Model-Constrained Non-Rigid Registration in Medicine

Der Technischen Fakultät der
Universität Erlangen–Nürnberg

zur Erlangung des Grades

DOKTOR–INGENIEUR

vorgelegt von

Dipl.-Inf. Volker Gerhard Daum

Erlangen — 2011

Deutscher Titel:

Modellbasierte, nichtstarre Registrierung in der Medizin

Als Dissertation genehmigt von der
Technischen Fakultät der
Universität Erlangen-Nürnberg

Dekan:	Prof. Dr.-Ing. M. Merklein
Berichterstatter:	Prof. Dr.-Ing. J. Hornegger
	Prof. K. Pohl

Acknowledgment

Scientific work is never done in isolation. It is always based on the work of others and it is enriched by the discussions we have with our scientific colleagues, friends and others.

I therefore want to thank, first and foremost, Prof. J. Hornegger. He not only brought me into the field of pattern recognition and medical image processing in the first place, but also supervised this work and always had an open ear and some encouragement for me, when it was really necessary. I am also indebted to Prof. Torsten Kuwert for providing a great source of medical knowledge and interesting, interdisciplinary discussions. I would also like to thank Prof. Kilian Pohl for reviewing this work and his many helpful comments.

I am especially grateful to Dieter Hahn and Marcus Prümmer, for all the scientific and non-scientific discussions, proof-reading my work and the fun we had and have together. I also have to thank Andreas Maier for proof-reading part of this work and for interesting discussions on the subject of computed tomography. I cannot individually thank everyone at the pattern recognition lab for the scientific exchange and for generally being great guys, so this will have to suffice. The lab is an overall great and very varied community. Keep it that way.

I would also like to thank the Max Planck Research School for Optics and Imaging (IMPRS-OPTIM) and the research project MoBiMed (subproject 01EZ0807) of the Bundesministerