To do so, the derivative of the joint histogram $\mathbf{H}$ has to be calculated. It can be done for the spatial transformation such as:

$$\begin{aligned} \frac{\partial \mathbf{H}(r,f)}{\partial \mu^{\xi}_{ijk}} &= \sum_{\vec{x} \in R} \beta^3 \left(e^{B(\vec{x})} \times R(\vec{x}) - r \right) \\ &\times \quad \left. \frac{\partial \beta^3 (\mathcal{F} - f)}{\partial \mathcal{F}} \right|_{\mathcal{F} = F(\mathbf{T}(\vec{x}))} \\ &\times \quad \left. \frac{\partial F(\vec{p})}{\partial \vec{p}} \right|_{\vec{p} = \mathbf{T}(\vec{x})} \frac{\partial \mathbf{T}(\vec{x})}{\partial \mu^{\xi}_{ijk}} \end{aligned} \tag{5.7}$$

where $\xi \in \{x, y, z\}$ refers to a single component of the vector for control point $\vec{\mu}_{ijk}$. The derivative according to the DBC parameters can be computed such as:

$$\begin{aligned} \frac{\partial \mathbf{H}(r,f)}{\partial b_{lmn}} &= \sum_{\vec{x} \in R} e^{B(\vec{x})} \times R(\vec{x}) \\ &\times \quad \left. \frac{\partial \beta^3 (\mathcal{B} - r)}{\partial \mathcal{B}} \right|_{\mathcal{B} = R(\vec{x}) e^{B(\vec{x})}} \\ &\times \quad \frac{\partial B(\vec{x})}{\partial b_{lmn}} \times \beta^3 \left(F\left(\mathbf{T}(\vec{x})\right) - f \right) \end{aligned} \tag{5.8}$$

5.2.2 Application to simulated brain data

As a first experiment, we performed registrations of an image to itself after an inhomogeneity field has been added. The inhomogeneity field I used is a realistic field available as part of the brain web database[4]. We here used an image from the brainweb database as it has the advantage of being free from any initial bias field. Such a registration has been performed both with and without a DBC field using a 20 mm spacing between the control points. The spatial control point spacings, δ_x^μ, δ_y^μ and δ_z^μ, have been set to 5 mm. All registrations have been performed using a multi-level approach with 3 steps, the weight of the spatial bending-energy penalty term has been set to 1%, whereas the DBC bending-energy weight was 10%. Only the bias field differs between the two input images, the spatial deformation was thus desired to be null for every voxel.

Figure 5.4 presents a box-plot representation (showing percentiles 0, 25, 50, 75 and 100) of the amplitude of displacement.

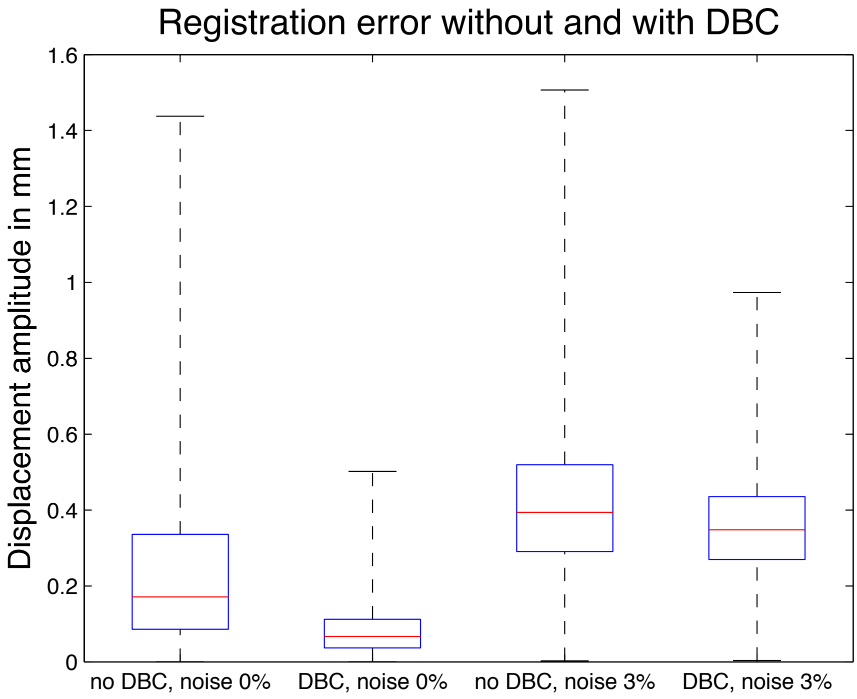
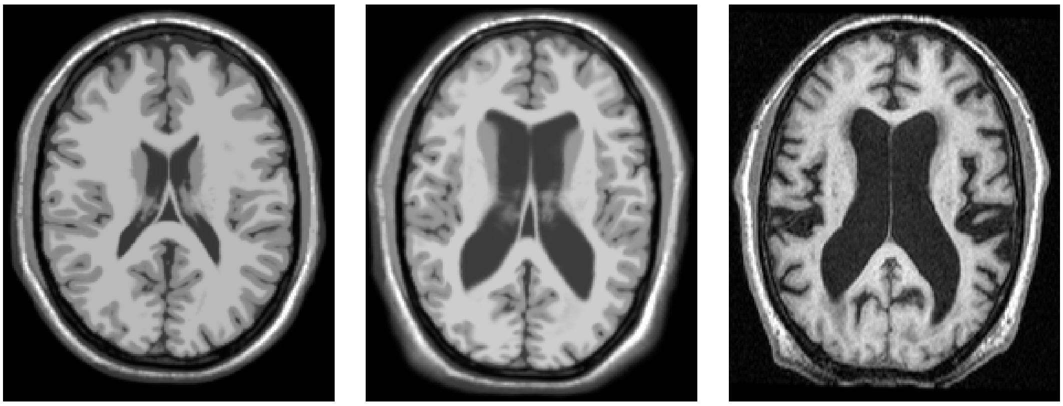


Figure 5.4: Amplitude of recovered displacement in the absence of true spatial deformation. Box-plot of errors over voxels.

For the second experiment, we deformed a bias field free image from the brainweb database 40 times, to match 40 target images. These target images consist of 20 Alzheimer's disease (AD) patients and 20 age-matched controls. To perform these registrations, we used the diffeomorphic demons algorithm [47]. Figure 5.5 illustrates these deformations and shows their amplitude. We then try to recover these deformations after adding the BrainWeb 20% bias field to one of the images. We compute the root mean square (RMS) error between the known simulated deformation field and the computed deformation field.

[4] http://www.bic.mni.mcgill.ca/brainweb/



Floating image

Warped image

Reference

Figure 5.5: Example of simulating known deformation, using the diffeomorphic demons algorithm.

Figure 5.6 presents the error for each registration, without and with the DBC. In order to improve the visibility of the majority of results, the figure's y-axis has been truncated, obscuring one subject's RMS errors outside the displayed range.[5] A paired t-test assessing the influence of the DBC reports a significant improvement (p-value 0.0028). The improvement of 0.0851 mm (95% confidence interval of [0.0311, 0.1391]) corresponds to 25.9% of the average RMS error without DBC.

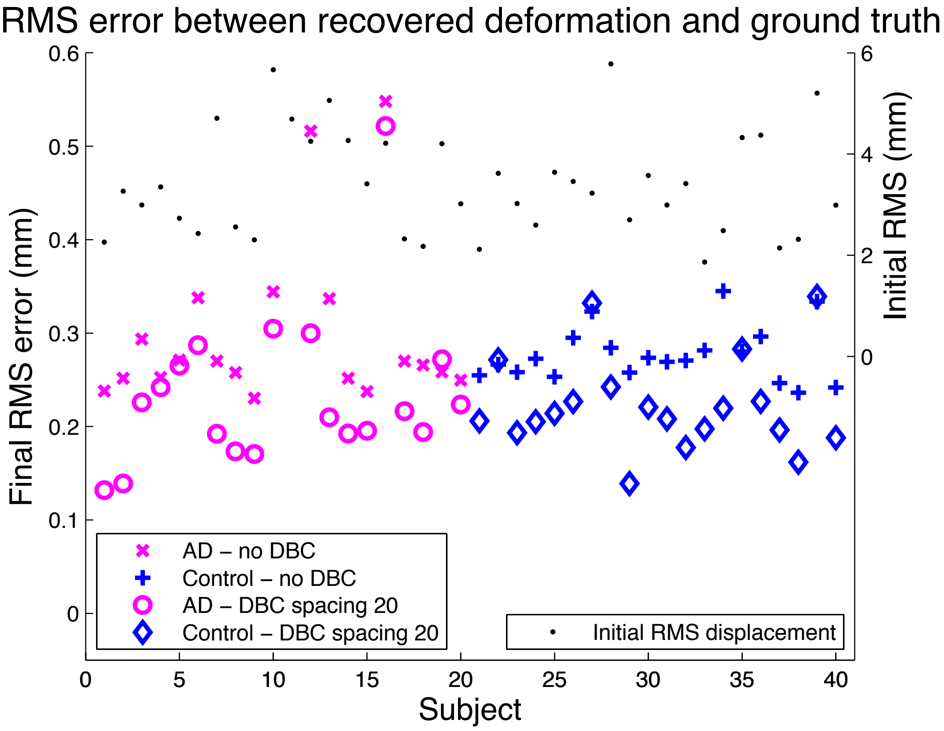


Figure 5.6: Influence of the DBC in recovering known deformation fields. Errors evaluated over 40 registrations.

The first experiment showed the desired attenuation of displacement when performing the registration with the DBC. It can be seen that the amount of noise in the input images affects the quality of the

[5] Subject 11: RMSE(no DBC) = 1.9075; RMSE(DBC) = 0.8182.

result; further work will aim at improving our algorithm's robustness to noise. As the displacements are lower when the DBC is used, one might speculate that the DBC could effectively be over-regularizing the deformation by modelling some differences that should be modelled by the spatial transformation. The second experiment therefore investigated the reproducibility in recovering a known simulated deformation. The obtained results show a statistically significant reduction in RMS error compared with the results obtained using the same algorithm without DBC. Due to the initial large amplitude deformation, shown by the initial RMS magnitude on the right of figure 5.6, we can conclude that the DBC does not act to over-regularize the registration.

Future work would involve further experiments on simulated data with different noise levels, and on real longitudinal data-sets. We will compare our novel method to the original DBC [118] and to multiple separate applications of conventional bias correction algorithms.

We presented a combined information-theoretic DBC and non-rigid registration, implemented in the FFD framework. The DBC is optimised using the analytical derivative of the NMI according to the DBC parameters and regularized with a bending energy penalty term. The bias field parameters are the control points of a cubic B-spline lattice overlaid on the reference image. We showed that our implementation significantly reduced the influence of a differential bias field while performing non-rigid registration. This improved registration of serial images should reduce sample sizes needed to detect longitudinal change using imaging-based biomarkers. Due to the use of the NMI, the combined registration and DBC technique should be widely applicable.

Chapter 6

A bio-mechanically constrained parametric registration algorithm

When studying brain images using non-rigid registration, the determinant of the Jacobian provides a measure of local volume change that is often of interest for quantifying deformations over time or between subjects. However, as each registration method produces a slightly different transformation (and equally importantly, via a different deformation mechanism) the Jacobian determinant maps vary both quantitatively and qualitatively. Moreover, the quality of the map (judged directly by clinicians, or indirectly via results of tensor-based morphometry) is not necessarily correlated with the quantitative accuracy of the registration. For example, using different techniques such as the Free-Form Deformation [29] (FFD), the fluid [42], the diffeomorphic demons algorithm [47] or symmetric normalization (Syn) [2], different Jacobian determinant maps are obtained even though the warped images all match the reference — see Fig. 6.1.

In order to generate smooth and plausible transformation with the FFD method, efforts have been made to impose constraints on the deformations. Rueckert *et al.* [29] proposed a penalty term based on the bending energy. Rolhfing *et al.* [69] presented another based on the logarithm of the Jacobian determinant. The Jacobian determinant was also embedded in a regularizer by Sdika [68]. However, simple constraints or penalty terms are either incapable of modelling large deformations or unable to prevent highly variable (or negative) Jacobians. Considering that the general aim of the above penalty terms is to favour physically plausible deformations, a natural alternative is to directly include a biomechanical regulariser, for example based on equations of continuum mechanics. Linear elastic registration has been used since the 1980s [122, 123], however, linearity breaks down for large deformations, limiting the flexibility of such methods. Fluid-mechanical regularisation allows large deformation without discontinuities, but also permits unrealistically severe distortions. This paper argues in favour of a nonlinear elastic regulariser coupled with a spline model, that should handle large but realistic deformations while maintaining an anatomically reasonable Jacobian map.

Yanovsky *et al.* [124] also investigated nonlinear elasticity. They developed a variational form which coupled similarity and elasticity functionals, using a linear strain energy function (Saint Venant-Kirchhoff model), and solved the system using finite differences. The development and solution of the

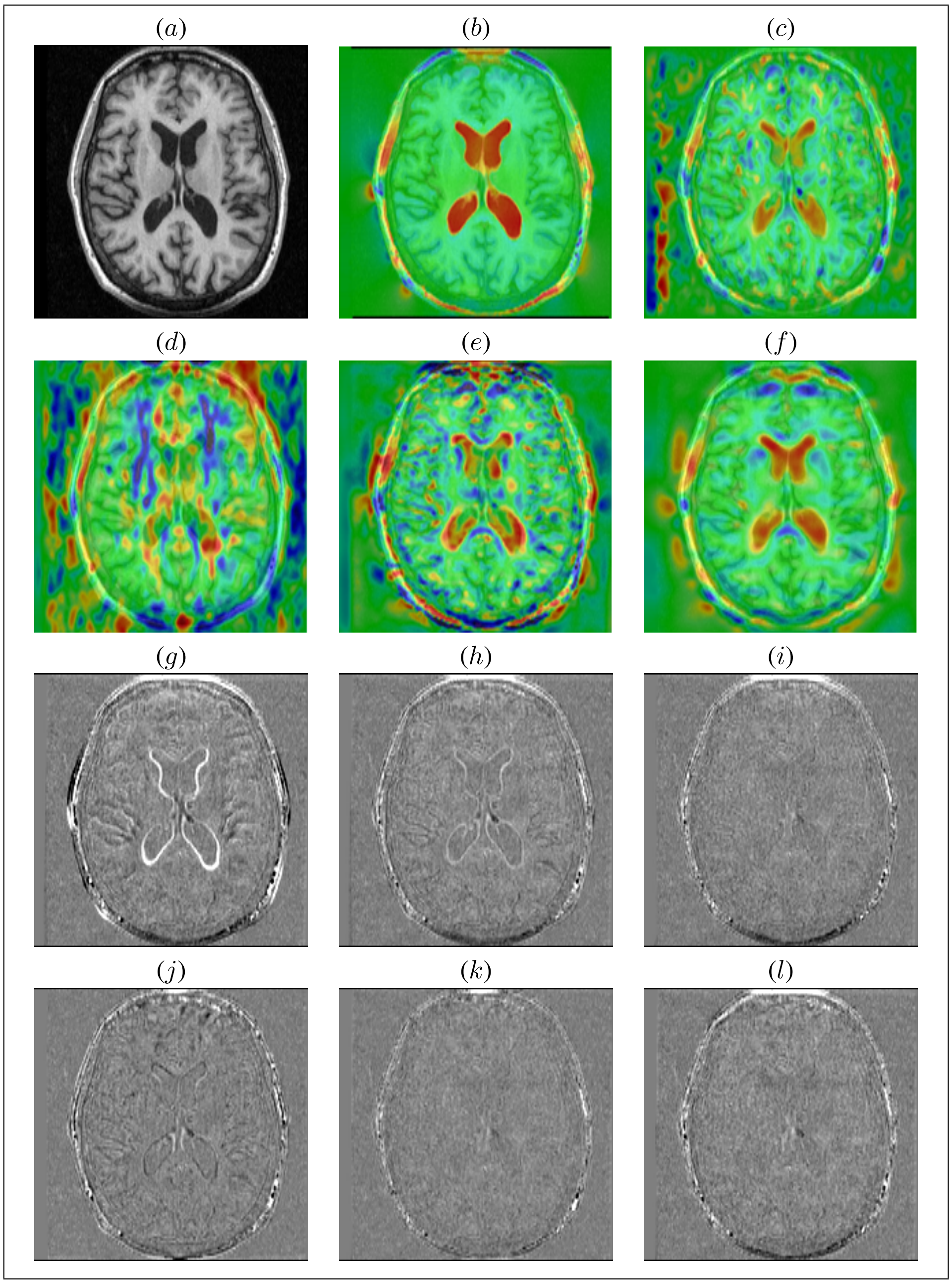


Figure 6.1: Variation in volume change distribution with different registration algorithms. A floating image has been registered to a reference image (a) using: fluid (b,h), SyN (c,i), demons (d,j), free-form deformation (e,k) and the proposed method (f,l). It can be appreciated from the difference images (bottom row) that all techniques successfully recovered the initial differences (f). However the Jacobian determinant maps (top row) reveal very different patterns of deformation. $\log_2(\det(J))$ is shown with colour range from -0.5 to 0.5).

coupled system was facilitated by an approximation for the material displacement derivatives.

I present a decoupled regularisation of the FFD algorithm using nonlinear elasticity. Solution of the equations of continuum mechanics is performed using the finite element method, which requires no approximation of the deformation components, and allows for incorporation of elaborate constitutive models. The deformation model is linked to an appropriate similarity metric by so-called *pseudo-forces* derived from the metric's gradient. The scheme is shown to produce both accurate and smooth deformation fields. I emphasise that in employing a continuum mechanics-based model my aim, in this case, is to produce physically consistent smooth transformations, not to model the physiology of the disease process itself; I do not claim, for example, that deformations associated with tissue loss are directly analogous to mechanical compressions.

In Section 1 the methods I employed is described, and in Section 2 I present three separate evaluation experiments, comparing the new method to the *classical* FFD algorithm on longitudinal MRI of Huntington's disease.

6.1 Method

6.1.1 Deformation model

I consider the floating image volume to be a continuous (but not necessarily homogeneous) elastic body with initial volume V_0. I assume that loading is entirely in the form of body forces $\mathbf{f}^B$. At any point in the body I define the deformation in terms of the Green-Lagrange strain tensor [125]

$$\mathbf{E} = (\mathbf{F}^T\mathbf{F} - \mathbf{I})/2, \tag{6.1}$$

where:

$$\mathbf{F} := \frac{\mathrm{d}\mathbf{x}}{\mathrm{d}\mathbf{X}} \tag{6.2}$$

is the deformation gradient, $\mathbf{I}$ is the second order identity tensor, and $\mathbf{x}$ and $\mathbf{X}$ are current and initial material point coordinates, respectively. Standard results from continuum mechanics dictate that deformations within the image volume must satisfy the equation of virtual work [126]:

$$\int_{V_0} \mathbf{S}\,\delta\mathbf{E}\,\mathrm{d}V = \int_{V_0} \mathbf{f}^B\,\delta\mathbf{u}\,\mathrm{d}V, \tag{6.3}$$

where $\delta\mathbf{E}$ are strain variations corresponding to virtual displacements $\delta\mathbf{u}$, and the left and right hand sides represent internal and external virtual work terms, respectively; Eqn. (6.3) is an equilibrium equation. $\mathbf{S}$ are second Piola-Kirchhoff stresses, which are related to the strains through the constitutive model [125]:

$$\mathbf{S} = \frac{\partial\Psi}{\partial\mathbf{E}}, \tag{6.4}$$

where Ψ is an non-linear elastic strain energy function. I note that use of kinematically consistent stress and strain measures means this formulation is valid even for large deformations. By seeking deformations of the image volume which satisfy these equilibrium and constitutive constraints I guarantee that a physically plausible transformation is obtained.

Eqn. (6.3) may be solved for the deformation field throughout the image volume using the finite element method (FEM) [126]. For simulation of large deformations a formulation capable of accommodating geometric nonlinearities must be used. I employ a total Lagrangian explicit dynamic (TLED) algorithm [127], which has been shown to be highly efficient for solving nonlinear soft tissue deformation problems [128, 127, 129]. The image volume is discretised into a regular octohedral mesh, similar to the grid used for B-spline-based methods [29], wherein grid points constitute finite element nodes. Each node has three displacement degrees of freedom, and I employ 8-node octohedral elements with trilinear interpolation and reduced integration [126]. Via standard methods [126] this discretisation renders Eqn. (6.3) into the following system of differential equations

$$\mathbf{M}\ddot{\mathbf{U}} + \mathbf{C}\dot{\mathbf{U}} + \mathbf{K}\left(\mathbf{U}\right).\mathbf{U} = \mathbf{R}, \tag{6.5}$$

where $\mathbf{M}$ and $\mathbf{C}$ are mass and structural damping matrices, respectively, $\mathbf{K}(\mathbf{U})$ is the system stiffness matrix, which depends on nodal displacements $\mathbf{U}$, and $\mathbf{R}$ are external loads. The over-dot notation denotes time-derivatives. Appropriate boundary conditions (nodal displacement constraints) are enforced (see Sect. 6.1.3), and loads in the form of pseudo-forces derived from the gradient of the employed similarity metric (see Sect. 6.1.2) are applied. Solutions (nodal displacements $\mathbf{U}$) to Eqn. (6.5) are then computed incrementally in time using the procedure detailed in [127, 129].

Once the nodal displacements have been computed for the current loading, interpolation is used to obtain a continuous deformation field. In this final step, rather than the trilinear interpolation functions of the finite elements themselves, I use a cubic B-spline scheme; the C^2 continuity of the deformation $\mathbf{T}$ ensures smoothly varying first derivatives and hence a smooth Jacobian map, $\det_{Jac} = |\nabla\mathbf{T}|$. The elastic model thus constitutes a regulariser for the B-spline model. Note that the Jacobian in this formula is different from $\mathbf{F}$, as it is based on the B-spline interpolation model. Other interpolation schemes may be more applicable for different applications.

6.1.2 Metric and optimisation

To evaluate the quality of the registration and optimise the node positions, I compute the Normalised Mutual Information (NMI) between the reference R and the deformed floating image $F(\mathbf{T})$. NMI is a voxel intensity-based information-theoretic similarity measure [37], which quantifies the shared information of the two images. It is defined as

$$\text{NMI} = \frac{H(R) + H(F(\mathbf{T}))}{H(R, F(\mathbf{T}))} \tag{6.6}$$

where $H(R)$ and $H(F(\mathbf{T}))$ are the marginal entropies of images R and $F(\mathbf{T})$, and $H(R, F(\mathbf{T}))$ denotes their joint entropy. The computation of each (Shannon) entropy $H = -p(e)\log(p(e))$, is based on the probabilities $p(e)$ of events derived from a joint histogram $\mathbf{H}$. This histogram indicates the probability of each combination of intensities in images R and $F(\mathbf{T})$. In order to fill the histogram I used the Parzen Window technique. This technique has been presented as more accurate than the generalised partial volume method as the joint histogram is less populated near the optimium [130]. Considering r and f as voxel intensities respectively in the reference image and the deformed floating image, the joint histogram $\mathbf{H}$ is filled as

$$\mathbf{H}(r, f) = \sum_{\mathbf{x}\in\Omega} \beta_r^3(R(\mathbf{x}); r)\beta_f^3(F(\mathbf{T}(\mathbf{x})); f) \tag{6.7}$$

where R is defined over the Ω domain and β_r^3 and β_f^3 are intensity kernels based on cubic splines.

To drive the displacement of the nodes I computed the gradient of the NMI at every node position. It is possible to compute such values for every node using the derivatives of the marginal and joint entropies:

$$\frac{\partial \mathrm{NMI}}{\partial \mu^{\xi}_{i,j,k}} = \frac{\frac{\partial H(R)}{\partial \mu^{\xi}_{i,j,k}} + \frac{\partial H(F(\mathbf{T}))}{\partial \mu^{\xi}_{i,j,k}} - \mathrm{NMI} \times \frac{\partial H(R,F(\mathbf{T}))}{\partial \mu^{\xi}_{i,j,k}}}{H(R, F(\mathbf{T}))} \tag{6.8}$$

These entropy derivatives are calculated by taken into account the deformation model $\mathbf{T}$ to fill the derivative of the joint histogram:

$$\frac{\partial \mathbf{H}(r,f)}{\partial \mu_{i,j,k}} = \sum_{\mathbf{x}\in\Omega} \beta^3_r(R(\mathbf{x});r)\, \frac{\partial \beta^3_f(v;f)}{\partial v}\bigg|_{v=F(\mathbf{T}(\mathbf{x}))} \frac{\partial F(\mathbf{T}(\mathbf{x}))}{\partial p}\bigg|_{p=\mathbf{T}_x} \frac{\partial \mathbf{T}(\mathbf{x})}{\partial \mu_{ijk}} \tag{6.9}$$

This approach provides the mathematical value of the gradient but involves significant computation redundancy, since every voxel is included in the neighborhood of several control points. Moreover it is memory intensive as each node requires one joint histogram per degree of freedom. In order to decrease this redundancy and the memory requirement, I propose a voxel-centric approximation of the node-centric gradient. I first compute the gradient value for every voxel, then gather the information from all voxels to obtain the nodal gradient values.

I computed the voxel-centric gradient values using the formulas in equations 6.8 and 6.9 where $\frac{\partial \mathbf{H}(r,f)}{\partial \mathbf{u}^{\xi}_{\mathbf{z}}}$ is computed by replacing $\frac{\partial \mathbf{T}(\mathbf{x})}{\partial \mu_{ijk}}$ with $\frac{\partial \mathbf{T}(\mathbf{x})}{\partial \mathbf{u}^{\xi}_{\mathbf{z}}}$ where $\frac{\partial \mathbf{T}(\mathbf{x})}{\partial \mathbf{u}^{\xi}_{\mathbf{z}}} = 1$ if $\mathbf{z} = \mathbf{x}$ as $\mathbf{T}(\mathbf{x}) = \mathbf{x} + \mathbf{u}(\mathbf{x})$.

In order to provide one gradient per node I weighted the gradient of each voxel such that voxels close to a node had more impact than voxels further away. However, weighting every voxel in the neighborhood of one node would lead to extra computation because of redundancy, as before. To avoid this, I applied a convolution window to the gradient field and so approximated the gradient for every node. The chosen convolution window was a cubic B-Spline curve which matched the basis functions in the deformation model in terms of node spacing; it was equivalent to $\frac{\partial \mathbf{T}(\mathbf{x})}{\partial \mu_{ijk}}$ in equation 6.9.

To optimise the tranformation I normalised the NMI gradients of all nodes and applied them as external forces in the TLED solver. Each time the solver was run the floating image was resampled and the metric value re-evaluated. A conjugate gradient ascent was then performed to find the external forces which best transformed the floating image in the direction of the gradient. The gradient was then recomputed and the line ascent re-performed. This loop iterated until no improvement superior to 0.1% of the similarity measure was produced.

6.1.3 Framework

I implemented the proposed algorithm for graphics processing unit (GPU) execution using the CUDA API from NVidia [102]. The proposed framework can be decomposed into four modules, as presented in figure 6.2.

Module 1: TLED solver.

The first module concerned the TLED solver. As described by Taylor *et al.* [129], the solver consisted of precomputation and online components. Since I deal with regular meshes in this application (and homogeneous material properties in the first instance), all finite elements have the same properties, which

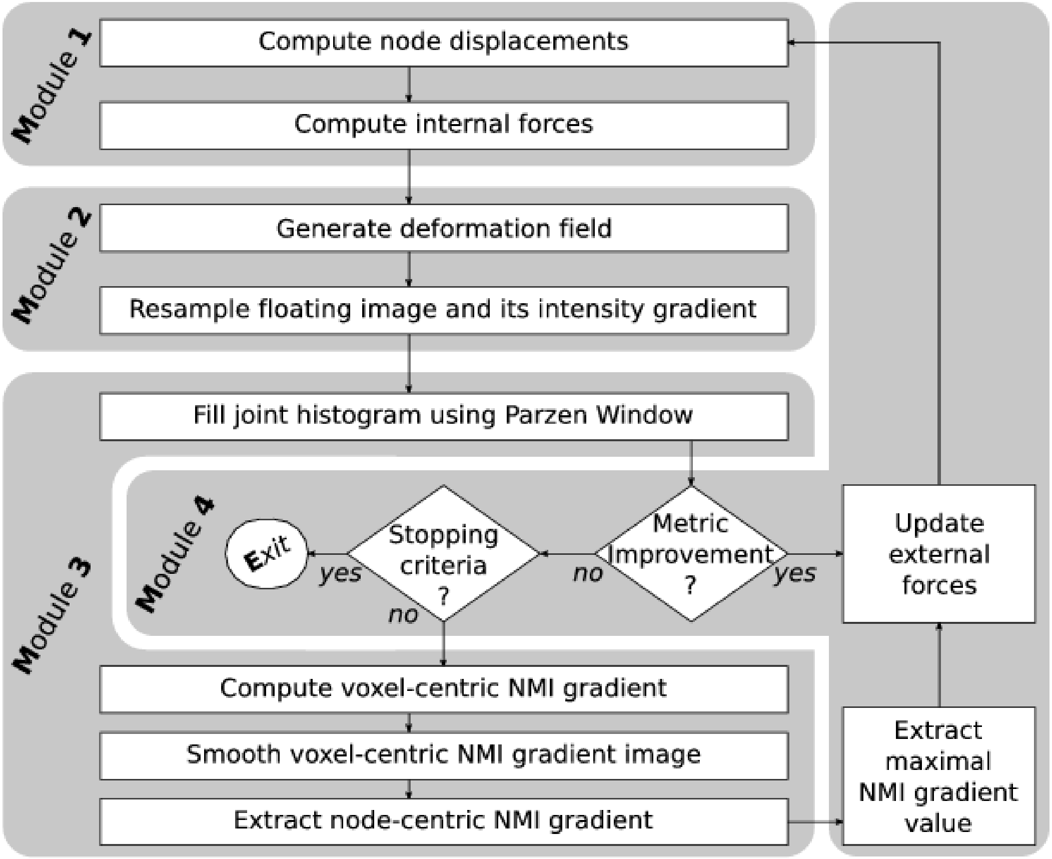

Module 1
Compute node displacements
Compute internal forces
Module 2
Generate deformation field
Resample floating image and its intensity gradient
Module 3
Fill joint histogram using Parzen Window
Module 4
Exit
yes
Stopping criteria ?
no
no
Metric improvement ?
yes
Update external forces
Compute voxel-centric NMI gradient
Smooth voxel-centric NMI gradient image
Extract node-centric NMI gradient
Extract maximal NMI gradient value

Figure 6.2: Framework of the presented algorithm

simplifies the computation and considerably decreases memory requirements; variables for only a single element need be precomputed and stored. Node positions and element geometries were computed on-the-fly according to image size and the user-defined node spacing.

The solution time step was estimated from $\Delta t_c = L_e/c$, where L_e is the characteristic element length and c is the dilatational wave speed of the material [126]. This formula provides an estimate of the stable time step in explicit dynamic analysis, assuming linear elasticity. I employed both kinematic and constitutive nonlinearities and consequently used a conservative time step twice smaller than Δt_c.

To avoid any free displacement of the volume, some points must be fixed. For each registration I generated a brain mask using BET [105] and dilated it. All nodes outside of the mask were then fixed.

The GPU computation consisted in two kernels. In the first the internal nodal forces were computed, and in the second the resulting nodal displacements were computed — see [129] for details.

Module 2: resampling of the floating image.

The second module dealt with the deformation and resampling of the floating image, and also comprised two kernels. The cubic B-spline interpolation was calculated for every voxel in the reference image in the first kernel. Once the deformation field was generated the floating image was resampled using a second kernel. The latter kernel also returned the intensity derivatives of the deformed image. These were required in the next module for the NMI gradient computation. The deformation field interpolation and the trilinear resampling of the floating image were split into different functions in order to increase the occupancy, and hence performance, of the GPU computation. The occupancy is the ratio between the number active computation threads on the GPU and the maximal number of threads that are available on the graphical card.

Module 3: metric and gradient calculation.

The third module consisted in the calculation of the NMI and its gradient. The NMI computation result was observed to be sensitive to the floating point precision used. Hence, the computation was performed on the CPU using double precision. As a consequence, after its resampling, the deformed image was transferred to the CPU memory. During this computation the logarithm of each probability was computed. As their computation for the NMI gradient would be redundant I stored each in a second histogram and subsequently transferred this back to the GPU memory. The NMI gradient was computed in a single kernel on the GPU, after which a series of other kernels were invoked for convolution window creation and gradient field smoothing. A last kernel then extracted the gradient value for each node.

Module 4: external force optimisation.

The last module was concerned with the update of the pseudo-forces and their optimisation. The maximal norm of the gradient was first extracted using a spread and gather approach, then the pseudo-forces were updated as described in the previous section.

Multi-scale framework

To improve the efficiency of the elastic model I developed a multi-level mesh approach. This allowed the system to recover gross deformations more quickly, and also helped in avoiding local minima. Beginning with a coarse level mesh (l_n) the deformation field was optimised as described. The next (denser) level mesh (l_{n+1}) was then obtained by subdividing elements in each dimension. Thus, I required the input forces for l_{n+1} which would reproduce the deformation field of l_n. These were computed as follows: (1) the deformed positions of new nodes were obtained by linearly interpolating those of existing nodes; (2) for this configuration, the nodal force contributions from each element were computed using kernel 1, module 1 (see [129] for details); (3) these force contributions were summed at each node to give the required l_{n+1} input forces. By construction, these inputs exactly balance the elastic forces in the desired configuration, and were used as the start values in the l_{n+1} optimisation scheme.

6.2 Evaluation

6.2.1 Data and Methods

The methods are evaluated on serial MR images of 33 patients with early Huntington's Disease (HD) and 14 healthy age- and gender-matched control subjects, imaged at baseline and 12-month follow-up; 23 and 9 of the respective groups were also scanned after 24 months. Three-dimensional T1-weighted MR images with 1.5 mm coronal slices of in-plane dimension 0.9375×0.9375 mm were acquired with a spoiled GRASS sequence at 1.5 T. Each follow-up image was registered to its baseline using an affine algorithm followed by each non-rigid algorithm. I also registered the 24-month follow-ups to the 12-month scans. The multi-scale approach used 3 levels, with the finest having 2.5 mm isotropic spacing between nodes. The TLED-solver used a Poisson ratio of 0.1. The FFD used a bending energy penalty term with a relative weight of 10%.

To derive inter-subject correspondence, I used the group-wise diffeomorphic registration algorithm DARTEL [3]. We used DARTEL in order to not introduce bias toward our deformation mode. The resulting transformations were used to spatially-normalise the Jacobian maps, and to inverse-normalise semi-automatic lateral ventricular and intra-cranial segmentations of the DARTEL average back to the original images. DARTEL uses a very different transformation model (exponentiation of velocity fields) to those evaluated here, thus helping to avoid bias.

6.2.2 Experiments

Validation of non-rigid registration algorithms is a challenging problem [131]. Direct measurement of correspondence errors [132] relies on time-consuming and error-prone manual identification of corresponding landmarks. Furthermore, unambigous landmarks may only be found in certain locations, away from which errors cannot be reliably determined. Using overlap indices of automatic registration-propogated segmentations and manually performed labellings [131] is also operater-dependent, and provides no information on the behaviour of the transformation inside the labelled objects. These two approaches may also be biased in favour of algorithms driven by landmark-matching or intensity differences respectively, e.g. feature-based methods such as HAMMER [64] may appear more successful in terms of matching manually identified landmarks if similar points are considered distinctive by both human and computer vision systems; conversely intensity-based methods could match (MR-visible) boundaries almost perfectly, while misaligning underlying structural homologies that require expert or contextual knowledge to infer [49].

Attempts have been made to quantify performance via direct comparison of estimated displacement fields [133] or Jacobian maps [134] on images related by simulated and hence known transformations. The key advantages of this are greater objectivity, and the potential for dense voxel-wise measurement of error. However, such a method is clearly only as valid as its simulation model. For physical deformation of breast images [133] a biomechanical FEM model should provide an excellent gold standard. For phenomenological modelling of brain atrophy [134] simulation seems well-suited to evaluating regional or global volume changes, but severely limited for the present application — evaluation of different regularisation approaches at the scale of individual voxels — since the simulation model (which only approximates an unknown biological model) cannot help but bias the evaluation towards similar physical regularisation models. In particular, given the clinical desire for anatomically plausible Jacobian maps, the strong influence of the physical model on the character of the Jacobians of the simulated transformations is problematic. In the hope of overcoming these challenges, I present a sophisticated validation strategy comprising three complementary experiments, described now.

Longitudinal consistency. It is clearly desirable for a registration algorithm to recover equivalent correspondences whether registering follow-up source images to their baseline targets or vice versa — the principle of inverse-consistency [57]. For the specific purpose of comparing regularisation methods, simple $(\mathrm{A} \leftarrow \mathrm{B}) \circ (\mathrm{B} \leftarrow \mathrm{A})$ consistency is flawed, because increasingly strong regularisation will tend towards the ‘perfect’ but trivial consistency of the identity. I argue here that this limitation can be ameliorated by using three time-point longitudinal imaging to evaluate the discrepancy between the composition of the two 12-month interval transformations $(A \leftarrow B) \circ (B \leftarrow C)$ and the direct 24-month interval registration $A \leftarrow C$. The hope being that overly influential regularisation will prevent the 24-month registration from recovering as much deformation as the two combined 12-month interval transformations, thus restoring merit to the consistency measure, which will still penalise inadequate regularisation that leads to erratic transformations. The discrepancy in mm for each registration algorithm is summarised by the voxel-wise mean over each subject’s intra-cranial mask.

Realism of ventricular changes. To directly address the clinically-motivated question of whether the Jacobian images are biological reasonable, the maps of determinant values are analysed over the segmented lateral ventricle region. I argue that (a) in the homogeneous cerebrospinal fluid (CSF), an ideal registration algorithm should recover homogeneous estimates of volume change, yielding low variance; and (b) in the abscence of gray- or white-matter expansion in either HD or healthy aging in adult subjects, I would expect either stable or expanding ventricles.

Jacobian-based group separation. The first two experiments have been designed to help avoid favouring over-regularised models by including three time-points, and by considering mean ventricular expansion in addition to Jacobian variance. However, to further reduce bias towards constrained transformations, the third experiment is inherently based on quantifying clinically-relevant information, in terms of the registration method’s power to discriminate HD from healthy aging. Unlike the commonest form of dementia, (sporadic) Alzheimer’s disease, HD status is known from genetic testing, providing a genuine ground truth for classification. I use a linear soft-margin Support Vector Machine (similar to that used in [135]), with a nested cross-validation procedure that leaves out each subject in turn, performs an inner leave-one-out loop to optimise the SVM’s C parameter, then classifies the left-out subject, which provides an unbiased estimate of the classification accuracy. The SVM’s kernel consists of the image-based inner-products of the subjects’ log-transformed determinants, meaning that classification accuracy should closely reflect the clinical information in these maps.

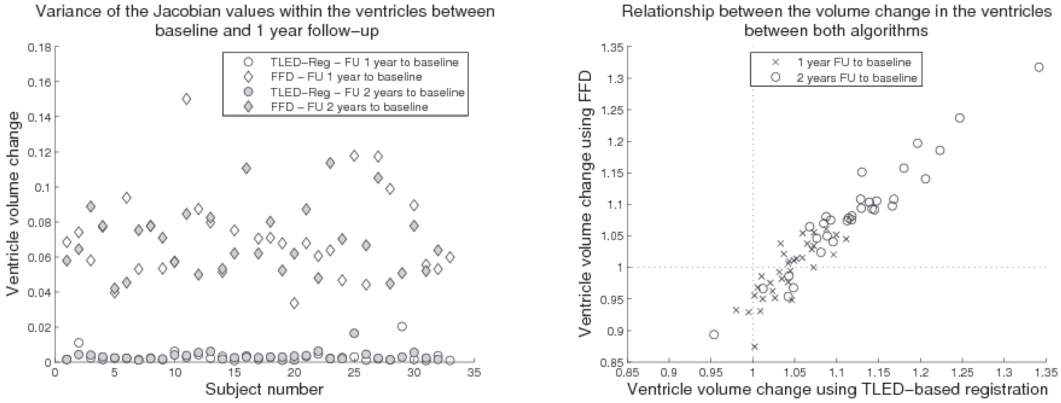
6.3 Results and discussion

Registration performance is summarised in table 6.1. Mean computation times were about 40 minutes per registration for the TLED-based method and approximately 40 seconds for my GPU-based implementation of the FFD: F_3D. Further work might also consist in decreasing the TLED-based registration computation time. The NMI is fractionally higher for the FFD algorithm; the differences being statistically significant when paired over subjects. The FFD algorithm produces widely varying Jacobian values, while the TLED-based method appears to produce more realistically smooth deformation gradients. The nonlinear elastic model has substantially reduced consistency errors compared to the FFD method. Greater consistency at the expense of lower NMI could simply indicate over-regularisation, however, the ventricular measurements indicate that in addition to featuring lower variability (Fig. 6.3), the TLED-based registration measurements actually show greater mean expansion, and are more biologically plausible in terms of having far fewer subjects with erroneously contracting ventricles.

	TLED-based reg		*classical* FFD	
	Follow-up 12 to Baseline	Follow-up 24 to Baseline	Follow-up 12 to Baseline	Follow-up 24 to Baseline
Normalised mutual information	1.2306	1.1908	1.2368	1.1933
Jacobian values range	[0.31 1.92]	[0.43 1.97]	[-1.91 6.93]	[-1.61 6.10]
Consistency mean error	0.29 mm		0.80 mm	
Classification accuracy (%)	74.5	87.5	63.8	71.9

Table 6.1: Summary of quantitative results for registration performance.

The TLED algorithm is more powerful at discriminating HD patients from controls than the classical FFD. A 95% confidence interval [136]. for the increase in the (paired) accuracies on the 12-month interval is $[-2.23\ 23.1]\%$, and $[-3.59\ 33.9]\%$ for the 24-month interval, indicating that the differences are not statistically significant. However, unlike the changes in NMI, improvements in accuracy of circa 10% and 15% would be *clinically* very significant if shown to generalise.



(a)

(b)

Figure 6.3: Comparison of ventricular expansion rates.

6.4 Conclusion

I have presented a novel method for efficient GPU-based non-rigid registration, regularised by a nonlinear elastic model. The most closely related work is [124] in which nonlinear elasticity is also used, but with a Jacobian matrix approximation employed to ease computation. The present approach should also be compared with Rueckert *et al.*'s [70] diffeomorphic version of the cubic spline algorithm, which allows large deformations with strictly positive Jacobian by composing a large number of smaller displacement fields.

A thorough evaluation of the method has been performed, showing that the nonlinear elastic regulariser improves the plausibility of the Jacobian maps while increasing the information they contain for automatic classification of neurodegenerative disease. In return for a slight decrease in NMI, the longitudinal consistency is greatly improved. The speed of the GPU-based nonlinear elastic registration will facilitate application to larger cohorts of images in the future, it should also make feasible more sophisticated regularisation models, for example permitting varying material properties in different brain tissues.

Chapter 7

Fast Free-Form Diffeomorphic Deformation: $\mathbf{F_3D^2}$

Diffeomorphic frameworks for non-rigid registration lead to one-to-one transformations, which preserve topology and ensure invertibility. These features are of great interest as they enable plausible biomedical analysis of volume, shape or rate of change over time. They are, for example, the core of morphometric studies where correspondence between subjects is crucial. Whereas, due to the limited amount of change between time points, longitudinal changes are usually captured using small-deformation based algorithms, differences between patients are larger and include complex shape variability. Several diffeomorphic methods have been presented in the last few years: notably the Large Deformation Diffeomorphic Metric Mapping (LDDMM) algorithm [59], Dartel [3], the log-demons [54] and Hernandez *et al.* [62] method. All these algorithms have in common that they are based on the structure of the infinite-dimensional group of diffeomorphisms (analogous to finite-dimensional Lie groups), in that compositions of diffeomorphisms are guaranteed to remain in the group, i.e. are also diffeomorphic [52]. In theory, the diffeomorphic formulation assumes continuous space and time; however, in practice, the implementation is based on a relatively small number of discrete time integration steps, and is spatially sampled at discrete voxels. The LDDMM method differs from the others as it can cover a larger range of diffeomorphisms due to its parametrisation being variable in time. However, this is achieved with significantly higher computation time.

The Free-Form Deformation (FFD) algorithm [29] is a well-known and established method which has been found to perform well for inter-subject registration [137], but which is not guaranteed to yield invertible transformations. One appealing aspect of the conventional FFD algorithm is its parametric nature, in which a relatively low dimensional set of degrees of freedom needs to be optimised (and stored, and potentially analysed), in the form of a lattice of control points for a cubic B-spline model. This more parsimonious parametrisation might have advantages for statistical analysis or shape modelling [138].

Rohlfing *et al.* [69] and Sdika [68] modified the FFD to ensure invertibility by using the Jacobian determinant respectively as a penalty term or as a constraint. However, these methods do not have the flexibility of diffeomorphic approaches to generate large deformations by integrating flows over time [59]. Rohlfing *et al.* also rely on finite differencing for the gradient of the objective function, which may reduce accuracy and makes the computation expensive since it requires extra resampling of sub-volumes of the image and multiple objective function evaluations.

Rueckert *et al.* [70] used compositions of spatially constrained transformations to yield large deformation diffeomorphisms. However, they found the method slightly worsened performance, perhaps because the individual transformations were very strongly constrained, meaning that a large number were required to be composed, potentially increasing errors. Moreover, this technique is computationally expensive since all the intermediate transformations have to be stored and composed.

Importantly, these three techniques for obtaining invertible free form deformations do not benefit from some elegant theoretical and practical aspects of the diffeomorphism group setting. For example, while they ensure invertibility in theory, in practice they do not easily yield the inverse transformations. Rueckert *et al.*'s method would in theory enable the computation of the inverse transformation if each of the many composed small deformations were kept so that they could be inverted and composed in reverse order. In contrast, methods that integrate flows can simply integrate them in reverse to obtain a consistent inverse [3]. Similarly, Riemannian interpolation, extrapolation and averaging procedures that account for the non-Euclidean structure of the group of diffeomorphisms should be superior to their simpler equivalents, as argued by Joshi *et al.* [139] regarding the estimation of group-wise average templates.

For the special case of diffeomorphisms in one-parameter subgroups, one can exponentiate stationary velocity fields using an efficient scaling-and-squaring approach [43]. Furthermore, log-Euclidean analysis of diffeomorphisms via their velocity fields greatly simplifies computation of averages or variability [43]. Directly modelling discretised versions of these continuous velocity fields leads to particularly efficient implementations [3, 47].

De Craene *et al.* [72, 73] proposed to use a non-stationary velocity field modelled with a spatio-temporal B-spline to analyse serial cardiac ultra-sound images. Their technique, large diffeomorphic free form deformation, takes advantage of the limited deformation occuring between each frame to reconstruct the whole cardiac motion. The cited diffeomorphic algorithms [59, 3, 47, 62] use a smoothing approach or a regularizer to ensure a smooth transformation and thus a one-to-one mapping. The smoothing or regularisation parameter value is thus of great importance to ensure unfolded deformation. Indeed, a too low smoothing could lead to folded transformation whereas a too high smoothing could lead to over-regularisation. We propose to take advantage of the spline's closed-form expression for the Jacobian matrix to penalize our transformation model. Using the squared log of the Jacobian determinant and its analytical derivative, we ensure that the transformation is unfolded for any penalty term weight.

We propose that the merits of Arsigny's log-Euclidean framework and the FFD's parametric nature can be combined; in this work we model a stationary velocity field in a B-spline basis, and efficiently approximate the scaling-and-squaring approach in the lower-dimensional space of the FFD control points before refining the transformation in the high-dimensional space of the reference image. As emphasised already, this automatically leads to a consistent inverse transformation, and enables a simple procedure for building group-wise (log-Euclidean) average templates. The appeal of the FFD is retained, along with its computational efficiency and ease of parallelisation [30].

Finally, by optimising the velocity field to maximise the Normalised Mutual Information (NMI), we obtain one of relatively few diffeomorphic implementations suitable for multi-modal fusion applications. This is of interest for example for EPI- (echo planar imaging) distortion correction where important deformation are corrected using MRI from different pulse sequences. Related works are Rueckert *et al.* [70] that we previously described and Modat *et al.* [19] which used the NMI to drive the diffeomorphic framework.

In the next section, we present the details of the proposed method: the fast free-form diffeomorphic deformation (F$_3$D^2). We will be focusing on the exponentiation of the lattice of control points and the velocity field optimisation. The third section describes how we determine our technique parameters using one database and how we validated the technique using brain MR image labels using another database. The experimentation is similar to the Klein *et al.* study [137] which enables direct comparison to other existing methods. The final section will discuss the method as well as its advantages and drawbacks compared to other algorithms.

7.1 Method

7.1.1 The Free-Form Deformation algorithm, brief overview

The FFD algorithm aims to transform a floating image F into a warped image $F(\mathbf{T})$ to maximise its similarity with a reference image R. The transformation $\mathbf{T}$ is computed through a cubic B-spline interpolation from a lattice of control points $\{\mu^{\xi}_{i,j,k}\}$ overlaid on the reference image. Indices i, j, k correspond to the points along the x-, y- and z-axis and ξ denotes the x-, y- or z-component. The spacing between each control point along each axis is denoted as δ_x, δ_y and δ_z. The cubic B-spline function is represented by β^3. The deformation field is thus computed as: $\mathbf{T}^{\xi}(\vec{x}) = \vec{x}^{\xi}+$

$$\sum_{i,j,k} \beta^3\left(\frac{x}{\delta_x} - i\right) \beta^3\left(\frac{y}{\delta_y} - j\right) \beta^3\left(\frac{z}{\delta_z} - k\right) \mu^{\xi}_{ijk}. \tag{7.1}$$

The control points $\{\mu^{\xi}_{i,j,k}\}$ are adjusted to maximise an objective function that is a balance between a measure of image similarity and a penalty term enforcing a smooth transformation. The similarity between the warped floating image and the reference image is assessed with the normalised mutual information (NMI) [37], an entropy-based measure that aims to quantify the amount of information that one image has about the other, and which is therefore suitable for multi-modal applications. The penalty term is usually the bending-energy[29], which enforces smoothness but does not necessarily yield one-to-one mappings. The FFD algorithm is not able to handle large deformation such as the classical circle to c shape registration case. Figure 7.1 shows two registration examples where the classical FFD algorithm failed.

In the next subsections, we will discuss the methodological modifications we have performed in order to guarantee a diffeomorphic transformation.

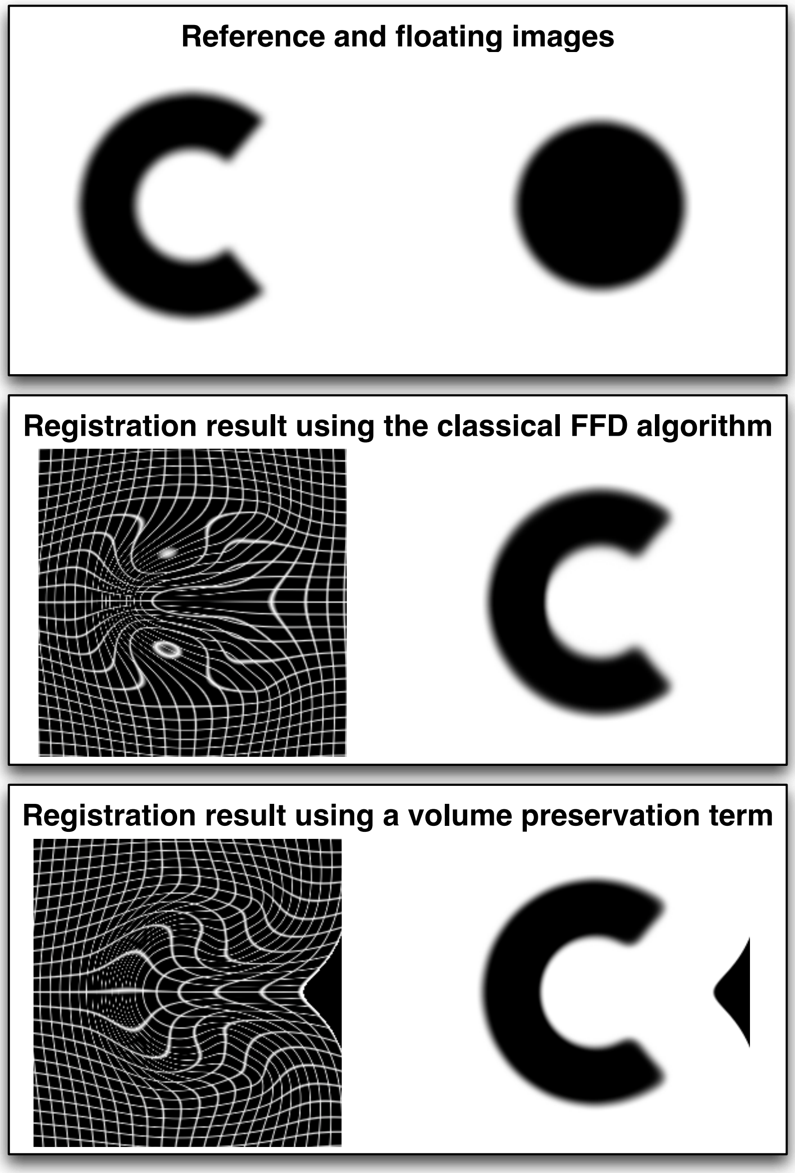

Reference and floating images
Registration result using the classical FFD algorithm
Registration result using a volume preservation term

Figure 7.1: Circle to c shape registration case using the FFD algorithm. Top row shows the reference and floating images. The middle row shows a registration result where folding has been generated. The bottom row shows a registration result using a volume preserving penalty term; it can be notice that the deformation is propagated to the border of the image.

7.1.2 Scaling-and-squaring on a control point lattice

While in the FFD method the transformation control point positions are directly optimised, in $\mathrm{F_3D^2}$ control points are used to parametrize a stationary velocity field from which the final transformation is computed through exponentiation.

Considering a smooth vector field $\mathbf{V}$, the exponential of this field is ensured to be a diffeomorphism [52]. The exponentiation of $\mathbf{V}$ can be computed using an Euler integration approach that consists of successive compositions of deformation field with an infinitesimally small time step. This computation can be efficiently approximated using a scaling-and-squaring approach as shown by Arsigny *et al.* [43] and employed in Dartel [3], the diffeomorphic demons [47] or Hernandez *et al.* approach [62].

Briefly, the scaling-and-squaring approach involves after an initial scaling of a velocity field, composing the resultant field with itself several times to approximate its integration over a large number of steps. Each composition step is performed using the previously obtained field. This procedure is illustrated by figure 7.2 where 6 squaring steps have been used from an initial velocity field to generate a circle to c deformation field.

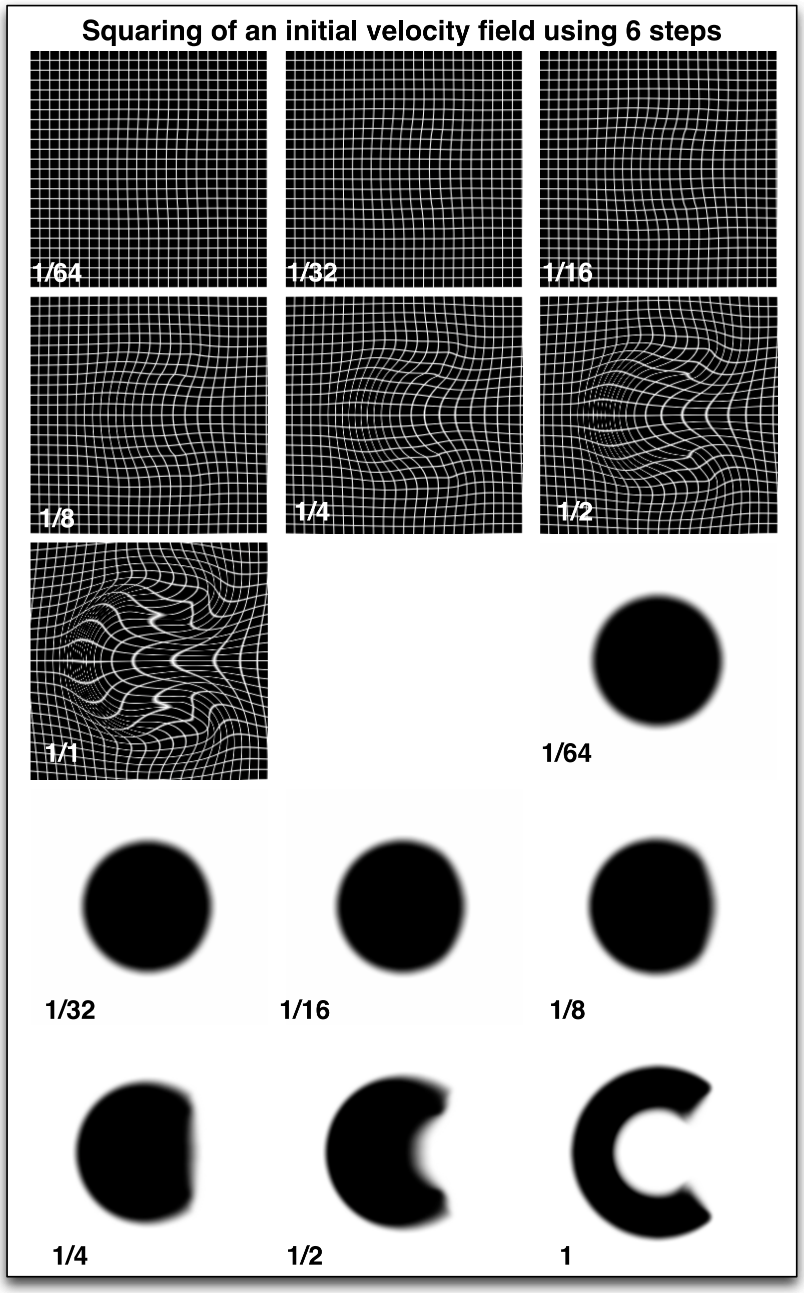


Figure 7.2: Illustration of the scaling-and-squaring approach. An Euler integration of 2^n steps of an initial velocity field can be efficiently approximated using n steps. $n = 6$ in this example.

The control points $\{\vec{\mu}\}$ are used to parametrize, using a cubic B-spline scheme, an initial deformation field $\mathbf{u}$ defined in the space of the reference image. This initial field can be then squared n times to obtain the final deformation field $\mathbf{U}$. In our approach, we approximate the final deformation field using the low-resolution space defined by the control point lattice. It enable us to more efficiently compute the final result as the exponentiation is performed in a sparser space than the voxel space. The stationary velocity field, defined by $\{\vec{\mu}\}$, is applied only at the control point initial position leading to an initial deformation field $\mathbf{u}_l$. The final deformation field in the control point space $\mathbf{U}_l$, is computed using squaring of the initial field through a cubic spline interpolation method. Using the final control point based deformation field, we can then compute an approximated field $\tilde{\mathbf{U}}$ defined in the reference image space. Table 7.1 presents inverse-consistency errors for the high- and low-resolution squaring method. For each configuration, image size of 64^3, 128^3 or 256^3 and control point spacing of 2.5, 5 or 10, we generated 100 random initial velocity field. We computed all the deformation fields $\mathbf{U}$ and $\tilde{\mathbf{U}}$ as well as their inverse respectively, as describe in subsection 7.1.3. We then composed, denoted by $\circ$, both transformations: forward and backward, and computed the mean euclidean distance to an identity transformation. Note that $\mathbf{U} \circ \mathbf{U^{-1}}$ and $\mathbf{U}^{-1} \circ \mathbf{U}$ gave similar inverse consistency errors. Consistency errors have been averaged for the different image size as the error did not change with different image dimension. All deformation fields have been generated using 8 squaring steps. The tests have been performed using a 3GHz Xeon processor and a single threaded implementation. Table 7.1 also reports the speed-up ratio between the computation of $\mathbf{U}$ and $\tilde{\mathbf{U}}$.

Table 7.1: Comparison of the dense- and coarse-resolution squaring. Tests have been performed using different image size (64^3, 128^3 and 256^3) and different control point spacing (2.5, 5 and 10 voxels width).

	$\delta = \mathbf{2.5}$	$\delta = \mathbf{5}$	$\delta = \mathbf{10}$
Error($\mathbf{U} \circ \mathbf{U}^{-1}$)	0.0229	0.0149	0.0080
Error($\tilde{\mathbf{U}} \circ \tilde{\mathbf{U}}^{-1}$)	0.0440	0.0856	0.1694
Speed-up ratio (64^3 image)	2.1	3.7	6.6
Speed-up ratio (128^3 image)	2.2	4.0	6.9
Speed-up ratio (256^3 image)	2.5	5.0	8.2

It can be noticed from table 7.1 that the inverse consistency error difference between the proposed approximation and the dense scaling-and-squaring approach is not negligible. In order to merge efficiency and accuracy, we then used a 2-steps approach where the registration degrees of freedom $\{\mu\}$ were first optimised using the scaling-and-squaring approach in the control point space. After convergence, the scaling-and-squaring approach was secondly performed in the dense reference image space until second convergence. It can also be notice from the table that the speed-up ratio is not proportional to the voxel versus control point number ratio. This is due to the interpolation technique used in both approaches. Whereas we used linear interpolation to perform squaring of the dense deformation field, we used cubic spline interpolation for the squaring in the control point space.

7.1.3 The inverse transformation

A diffeomorphic transformation $\mathbf{T}$ implies invertibility in a sense that $\mathbf{T}^{-1}$ is defined. However, not all diffeomorphic algorithms explicitly provide the $\mathbf{T}^{-1}$ transformation.

Since our transformation is based on an exponential map, we know that $\exp(-\mathbf{V}) = (\exp \mathbf{V})^{-1}$ [52]. It it thus possible, using the scaling-and-squaring approach and $\{-\vec{v}_{i,j,k}\}$ as a stationary velocity field, to estimate $\mathbf{T}^{-1}$. The same technique could be applied in both the voxel and the control point space. In order to assess, the influence of the number of squaring steps on the inverse consistency, we generated 100 stationary velocity fields that we integrated using different number of steps. We report in table 7.2 the inverse consistency errors computed similarly to those in table 7.1. Tests have been performed using a 256^3 image size and a 5 voxels-width control point spacing. It can be notice that the inverse consistency error decreases as the number of squaring step increases and then increases after a plateau from 6 to 9 steps. While one could expect the inverse consistency error to always decrease with a step number increase; this is not the case due to precision error. Note that the result reported in table 7.2 have been generated using double precision but similar numbers are obtained using single precision.

Table 7.2: Inverse consistency error using different squaring step number.

Step number	Error($\mathbf{U} \circ \mathbf{U}^{-1}$)	Error($\mathbf{U}^{-1} \circ \mathbf{U}$)
2	0.0390	0.0389
3	0.0243	0.0244
4	0.0184	0.0184
5	0.0161	0.0161
6	0.0153	0.0153
7	0.0150	0.0150
8	0.0149	0.0149
9	0.0154	0.0154
10	0.0170	0.0170
11	0.0221	0.0220
12	0.0338	0.0338

7.1.4 Optimisation and deformation field regularisation

Within the FFD framework, the objective function to maximise consists of a similarity measure (NMI) and one or several penalty terms P which enforces a smooth deformation. The penalty terms we used are the bending-energy, $\mathcal{P}_{\text{BE}}$, the divergence of the velocity field, $\mathcal{P}_{\text{div}}$, and similarly to Rohlfing *et al.* [69] a term $\mathcal{P}_{\text{Jac}}$ based on the Jacobian determinant:

$$\mathcal{P}_{\text{Jac}} = \log^2(\det(\text{Jac}(\mathbf{T}))). \tag{7.2}$$

In order to generate a diffeomorphism, one must ensure that the scaled velocity field is a diffeomorphic transformation. Non-parametric approaches ensure diffeomorphism by using an appropriate amount of smoothing or regularization to insure topological correctness. Our approach enables us to analytically compute Jacobian determinant and its gradient from the B-spline parametrisation in the initial dense deformation field. By using $\mathcal{P}_{\text{Jac}}$ as an extra penalty term we are ensuring diffeomorphic transformation.

The objective function $\mathcal{O}(R, F; \vec{\mathbf{v}})$ to maximise is thus:

$$\begin{aligned}\mathcal{O}(R, F; \vec{\mathbf{v}}) &= \alpha \times \text{NMI} \\ &- \beta \times \mathcal{P}_{\text{BE}} \\ &- \gamma \times \log^2(\det(\text{Jac}(\mathbf{T}))) \\ &- \epsilon \times \mathcal{P}_{\text{div}},\end{aligned} \tag{7.3}$$

The trade-off values α, β, γ and ϵ are chosen to sum to 1.

In order to maximise $\mathcal{O}(R, F; \vec{\mathbf{v}})$, the derivatives of the NMI and the derivative of the penalty terms have to be computed:

$$\begin{aligned}\frac{\partial \mathcal{O}(R, F; \vec{\mathbf{v}})}{\partial v^{\xi}_{i,j,k}} &= \alpha \times \frac{\partial \text{NMI}}{\partial v^{\xi}_{i,j,k}} \\ &- \beta \times \frac{\partial \mathcal{P}_{\text{BE}}}{\partial v^{\xi}_{i,j,k}} \\ &- \gamma \times \frac{\partial \mathcal{P}_{\text{Jac}}}{\partial v^{\xi}_{i,j,k}} \\ &- \epsilon \times \frac{\partial \mathcal{P}_{\text{div}}}{\partial v^{\xi}_{i,j,k}}.\end{aligned} \tag{7.4}$$

Using the classical FFD framework, that assumes a small deformation model, the NMI gradient is computed using R and $F(\mathbf{T})$. Typically, the derivatives of a paired intensity distribution are computed, from which the derivatives of the marginal and joint entropies are assessed before obtaining the gradients of the NMI for each control point. To use a stationary velocity field parametrisation, one must exponentiate the gradient. We first compute the similarity measure gradients between R and $F(\mathbf{T})$ and then use the scaling-and-squaring approach to exponentiate the gradient. The pipeline is as follow:

- Compute the voxel-wise NMI gradient $\mathbf{G}$ between R and $F(\mathbf{T})$
- Compute the initial deformation field $\mathbf{u}_{-1/2^n}$ by negating and scaling the velocity field.
- Compute for each voxel the Jacobian matrix $\mathbf{J}_{-1/2^n}$ from $\mathbf{u}_{-1/2^n}$
- Update the gradient:
  $\mathbf{G}(\vec{x}) = \mathbf{G}(\vec{x}) + \mathbf{u}_{-1/2^n}(\mathbf{G}(\vec{x})) \times \mathbf{J}_{-1/2^n}(\vec{x})^T$
- Perform similarly for each subsequent squaring step.
- Extract the node-wise NMI gradient from the obtained voxel-wise gradient, as in [30].

Using a stationary velocity field parametrisation, we are able to compute the inverse which enable us to use a symmetric formulation of the similarity measure.

$$\text{NMI} = \frac{H(R) + H(F(\mathbf{T}))}{H(R, F(\mathbf{T}))} + \frac{H(F) + H(R(\mathbf{T}^{-1}))}{H(F, R(\mathbf{T}^{-1}))} \tag{7.5}$$

where $H(R)$, $H(F)$, $H(F(\mathbf{T}))$ and $H(R(\mathbf{T}^{-1}))$ are the marginal entropy and $H(R, F(\mathbf{T}))$ and $H(F, T(\mathbf{T}^{-1}))$ are the joint entropies of pair of images. These entropy values are computed from a joint histogram, which in our implementation is based on a Parzen window approach where a cubic B-spline is used to define the window.

The penalty terms are computed directly on the velocity field and the same approach would be use in the classical FFD. We refer the reader to [30] for an efficient analytical computation of the bending-energy term. Rohlfing *et al.* [69] used an optimisation based on a symmetric difference evaluation of the objective function. Novelly here, for efficiency and accuracy, we use the analytical derivative of the objective function. The Jacobian based penalty term can be differentiated with regards to each degree of freedom $v^{\xi}_{i,j,k}$, as:

$$\begin{aligned} \frac{\partial \mathcal{P}_{\text{Jac}}}{\partial v^{\xi}_{i,j,k}} &= 2 \times \log\left(\det(\text{Jac}(\mathbf{T}))\right) \\ &\times \ \text{Tr}\left(\text{Jac}^{-1}(\mathbf{T}) \frac{\partial \text{Jac}(\mathbf{T})}{\partial v^{\xi}_{i,j,k}}\right), \end{aligned} \tag{7.6}$$

where $\text{Tr}(.)$ is the trace operator. For efficiency reasons, one can compute the gradient only at the control point position. Note that this approximation only ensure strictly positive Jacobian determinant at the control point positions and not in the continuous space of the transformation.

The set of all derivatives $\{\partial \mathcal{O}_{R,F(\mathbf{T})} / \partial v^{\xi}_{i,j,k}\}$ are then used to perform a conjugate gradient ascent, updating all degrees of freedom concurrently.

While the penalty terms promote a smooth transformation, they do not ensure unfolded deformation during the line-ascent optimisation due to the different trade-off values used for the metric and the constraints. No folded solution can be accepted as better than an unfolded one since the penalty term would return a NaN value. However, since the optimisation is in our case performed concurrently for every degree of freedom, a folded voxel could stop the line-ascent process. Karaçali and Davatzikos [108] proposed to locally unfold areas with negative Jacobian determinants. However, their method is based on a non-parametric deformation model and is thus not directly suitable for a cubic B-spline deformation model. Andersson *et al.* use this approach to unfold deformation fields[1] in their FNIRT software (`http://www.fmrib.ox.ac.uk/fsl/fnirt/index.html`). Instead, we propose to take advantage again of the analytical definition of the Jacobian matrices. We locally use the gradient of the Jacobian determinant of the folded voxel in order to increase their values. This is done by modifying the velocity field values along this Jacobian determinant gradient line. This approach has the advantage to be performed directly on the transformation parameters $\{\vec{\mu}\}$ and does not require approximation by going to a dense deformation field and back to the B-spline parameters. We applied this *unfolding step* each time a negative Jacobian determinant is produced while performing the gradient ascent.

7.1.5 GPU-based implementation

Based on previous work [30], $\mathrm{F_3D^2}$ has been formulated using a CPU- and a GPU-based implementation. Both versions of the code are freely available as part of the NiftyReg package[1]. The GPU-based implementation uses CUDA, an API developed by NVidia [4][2]. All functions have been implemented using CUDA except the filling of the joint-histogram through a Parzen window approach. Indeed, different results are obtained using single or double precision and we found that using CUDA and double precision[3] we were not getting any speed improvement. Note that some joint-histogram filling technique on the GPU have been proposed in the literature [78, 79] but to the best of our knowledge they all use filling through linear interpolation [34] which does not allow analytical derivation. We refer the reader to [30, 21] for a CPU- versus GPU-based implementation speed-up evaluation within the NiftyReg package. A drawback of the two-kernel approach is that the intermediate Jacobian matrices have to be stored; up to 448MBs for a 200^3 image for example. Moreover, due to the composition scheme, the initial position of the voxel cannot be estimated prior to computation meaning that each thread should iterate over every voxel in the image to assess if they are in the local support area of the current control point. This implementation does not provide any significant speed-up and is not applicable for large image size. Indeed, the computation within a thread is too time consuming and is not suitable for a CUDA-based implementation. This is due to the fact that the CUDA API has a feature that stops a thread when its running time exceeds a few seconds (2 seconds for example when using Windows as an operating system). For these reasons, we did not report any computation time using the GPU-based implementation and the Jacobian-based penalty term gradient based on the velocity field. The difficulty to efficiently implement this function for a graphical card emphasizes the need for approximation methods.

7.2 Experiments and results

The proposed method has been assessed against the fast free-form deformation [30] ($\mathrm{F_3D}$, part of the NiftyReg package) that directly optimises the control point final positions. We performed evaluation by following the strategy used in Klein *et al.* [137] for evaluation of non-rigid registration algorithms to allow direct comparison.

The MGH10[4] database consists of 10 MRI acquired at the MGH/MIT/HMS Athinoula A. Martinos Center for Biomedical Imaging using a 3T Siemens scanner. Subject ages range from 22 to 29. Each brain has been manually parcellated into 73 regions of interest. Each of the 10 brain images have been register to the 9 remaining images. The registration were performed using the $\mathrm{F_3D}$ algorithm, a fast implementation of the standard FFD, the proposed method first without any approximation and then using the approximation described in section 7.1.2. Registrations were performed after inhomegeneity correction and only voxel within the brain mask were considered for the optimisation process. It resulted in a total of 90×3 registrations which transformations were used to propagate regions of interest between subject. We then used a segmentation overlap measure as a measure of registration accuracy. For every

[1] http://sourceforge.net/projects/niftyreg

[2] http://www.nvidia.com

[3] CUDA-enabled cards have a limited amount of processor that can handle double precision computation.

[4] Data information can be found here: `http://speechlab.bu.edu/imaging.php`

registration, the control point spacing was set to 2.5 voxels along each axis. This value was chosen to reproduce the parameters used with IRTK[5] [29]. The bending-energy weight was set to 1% for $\mathrm{F_3D}$ and 10% for $\mathrm{F_3D^2}$. The Jacobian based constraint and the divergence were both set to 20% for the $\mathrm{F_3D^2}$ implementation. These trade-off values were defined empirically using visual assessment on others images. The stationary velocity field were integrated using 6 steps. Similarly to [137] we used the target overlap (TO) (eq. 7.7) to assess overlap between the ground truth (GT) and the propagated (PS) segmentations.

$$\mathrm{TO} = \frac{\mathrm{GT} \cap \mathrm{PS}}{\mathrm{GT}} \tag{7.7}$$

Results are presented in figure 7.3 and are directly comparable to those obtained by Klein *et al.* [137]. For comparison, we also report the TO values using only affine registration. The three approaches lead to similar results and are also similar to the best segmentation propagation reported in [137]. It is worth noting that no folding were generated using the proposed $\mathrm{F_3D^2}$ approaches whereas some negative Jacobian determinant values were produced using $\mathrm{F_3D}$.

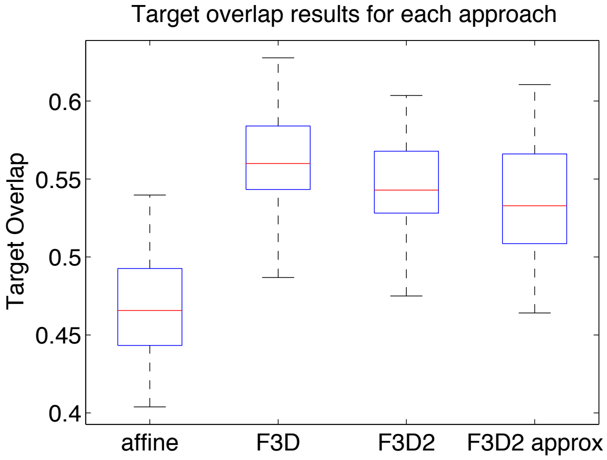


Figure 7.3: Segmentation propagation results. Target overlap are presented after affine registration and three different non-rigid registration approaches.

All CPU-based registrations have been performed on a computer cluster with processors ranging from 1.8 to 3GHz. Using single-threaded implementation[6], the mean computation time were 4.7(0.8), 64.7(21.1) and 15.7(4.9) minutes for $\mathrm{F_3D}$, $\mathrm{F_3D^2}$ and $\mathrm{F_3D^2}$ with approximation.

7.3 Discussion

We presented an implementation of the log-Euclidean method in the FFD framework. The method takes advantage of the scaling-and-squaring approach to efficiently obtain the deformation field using a fewer degrees of freedom. The stationary velocity field parametrisation enabled the optimisation of the NMI using a symmetric approach where forward and backward transformation are considered.

[5] http://www.doc.ic.ac.uk/∼dr/software/

[6] Note that the released implementation handle multi-core CPU architecture.

The proposed framework is guaranteed to avoid folding by using an analytical implementation of a Jacobian-based penalty term. Non-parametric formulations for non-rigid registration [59, 3, 47, 62] ensure topologically consistent transformations using a smoothing parameter or regularisation of the velocity field. However, if the amount of regularisation is too low or the time-step too high, folding can still occur. Using $\mathrm{F_3D^2}$ the continuous property of the spline and the Jacobian-based penalty term ensure unfolded deformation for any control point spacing value and any penalty term trade-off values.

An efficient implementation of a directly invertible deformation would be helpful for several applications: groupwise registration of a large dataset for phenotyping using segmentation propagation [24]; joint analysis of volumes and surfaces, since the latter are typically transformed by forward-mapping their vertex coordinates, instead of the backward-mapping more common for volumetric warping; or the inter-subject transformation of within-subject deformation- or motion fields [140].

De Craene *et al.* [72, 73] use Euler integration steps to follow the path of each voxel through a non-stationary velocity field over time. In their cardiac application, they have a large number of time-points (around 18 images per cardiac cycle), but do not specify the number or spacing of the control points in time (in space they have an initial grid of $3 \times 5 \times 5$ which is refined twice). With only two time-points, and a small number of control points in time, the velocity field would become approximately stationary in time, as ours is. However, two important differences remain. Firstly, by following just the points starting at the original control point locations, we have a far less computationally expensive algorithm, but potentially at the cost of modelling large deformations less accurately. The second difference is their use of evenly spaced Euler time-steps, instead of the scaling-and-squaring approach. The scaling-and-squaring approach for matrix exponentiation has been thoroughly studied and found to be superior to more naive implementations [141].

However, the exponentiation (integration) of velocity fields has been less thoroughly studied; for example, Arsigny [43] did not compare the method to any simpler approaches, so it remains possible that uniform Euler integration could be preferable. We intend to investigate both of these aspects in future work.

Chapter 8

Conclusion

8.1 Summary

This thesis has presented work to address some of the challenges of medical image registration. Based on the work of Rueckert *et al.* [29], the popular FFD framework has been expanded with the aim of improving its efficiency, accuracy and robustness.

Efficiency was addressed with a careful implementation of the technique using a rigorous mathematical framework. A fast optimisation was enabled using (i), an analytical derivative of the objective function and (ii) a voxel-based to node-based approach for a cubic B-spline model. Due to the parallel-friendly algorithm, a multi-threaded implementation has been developed using GPUs. The algorithm has been presented in chapter 4. This implementation was used for time critical neurosurgical application and it has been shown to be robust and fast enough to meet the requirements. A direct comparison a previous implementation showed a speed-up up to two orders of magnitude in some cases. The multi-threaded GPU implementation lead to a speed up of 10 to 20 versus single-thread CPU implementation of the same algorithm. The proposed efficient implementation has also been shown to be accurate. Using segmentation-propagation technique on a set of 320 labels we showed that the proposed algorithm was efficient but also at least as accurate previous implementation of the FFD. In some cases, it outperformed the classical FFD implemented as in Rueckert *et al.* [29]. This was due to a more accurate objective function derivative as well as the advantage of running the algorithm until convergence. The proposed implementation has been used in a Grand Challenge context where it performed similarly to the best algorithm (ranked second out of 20 at the time of the workshop) but was computationally more efficient [142].

Chapter 5 gathered further work on the proposed framework to improve robustness. This has been achieved by modifying the objective function. I added some features that ensure one-to-one mapping and thus avoid folding and ensure bijective transformation. A folding correction scheme for a cubic B-splines deformation model has also been presented in this chapter. The topology conservation was performed using efficient derivative of Jacobian based measures. Another way to increase the algorithm robustness was to use a iterative differential bias field correction scheme. It lead to a registration algorithm less sensitive to intensity inhomogeneity. This contribution enhanced the algorithm robustness to artefact and its usefulness has been shown on synthetic data. We used a second lattice of control point to parametrize

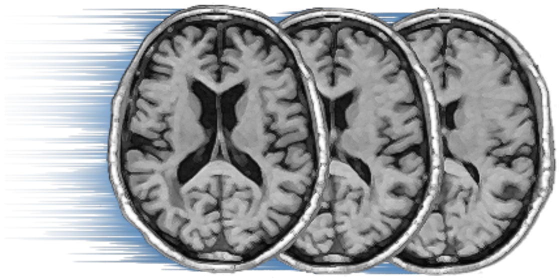


Figure 8.1: NiftyReg logo

through a B-splines scheme the intensities modulation. Optimisation of the intensity modulation was achieved by maximising the NMI.

Within this thesis, I also presented, in chapter 6 some preliminary work where a non-linear elastic model as been incorporated in the framework to ensure bio-mechanically plausible deformation. The NMI derivative at each node was scaled and used as ”pseudo forces” to deform the model. The framework has only been used on neuro-degenerative patient images and as such did not fully reach its full potential: modelling mechanical deformation.

Taken advantage of a diffeomorphic framework, I also presented in chapter 7 a modification of the FFD that deal with topology conservation. Using a stationary velocity field also enabled the direct computation of the forward and backward transformation. Validation of the method has been performed using segmentation propagation based on publicly available database. It enables direct comparison with other state of the art algorithm.

Most of the methods presented in this thesis have been implemented in the NiftyReg package (figure 8.1). This package has been released online under a BSD licence [1]. The package can be freely downloaded from `http://sourceforge.net/projects/niftyreg`. NiftyReg contains algorithms to perform non-rigid registration but also algorithm for rigid and affine warping. The global registration algorithm is based on Ourselin *et al.* work [31]. This package has already been used in several application within and outside the Centre for Medical Image Computing. Its has been used for example for segmentation propagation [143, 144, 145], for lung registration [146], for colon registration using a 2 dimensional tubular approach [147], neurosurgery [21] or for breast registration [148].

8.2 Future work

An interesting extension of the work presented in the thesis would be longitudinal registration. Some works that consider longitudinal data and perform symmetric registration or use symmetric measure have already been proposed in the literature [57, 2, 62]. However, very little has been done using more than two time points. De Creane *et al.*[73] uses longitudinal data from heart US to reconstruct a full cycle. Heart applications are however different from neuro-degenerative process. Indeed, a cyclic motion can be assumed and used to model the deformation. Durrleman *et al.*[149] developed a framework to co-register two or more sets of longitudinal data. Future work will investigate intra-patient longitudinal data

[1] `http://en.wikipedia.org/wiki/BSD_licenses`

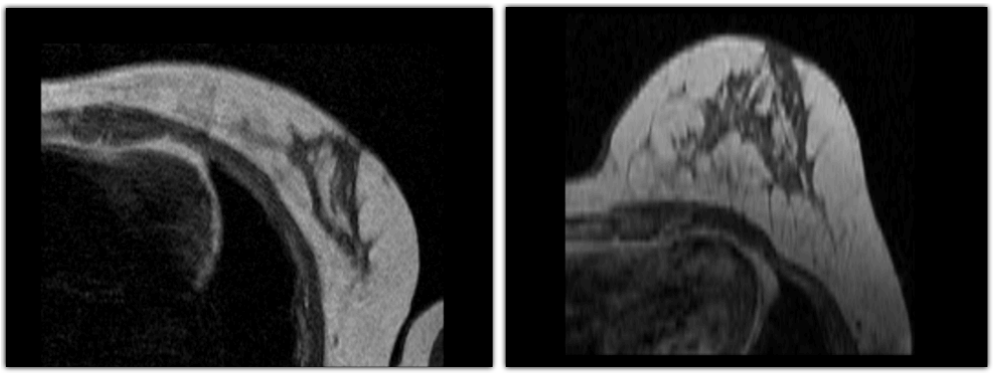


Figure 8.2: MRI prone and supine acquisitions. Figure from Carter *et al.* [5]

registration in neuro-degenerative disease. As the degeneration process can not be assumed linear, using the time of scan to model the deformation could lead to bias. Moreover no information is known before the first time point or after the last time point. Could we use registration results including deformation as well as velocity as a biomarker when derive from two time points or more? Using longitudinal data with multi-scale information is also an area of future work. Indeed nowadays longitudinal data are available but also images from different modalities or pulse sequences, diagnostic by experts, genetic markers identification or protein blood or spinal fluid concentration for example. Using longitudinal data and multi-scale information, can we build model of disease progression? Could registration be used to identify the disease progression stage of a patient using such models?

As already mentioned, the work proposed in chapter 6 using non-linear elastic model was preliminary. Future work could consist in applying the developed algorithm to mechanical deformations. Two examples of application are (i) breast imaging and (ii) neurosurgery. While MRI of the breast is performed, two images are acquired: patients lie either in a prone or a supine position as shown in figure 8.2. Carter *et al.* [5] proposed to use finite element model to which they apply gravity. Han *et al.* [148] uses the same approach but also modified the tissues properties to improve the final result. Their method was used to initialise a non-rigid registration algorithm. The next step would be to link the modelling and the registration in a combined framework. Another application is brain surgery. As shown in figure 1.5 or 4.6 large deformation occurs during surgery. These displacements are mostly due to mechanical constraint such as resections or boundary condition modifications.

The NiftyReg package has been targeting efficiency while maintaining accuracy and robustness. The package could still evolve has I am planning to do some modifications. At the moment it contains single threaded CPU-based or multi-threaded GPU-based implementations. Future work will consists of implementing multi-threaded CPU implementation. Some NiftyReg modules using convolution will be re-factored to take advantage of efficient computation in the frequency domain. The current implemented registration algorithms, global and local, do not provide inverse consistency. Future work will include inverse-consistency. This could be achieved by optimising concurrently forward and backward transformation.

Bibliography

[1] J. Andersson, M. Jenkinson, S. Smith, and J. Andersson, "Non-linear registration, aka spatial normalisation FMRIB technical report TR07JA2," Oxford, UK: FMRIB Analysis Group of the University of Oxford, Tech. Rep., 2007. [Online]. Available: https://www.fmrib.ox.ac.uk/analysis/techrep/tr07ja2/tr07ja2.pdf

[2] B. B. Avants, C. L. Epstein, M. Grossman, and J. C. Gee, "Symmetric diffeomorphic image registration with cross-correlation: evaluating automated labeling of elderly and neurodegenerative brain," *Medical Image Analysis*, vol. 12, no. 1, pp. 26–41, Feb 2008.

[3] J. Ashburner, "A fast diffeomorphic image registration algorithm," *NeuroImage*, vol. 38, no. 1, pp. 95–113, Oct 2007.

[4] NVIDIA, *NVIDIA CUDA Programming Guide Version 3.1*, 2010.

[5] T. Carter, C. Tanner, N. Beechey-Newman, D. Barratt, and D. Hawkes, "Mr navigated breast surgery: method and initial clinical experience," in *Proc. MICCAI'08*, no. Pt 2, Jan 2008, pp. 356–63.

[6] R. L. Wahl, L. E. Quint, R. D. Cieslak, A. M. Aisen, R. A. Koeppe, and C. R. Meyer, ""'anatometabolic" tumor imaging: fusion of FDG PET with CT or MRI to localize foci of increased activity," *J Nucl Med*, vol. 34, no. 7, pp. 1190–7, Jul 1993.

[7] P. A. Freeborough and N. C. Fox, "The boundary shift integral: an accurate and robust measure of cerebral volume changes from registered repeat MRI," *IEEE Transactions on Medical Imaging*, vol. 16, no. 5, pp. 623–9, Oct 1997.

[8] K. K. Leung, J. Barnes, G. R. Ridgway, J. W. Bartlett, M. J. Clarkson, K. Macdonald, N. Schuff, N. C. Fox, S. Ourselin, and A. D. N. Initiative, "Automated cross-sectional and longitudinal hippocampal volume measurement in mild cognitive impairment and Alzheimer's disease," *NeuroImage*, vol. 51, no. 4, pp. 1345–59, Jul 2010.

[9] L. A. van de Pol, E. S. C. Korf, W. M. van der Flier, H. R. Brashear, N. C. Fox, F. Barkhof, and P. Scheltens, "Magnetic resonance imaging predictors of cognition in mild cognitive impairment," *Arch Neurol*, vol. 64, no. 7, pp. 1023–8, Jul 2007.

[10] B. J. Nieman, A. M. Flenniken, S. L. Adamson, R. M. Henkelman, and J. G. Sled, "Anatomical phenotyping in the brain and skull of a mutant mouse by magnetic resonance imaging and computed tomography," *Physiol Genomics*, vol. 24, no. 2, pp. 154–62, Jan 2006.

[11] K. V. Leemput, F. Maes, D. Vandermeulen, and P. Suetens, "Automated model-based tissue classification of MR images of the brain," *IEEE Transactions on Medical Imaging*, vol. 18, no. 10, pp. 897–908, Oct 1999.

[12] B. Fischer and J. Modersitzki, "Ill-posed medicine—an introduction to image registration," *Inverse Problems*, vol. 24, p. 4008, Jun 2008.

[13] W. R. Crum, T. Hartkens, and D. L. G. Hill, "Non-rigid image registration: theory and practice," *British Journal of Radiology*, vol. 77 Spec No 2, pp. S140–53, Jan 2004.

[14] A. Gholipour, N. Kehtarnavaz, R. W. Briggs, M. Devous, and K. S. Gopinath, "Brain functional localization: a survey of image registration techniques," *IEEE Transactions on Medical Imaging*, vol. 26, no. 4, pp. 427–451, 2007.

[15] M. Holden, "A review of geometric transformations for nonrigid body registration," *IEEE Transactions on Medical Imaging*, vol. 27, no. 1, pp. 111–128, 2008.

[16] M. Modat, Z. A. Taylor, J. Barnes, D. J. Hawkes, N. C. Fox, and S. Ourselin, "Fast free-form deformation using the normalised mutual information gradient and graphics processing units," in *High-Performance MICCAI workshop*, Jan 2008.

[17] M. Modat, G. R. Ridgway, Z. Taylor, D. J. Hawkes, N. C. Fox, and S. Ourselin, "A parallel-friendly normalized mutual information gradient for free-form registration," in *Proceedings of SPIE*, Jan 2009.

[18] M. Modat, G. R. Ridgway, Z. A. Taylor, M. Lehmann, J. Barnes, D. J. Hawkes, N. C. Fox, and S. Ourselin, "Fast free-form deformation using graphics processing units," *Comput Meth Prog Bio*, vol. 98, no. 3, pp. 278–84, Jun 2010.

[19] M. Modat, G. R. Ridgway, D. J. Hawkes, N. C. Fox, and S. Ourselin, "Nonrigid registration with differential bias correction using normalised mutual information," in *Biomedical Imaging: From Nano to Macro, 2010 IEEE International Symposium on*, 2010, pp. 356 – 359.

[20] M. Modat, G. R. Ridgway, P. Daga, M. J. Cardoso, D. J. Hawkes, J. Ashburner, and S. Ourselin, "Log-Euclidean free-form deformation," in *Proc. of SPIE*, 2011.

[21] P. Daga, M. Modat, C. Micallef, L. Mancini, M. White, M. J. Cardoso, N. Kitchen, A. W. McEvoy, J. Thornton, T. Yousry, D. Hawkes, and S. Ourselin, "Near real time brain shift estimation for interventional MRI suite," in *High-Performance MICCAI workshop*, 2010.

[22] J. Cleary, A. Price, M. Modat, D. Thomas, P. Scambler, V. Kyriakopoulu, R. Ordidge, S. Our-Selin, and M. Lythgoe, "Novel phenotyping methods using microMRI," *Genetics Research*, vol. 91, no. 2, pp. 138–138, 2009.

[23] J. O. Cleary, M. Modat, A. Price, N. Greene, D. Thomas, P. Scambler, D. J. Hawkes, R. Ordidge, S. Ourselin, and M. Lythgoe, "Phenotyping in the mouse embryo using a microMRI atlas," in *Proceedings 17th Scientific Meeting, International Society for Magnetic Resonance in Medicine, Honolulu*, Nov 2009, p. 825.

[24] J. O. Cleary*, M. Modat*, F. C. Norris, A. N. Price, S. A. Jayakody, J. P. Martinez-Barbera, N. D. E. Greene, D. J. Hawkes, R. J. Ordidge, P. J. Scambler, S. Ourselin, and M. F. Lythgoe, "Magnetic resonance virtual histology for embryos: 3D atlases for automated high-throughput phenotyping," *NeuroImage*, vol. 54, no. 2, pp. 769–78, 2011, PMID: 20656039.
*equal contribution.

[25] F. C. Norris, J. O. Cleary, M. Modat, A. N. Price, K. McCue, S. Beddow, P. J. Scambler, S. Ourselin, and M. F. Lythgoe, "Propagation-based morphometry in an ex vivo mouse embryo atlas - assessment and validation," in *Proceedings 18th Scientific Meeting, International Society for Magnetic Resonance in Medicine, Stockholm*, 2010.

[26] J. McClelland, S. Hughes, M. Modat, S. Ahmad, D. Landau, S. Ourselin, and D. J. Hawkes, "Study of inter-fraction variations in respiratory motion using deformable registration based motion models," in *Proc. of the XVIth ICCR*, 2010.

[27] M. Modat, J. McClelland, and S. Ourselin, "Lung registration using the NiftyReg package," in *MICCAI workshop: Evaluation of Methods for Pulmonary Image Registration*, 2010.

[28] H. Roth, J. McClelland, M. Modat, D. Boone, M. Hu, S. Ourselin, G. Slabaugh, S. Halligan, and D. J. Hawkes, "Establishing spatial correspondence between the inner colon surface visualized in prone and supine colon CT," in *Proc. MICCAI'10*, 2010.

[29] D. Rueckert, L. I. Sonoda, C. Hayes, D. L. G. Hill, M. O. Leach, and D. J. Hawkes, "Nonrigid registration using free-form deformations: application to breast MR images," *IEEE Transactions on Medical Imaging*, vol. 18, no. 8, pp. 712–21, Aug 1999.

[30] M. Modat, Z. Taylor, G. Ridgway, J. Barnes, E. Wild, D. J. Hawkes, N. Fox, and S. Ourselin, "Nonlinear elastic spline registration: Evaluation with longitudinal Huntington's disease data," in *Workshop on Biomedical Image Registration*, 2010, pp. 128–139.

[31] S. Ourselin, A. Roche, G. Subsol, X. Pennec, and N. Ayache, "Reconstructing a 3D structure from serial histological sections," *Image and Vision Computing*, vol. 19, no. 1-2, pp. 25–31, Jan 2001.

[32] C. Studholme, D. L. G. Hill, and D. J. Hawkes, "Multiresolution voxel similarity measures for MR-PET registration," in *Information processing in Medical Imaging (IPMI)*, Jan 1995.

[33] C. E. Shannon, “The mathematical theory of communication,” *Bell Syst. Tech. J.*, vol. 27, pp. 379–423/623–656, Jan 1948.

[34] F. Maes, A. Collignon, D. Vandermeulen, G. Marchal, and P. Suetens, “Multimodality image registration by maximization of mutual information,” *IEEE Transactions on Medical Imaging*, vol. 16, no. 2, pp. 187–98, Apr 1997.

[35] D. Mattes, D. R. Haynor, H. Vesselle, T. K. Lewellen, and W. Eubank, “PET-CT image registration in the chest using free-form deformations,” *IEEE Transactions on Medical Imaging*, vol. 22, no. 1, pp. 120–8, Jan 2003.

[36] P. Viola and W. Wells, “Alignment by maximization of mutual information,” *Int J Comput Vision*, vol. 24, no. 2, pp. 137–154, 1997.

[37] C. Studholme, D. Hill, and D. J. Hawkes, “An overlap invariant entropy measure of 3D medical image alignment,” *Pattern Recognition*, vol. 32, no. 1, pp. 71–86, Jan 1999.

[38] D. Loeckx, P. Slagmolen, F. Maes, D. Vandermeulen, and P. Suetens, “Nonrigid image registration using conditional mutual information,” *IEEE Transactions on Medical Imaging*, vol. 29, no. 1, pp. 19–29, Jan 2010.

[39] X. Zhuang, D. Hawkes, and S. Ourselin, “Spatial information encoded mutual information for nonrigid registration,” in *Workshop on Biomedical Image Registration*, vol. 6204, Jan 2010, pp. 246–257.

[40] N. Cahill, “Normalized measures of mutual information with general definitions of entropy for multimodal image registration,” in *Workshop on Biomedical Image Registration*, vol. 6204/2010, Jan 2010, pp. 258–268.

[41] G. E. Christensen, R. Rabbitt, and M. I. Miller, “3D brain mapping using a deformable neuroanatomy,” *Phys. Med. Biol*, vol. 39, p. 609418, 1994.

[42] G. E. Christensen, R. D. Rabbitt, and M. I. Miller, “Deformable templates using large deformation kinematics,” *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, vol. 5, no. 10, pp. 1435–47, Jan 1996.

[43] V. Arsigny, O. Commowick, X. Pennec, and N. Ayache, “A log-euclidean framework for statistics on diffeomorphisms,” *Medical image computing and computer-assisted intervention : MICCAI International Conference on Medical Image Computing and Computer-Assisted Intervention*, vol. 9, no. Pt 1, pp. 924–31, Jan 2006.

[44] J. P. Thirion, “Image matching as a diffusion process: an analogy with Maxwell’s demons,” *Medical Image Analysis*, vol. 2, no. 3, pp. 243–60, Sep 1998.

[45] S. Klein, M. Staring, and J. P. W. Pluim, “Evaluation of optimization methods for nonrigid medical image registration using mutual information and b-splines,” *IEEE transactions on image processing*, vol. 16, no. 12, pp. 2879–2890, 2007.

[46] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. New York, NY, USA: Cambridge University Press, 2007.

[47] T. Vercauteren, X. Pennec, A. Perchant, and N. Ayache, “Diffeomorphic demons: efficient non-parametric image registration,” *NeuroImage*, vol. 45, no. 1 Suppl, pp. S61–72, Mar 2009.

[48] P. Thevenaz and M. Unser, “Optimization of mutual information for multiresolution image registration,” *IEEE transactions on image processing*, vol. 9, no. 12, pp. 2083–2099, 2000.

[49] W. R. Crum, D. L. G. Hill, and D. J. Hawkes, “Information theoretic similarity measures in non-rigid registration,” *Information processing in medical imaging : proceedings of the conference*, vol. 18, pp. 378–87, Jul 2003.

[50] B. Horn and B. Schunck, “Determining optical flow,” *Artificial intelligence*, vol. 17, no. 1-3, pp. 185–203, Jan 1981.

[51] X. Pennec, P. Cachier, and N. Ayache, “Understanding the” demon’s algorithm”: 3D non-rigid registration by gradient descent,” in *MICCAI’99*, 1999, pp. 597–605.

[52] W. Rossmann, *Lie groups: an introduction through linear groups*. Oxford University Press, Jan 2002.

[53] B. T. T. Yeo, M. R. Sabuncu, T. Vercauteren, N. Ayache, B. Fischl, and P. Golland, “Spherical demons: fast diffeomorphic landmark-free surface registration,” *IEEE Transactions on Medical Imaging*, vol. 29, no. 3, pp. 650–68, Mar 2010.

[54] T. Vercauteren, X. Pennec, A. Perchant, and N. Ayache, “Symmetric log-domain diffeomorphic registration: A demons-based approach,” in *Proc. MICCAI’08*, vol. 5241, 2008, pp. 754–761.

[55] P. Freeborough and N. Fox, “Modeling brain deformations in Alzheimer disease by fluid registration of serial 3D MR images,” *Journal of Computer Assisted Tomography*, vol. 22, pp. 838–843, 1998.

[56] N. Cahill, J. Noble, and D. Hawkes, “Fourier methods for nonparametric image registration,” in *Computer Vision and Pattern Recognition, 2007. CVPR ’07. IEEE Conference on*, 2007, pp. 1 – 8.

[57] G. E. Christensen and H. Johnson, “Consistent image registration,” *IEEE Transactions on Medical Imaging*, vol. 20, no. 7, pp. 568–582, 2001.

[58] Cachier and Rey, “Symmetrization of the non-rigid registration problem using inversion-invariant energies: Application to multiple sclerosis,” in *Proc. MICCAI’00*, vol. 1935, 2000, pp. 697–708.

[59] M. Beg, M. I. Miller, A. Trouve, and L. Younes, “Computing large deformation metric mappings via geodesic flows of diffeomorphisms,” *Int J Comput Vision*, vol. 61, no. 2, pp. 139–157, 2005.

[60] L. Risser, F. Vialard, M. Murgasova, and D. Holm, “Large deformation diffeomorphic registration using fine and coarse strategies,” in *Workshop on Biomedical Image Registration*, vol. 6204, 2010, pp. 186–197.

[61] M. Hernandez and A. F. Frangi, “Non-parametric geodesic active regions: method and evaluation for cerebral aneurysms segmentation in 3DRA and CTA,” *Medical Image Analysis*, vol. 11, no. 3, pp. 224–41, Jun 2007.

[62] M. Hernandez, M. Bossa, and S. Olmos, “Registration of anatomical images using paths of diffeomorphisms parameterized with stationary vector field flows,” *International Journal of Computer of Computer Vision*, vol. 85, pp. 291–306, Jan 2009.

[63] J. Ashburner and K. J. Friston, “Nonlinear spatial normalization using basis functions,” *Hum. Brain Mapp.*, vol. 7, pp. 254–266, 1999.

[64] D. Shen and C. Davatzikos, “HAMMER: Hierarchical attribute matching mechanism for elastic registration,” *IEEE Transactions on Medical Imaging*, vol. 21, no. 11, pp. 1421–1439, 2002.

[65] E. Bardinet, L. Cohen, and N. Ayache, “Tracking and motion analysis of the left ventricle with deformable superquadrics,” *Medical Image Analysis*, vol. 1, no. 2, pp. 129–149, 1996.

[66] A. Lee, G. Wolberg, and S. Shin, “Scattered data interpolation with multilevel b-splines,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 3, no. 3, pp. 228–244, 1997.

[67] A. Lee, G. Wolberg, K. Chwa, and S. Shin, “Image metamorphosis with scattered feature constraints,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 2, no. 4, pp. 337–354, 1996.

[68] M. Sdika, “A fast nonrigid image registration with constraints on the jacobian using large scale constrained optimization,” *IEEE Transactions on Medical Imaging*, vol. 27, no. 2, pp. 271–281, 2008.

[69] T. Rohlfing, C. R. M. Jr, D. Bluemke, and M. Jacobs, “Volume-preserving nonrigid registration of MR breast images using free-form deformation with an incompressibility constraints,” *IEEE Transactions on Medical Imaging*, vol. 22, no. 6, pp. 730–741, 2003.

[70] D. Rueckert, P. Aljabar, R. A. Heckemann, J. V. Hajnal, and A. Hammers, “Diffeomorphic registration using b-splines,” *Medical image computing and computer-assisted intervention : MICCAI International Conference on Medical Image Computing and Computer-Assisted Intervention*, vol. 9, no. Pt 2, pp. 702–9, Jan 2006.

[71] Y. Choi and S. Lee, “Injectivity conditions of 2D and 3D uniform cubic b-spline functions,” *Graphical Models*, vol. 62, no. 6, pp. 411 – 427, 2000.

[72] M. De Craene, O. Camara, B. Bijnens, and A. F. Frangi, “Large diffeomorphic FFD registration for motion and strain quantification from 3D-US sequences,” in *Functional Imaging and Modeling of the Heart*, 2009, pp. 437–446.

[73] M. De Craene, G. Piella, N. Duchateau, E. Silva, A. Doltra, H. Gao, J. D’hooge, O. Camara, J. Brugada, M. Sitges, and A. Frangi, “Temporal diffeomorphic free-form deformation for strain quantification in 3D-US images,” in *MICCAI’2010*, 2010.

[74] J. A. Schnabel, D. Rueckert, M. Quist, J. Blackall, A. Castellano-Smith, T. Hartkens, G. Penney, W. Hall, D. Liu, and C. Truwit, “A generic framework for non-rigid registration based on non-uniform multi-level free-form deformations,” in *Lecture Notes in Computer Science*, 2001, pp. 573–581.

[75] S. C. Joshi and M. I. Miller, “Landmark matching via large deformation diffeomorphisms,” *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, vol. 9, no. 8, pp. 1357–70, Jan 2000.

[76] S. Durrleman, X. Pennec, A. Trouvé, P. Thompson, and N. Ayache, “Inferring brain variability from diffeomorphic deformations of currents: an integrative approach,” *Medical Image Analysis*, vol. 12, no. 5, pp. 626–37, Oct 2008.

[77] R. Szeliski and S. Lavallée, “Matching 3-d anatomical surfaces with non-rigid deformations using octree-splines,” *International Journal of Computer Vision*, vol. 18, pp. 171–186, 1996, 10.1007/BF00055001.

[78] R. Shams, P. Sadeghi, R. Kennedy, and R. Hartley, “A survey of medical image registration on multicore and the gpu,” *Signal Processing Magazine, IEEE DOI - 10.1109/MSP.2009.935387*, vol. 27, no. 2, pp. 50–60, 2010.

[79] O. Fluck, C. Vetter, W. Wein, A. Kamen, B. Preim, and R. Westermann, “A survey of medical image registration on graphics hardware,” *Comput Methods Programs Biomed*, 2011, in press.

[80] R. Strzodka, M. Droske, and M. Rumpf, “Fast image registration in dx9 graphics hardware,” *Journal of Medical Informatics and Technologies*, vol. 6, no. 43-49, p. 143, 2003.

[81] C. Vetter, C. Guetter, C. Xu, and R. Westermann, “Non-rigid multi-modal registration on the gpu,” in *Proc of SPIE*, vol. Proc. SPIE 6512, 2007.

[82] Z. Fan, C. Vetter, C. Guetter, D. Yu, and C. Xu, “Optimized GPU implementation of learning-based non-rigid multi-modal registration,” in *SPIE*, vol. Proc. SPIE 6914, 2008.

[83] G. C. Sharp, N. Kandasamy, H. Singh, and M. Folkert, “Gpu-based streaming architectures for fast cone-beam ct image reconstruction and demons deformable registration,” *Phys Med Biol*, vol. 52, no. 19, pp. 5771–83, Oct 2007.

[84] N. Courty and P. Hellier, “Accelerating 3d non-rigid registration using graphics hardware,” *International Journal of Image and Graphics*, vol. 8, no. 1, pp. 81–98, 2008.

[85] S. S. Samant, J. Xia, P. Muyan-Ozcelik, and J. D. Owens, “High performance computing for deformable image registration: towards a new paradigm in adaptive radiotherapy,” *Med. Phys.*, vol. 35, no. 8, pp. 3546–53, Aug 2008.

[86] X. Gu, H. Pan, Y. Liang, R. Castillo, D. Yang, D. Choi, E. Castillo, A. Majumdar, T. Guerrero, and S. B. Jiang, “Implementation and evaluation of various demons deformable image registration algorithms on a gpu,” *Phys Med Biol*, vol. 55, no. 1, pp. 207–19, Jan 2010.

[87] K. Ø. Noe, B. D. D. Senneville, U. V. Elstrøm, K. Tanderup, and T. S. Sørensen, “Acceleration and validation of optical flow based deformable registration for image-guided radiotherapy,” *Acta Oncol*, vol. 47, no. 7, pp. 1286–93, Jan 2008.

[88] K. Ø. Noe, K. Tanderup, J. C. Lindegaard, C. Grau, and T. S. Sørensen, “Gpu accelerated viscous-fluid deformable registration for radiotherapy,” *Stud Health Technol Inform*, vol. 132, pp. 327–32, Jan 2008.

[89] G. Soza, M. Bauer, P. Hastreiter, C. Nimsky, and G. Greiner, “Non-rigid registration with use of hardware-based 3d bézier functions,” in *MICCAI’02*, vol. LNCS 2489, no. 2, 2002, pp. 549–556.

[90] B. Li, A. A. Young, and B. R. Cowan, “GPU accelerated non-rigid registration for the evaluation of cardiac functions,” in *MICCAI’08*, vol. LNCS 5242, no. 2, 2008, pp. 880–887.

[91] W. Plishker, O. Dandekar, S. S. Bhattacharyya, and R. Shekhar, “Towards systematic exploration of tradeoffs for medical image registration on heterogeneous platforms,” in *IEEE Biomedical Circuits and Systems Conference*, 2008, pp. 53 – 56.

[92] M. Teßmann, C. Eisenacher, F. Enders, M. Stamminger, and P. Hastreiter, “GPU accelerated normalized mutual information and B-spline transformations,” in *Eurographics Workshop on Visual Computing for Biomedicine*, Jan 2008, pp. 117—124.

[93] D. Ruijters, B. M. T. H. Romeny, and P. Suetens, “Gpu-accelerated elastic 3d image registration for intra-surgical applications,” *Comput Methods Programs Biomed*, 2011, in press.

[94] R. J. Lapeer, S. K. Shah, and R. S. Rowland, “An optimised radial basis function algorithm for fast non-rigid registration of medical images,” *Comput Biol Med*, vol. 40, no. 1, pp. 1–7, Jan 2010.

[95] G. R. Joldes, A. Wittek, M. Couton, S. K. Warfield, and K. Miller, “Real-time prediction of brain shift using nonlinear finite element algorithms,” in *MICCAI09*, no. Pt 2, 2009, pp. 300–7.

[96] O. Fluck, S. Aharon, D. Cremers, and M. Rousson, “Gpu histogram computation,” in *SIGGRAPH ’06: SIGGRAPH 2006 Research posters*, 2006.

[97] T. Scheuermann and J. Hensley, “Efficient histogram generation using scattering on gpus,” in *I3D ’07: Proceedings of the 2007 symposium on Interactive 3D graphics and games*, 2007.

[98] R. Shams, P. Sadeghi, R. Kennedy, and R. Hartley, “Parallel computation of mutual information on the gpu with application to real-time registration of 3d medical images,” *Comput Methods Programs Biomed*, vol. 99, no. 2, pp. 133–46, Aug 2010.

[99] J. Jiang, W. Luk, and D. Rueckert, “FPGA-based computation of free-form deformations in medical image registration,” in *Field-Programmable Technology, IEEE International Conference on*, 2003, pp. 234– 241.

[100] T. Rohlfing and C. R. M. Jr, “Nonrigid image registration in shared-memory multiprocessor environments with application to brains, breasts, and bees,” *IEEE Transactions on Information Technology in Biomedicine*, vol. 7, no. 1, pp. 16–25, 2003.

[101] J. Rohrer, L. Gong, and G. Székely, “Parallel mutual information based 3D non-rigid registration on a multi-core platform,” in *High-Performance MICCAI*, 2008.

[102] Nvidia, *NVIDIA CUDA Programming Guide, version 2.3*, 2009.

[103] D. Chan, N. C. Fox, R. I. Scahill, W. R. Crum, J. L. Whitwell, G. Leschziner, A. M. Rossor, J. M. Stevens, L. Cipolotti, and M. N. Rossor, “Patterns of temporal lobe atrophy in semantic dementia and Alzheimer’s disease,” *Annals of Neurology*, vol. 49, no. 4, pp. 433–42, Apr 2001.

[104] M. Jenkinson and S. M. Smith, “A global optimisation method for robust affine registration of brain images,” *Medical Image Analysis*, vol. 5, no. 2, pp. 143–156, 2001.

[105] S. M. Smith, “Fast robust automated brain extraction,” *Hum. Brain Mapp.*, vol. 17, no. 3, pp. 143–155, Nov 2002.

[106] T. Rohlfing, R. Brandt, R. Menzel, and C. R. M. Jr, “Evaluation of atlas selection strategies for atlas-based image segmentation with application to confocal microscopy images of bee brains,” *NeuroImage*, vol. 21, no. 4, pp. 1428–1442, 2004.

[107] J. Ashburner, J. L. Andersson, and K. J. Friston, “Image registration using a symmetric prior–in three dimensions,” *Hum. Brain Mapp.*, vol. 9, no. 4, pp. 212–25, Apr 2000.

[108] B. Karaçali and C. Davatzikos, “Estimating topology preserving and smooth displacement fields,” *IEEE Trans. Med. Imag.*, vol. 23, no. 7, pp. 868–80, Jul 2004.

[109] C. Tanner, J. Schnabel, A. Degenhard, A. Castellano-Smith, C. Hayes, M. Leach, D. Hose, D. Hill, and D. Hawkes, “Validation of volume-preserving non-rigid registration: Application to contrast-enhanced mr-mammography,” in *Proc. MICCAI’02*, vol. 2488, 2002, pp. 307–314.

[110] Y. Yin, E. Hoffman, and C. Lin, “Mass preserving nonrigid registration of ct lung images using cubic b-spline,” *Med. Phys.*, vol. 36, pp. 4213–4222, 2009.

[111] K. Ding, Y. Youbing, K. Cao, G. Christensen, C. Lin, and J. Hoffman, E.A.and Reinhardt, "Evaluation of lobar biomechanics during respiration using image registration," in *Proc. MICCAI'09*, vol. 5761, 2009, pp. 739–746.

[112] J. G. Sled, A. P. Zijdenbos, and A. C. Evans, "A nonparametric method for automatic correction of intensity nonuniformity in MRI data." *IEEE Trans. Med. Imag.*, vol. 17, no. 1, pp. 87–97, Feb. 1998.

[113] M. S. Cohen, R. M. DuBois, and M. M. Zeineh, "Rapid and effective correction of RF inhomogeneity for high field magnetic resonance imaging." *Hum Brain Mapp*, vol. 10, no. 4, pp. 204–211, Aug. 2000.

[114] U. Vovk, F. Pernus, and B. Likar, "A review of methods for correction of intensity inhomogeneity in MRI," *IEEE Transactions on Medical Imaging*, vol. 26, no. 3, pp. 405–421, March 2007.

[115] R. I. Scahill and N. C. Fox, "Longitudinal imaging in dementia," *Br J Radiol*, vol. 80 Spec No 2, pp. S92–8, Dec 2007.

[116] P. A. Freeborough and N. C. Fox, "The boundary shift integral: an accurate and robust measure of cerebral volume changes from registered repeat MRI." *IEEE Transactions on Medical Imaging*, vol. 16, no. 5, pp. 623–629, Oct. 1997.

[117] R. G. Boyes, D. Rueckert, P. Aljabar, J. Whitwell, J. M. Schott, D. L. G. Hill, and N. C. Fox, "Cerebral atrophy measurements using Jacobian integration: comparison with the boundary shift integral." *Neuroimage*, vol. 32, no. 1, pp. 159–169, Aug. 2006.

[118] E. B. Lewis and N. C. Fox, "Correction of differential intensity inhomogeneity in longitudinal MR images." *Neuroimage*, vol. 23, no. 1, pp. 75–83, Sep. 2004.

[119] C. Studholme, V. Cardenas, E. Song, F. Ezekiel, A. Maudsley, and M. Weiner, "Accurate template-based correction of brain MRI intensity distortion with application to dementia and aging," *IEEE Transactions on Medical Imaging*, vol. 23, no. 1, pp. 99–110, Jan. 2004.

[120] C. Studholme, C. Drapaca, B. Iordanova, and V. Cardenas, "Deformation-based mapping of volume change from serial brain MRI in the presence of local tissue contrast change." *IEEE Transactions on Medical Imaging*, vol. 25, no. 5, pp. 626–639, May 2006.

[121] X. Zhuang, D. J. Hawkes, and S. Ourselin, "Unifying encoding of spatial information in mutual information for nonrigid registration." *Inf Process Med Imaging*, vol. 21, pp. 491–502, 2009.

[122] C. Broit, "Optimal registration of deformed images," Ph.D. dissertation, University of Pennsylvania, 1981.

[123] R. Bajcsy and S. Kovačič, "Multiresolution elastic matching," *Comput. Vision Graph. Image Process.*, vol. 46, pp. 1–21, 1989.

[124] I. Yanovsky, C. Le Guyader, A. Leow, P. Thompson, and L. Vese, “Nonlinear elastic registration with unbiased regularization in three dimensions,” in *Computational Biomechanics for Medicine III, MICCAI’08 workshop*, 2008.

[125] G. Holzapfel, *Nonlinear Solid Mechanics: A Continuum Approach for Engineering*. Chichester: John Wiley & Sons, 2000.

[126] K.-J. Bathe, *Finite Element Procedures*. Prentice Hall, 1996.

[127] K. Miller, G. Joldes, D. Lance, and A. Wittek, “Total Lagrangian explicit dynamics finite element algorithm for computing soft tissue deformation,” *Communications in Numerical Methods in Engineering*, vol. 23, p. 121, 2007.

[128] G. Szekely, C. Brechbühler, R. Hutter, A. Rhomberg, N. Ironmonger, and P. Schmid, “Modelling of soft tissue simulation for laparscopic surgery simulation,” *Medical Image Analysis*, vol. 4, pp. 57–66, 2000.

[129] Z. Taylor, M. Cheng, and S. Ourselin, “High-speed nonlinear finite element analysis for surgical simulation using graphics processing units,” *IEEE Transactions on Medical Imaging*, vol. 27, pp. 650–663, 2008.

[130] D. Loeckx, “Automated nonrigid intra-patient image registration using b-splines,” Ph.D. dissertation, Katholieke Universiteit Leuven, 2006.

[131] W. R. Crum, D. Rueckert, M. Jenkinson, D. Kennedy, and S. M. Smith, “A framework for detailed objective comparison of non-rigid registration algorithms in neuroimaging,” in *7th International Conference on Medical Image Computing and Computer Assisted Intervention*, vol. 3216, 2004, pp. 679–686.

[132] P. Hellier, C. Barillot, I. Corouge, B. Gibaud, G. L. Goualher, D. L. Collins, A. Evans, G. Malandain, N. Ayache, G. E. Christensen, and H. J. Johnson, “Retrospective evaluation of intersubject brain registration,” *IEEE Trans Med Imaging*, vol. 22, pp. 1120–1130, 2003.

[133] J. A. Schnabel, C. Tanner, A. D. Castellano-Smith, A. Degenhard, M. O. Leach, D. R. Hose, D. L. G. Hill, and D. J. Hawkes, “Validation of nonrigid image registration using finite-element methods: application to breast MR images,” *IEEE Trans Med Imaging*, vol. 22, pp. 238–247, 2003.

[134] O. Camara, J. A. Schnabel, G. R. Ridgway, W. R. Crum, A. Douiri, R. I. Scahill, D. L. G. Hill, and N. C. Fox, “Accuracy assessment of global and local atrophy measurement techniques with realistic simulated longitudinal Alzheimer’s disease images.” *Neuroimage*, vol. 42, pp. 696–709, 2008.

[135] S. Klöppel, C. M. Stonnington, C. Chu, B. Draganski, R. I. Scahill, J. D. Rohrer, N. C. Fox, C. R. Jack, J. Ashburner, and R. S. J. Frackowiak, “Automatic classification of MR scans in Alzheimer’s disease.” *Brain*, vol. 131, pp. 681–689, 2008.

[136] R. G. Newcombe, “Improved confidence intervals for the difference between binomial proportions based on paired data.” *Stat Med*, vol. 17, pp. 2635–2650, 1998.

[137] A. Klein, J. Andersson, B. Ardekani, J. Ashburner, B. Avants, M. Chiang, G. Christensen, D. Collins, J. Gee, P. Hellier *et al.*, “Evaluation of 14 nonlinear deformation algorithms applied to human brain MRI registration,” *Neuroimage*, vol. 46, no. 3, pp. 786–802, 2009.

[138] D. Rueckert, A. F. Frangi, and J. A. Schnabel, “Automatic construction of 3-D statistical deformation models of the brain using nonrigid registration,” *IEEE Trans. Med. Imag.*, vol. 22, no. 8, pp. 1014–25, Aug 2003.

[139] S. Joshi, B. Davis, M. Jomier, and G. Gerig, “Unbiased diffeomorphic atlas construction for computational anatomy,” *Neuroimage*, vol. 23, pp. 151–160, 2004.

[140] A. Rao, R. Chandrashekara, G. Sanchez-Ortiz, R. Mohiaddin, P. Aljabar, J. Hajnal, B. Puri, and D. Rueckert, “Spatial transformation of motion and deformation fields using nonrigid registration,” *IEEE Trans. Med. Imag.*, vol. 23, no. 9, pp. 1065–1076, 2004.

[141] C. Moler and C. Van Loan, “Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later,” *SIAM Review*, vol. 45, no. 1, pp. 3–49, 2003.

[142] K. Murphy, B. van Ginneken, J. Reinhardt, S. Kabus, K. Ding, X. Deng, K. Cao, K. Du, G. Christensen, V. Garcia, T. Vercauteren, N. Ayache, O. Commowick, G. Malandain, B. Glocker, N. Paragios, N. Navab, V. Gorbunova, J. Sporring, M. de Bruijne, X. Han, M. Heinrich, J. Schnabel, M. Jenkinson, C. Lorenz, M. Modat, J. McClelland, S. Ourselin, S. Muenzing, M. Viergever, D. D. Nigris, D. Collins, T. Arbel, M. Peroni, R. Li, G. Sharp, A. Schmidt-Richberg, J. Ehrhardt, R. Werner, D. Smeets, D. Loeckx, G. Song, N. Tustison, B. Avants, J. Gee, M. Staring, S. Klein, B. Stoel, M. Urschler, M. Werlberger, J. Vandemeulebroucke, S. Rit, D. Sarrut, and J. Pluim, “Evaluation of registration methods on thoracic ct: The empire10 challenge,” *IEEE Transactions on Medical Imaging*, 2011, 10.1109/TMI.2011.2158349, in press.

[143] K. K. Leung, J. Barnes, M. Modat, G. R. Ridgway, J. W. Bartlett, N. C. Fox, S. Ourselin, and A. D. N. Initiative, “Brain maps: an automated, accurate and robust brain extraction technique using a template library,” *NeuroImage*, vol. 55, no. 3, pp. 1091–108, Apr 2011.

[144] M. J. Cardoso, M. J. Clarkson, G. R. Ridgway, M. Modat, N. C. Fox, S. Ourselin, and A. D. N. Initiative, “Load: a locally adaptive cortical segmentation algorithm,” *NeuroImage*, vol. 56, no. 3, pp. 1386–97, Jun 2011.

[145] R. A. Heckemann, S. Keihaninejad, P. Aljabar, K. R. Gray, C. Nielsen, D. Rueckert, J. V. Hajnal, A. Hammers, and T. A. D. N. Initiative, “Automatic morphometry in alzheimer’s disease and mild cognitive impairment,” *NeuroImage*, vol. 56, no. 4, pp. 2024–2037, Jun 2011.

[146] J. R. McClelland, S. Hughes, M. Modat, A. Qureshi, S. Ahmad, D. B. Landau, S. Ourselin, and D. J. Hawkes, “Inter-fraction variations in respiratory motion models,” *Phys Med Biol*, vol. 56, no. 1, pp. 251–72, Jan 2011.

[147] H. R. Roth, J. R. McClelland, D. J. Boone, M. Modat, M. J. Cardoso, T. E. Hampshire, M. Hu, S. Punwani, S. Ourselin, G. G. Slabaugh, S. Halligan, and D. J. Hawkes, “Registration of the endoluminal surfaces of the colon derived from prone and supine ct colonography,” *Med Phys*, vol. 38, no. 6, pp. 3077–3089, Jan 2011.

[148] L. Han, J. Hipwell, T. Mertzanidou, T. Carter, M. Modat, S. Ourselin, and D. Hawkes, “A hybrid fem-based method for aligning prone and supine images for image guided breast surgery,” in *Biomedical Imaging: From Nano to Macro, 2011 IEEE International Symposium on*, 2011, pp. 1239 – 1242.

[149] S. Durrleman, X. Pennec, A. Trouve, G. Gerig, and N. Ayache, “Spatiotemporal atlas estimation for developmental delay detection in longitudinal datasets,” in *Proc. MICCAI’09*, vol. 5761, 2009, pp. 297–304.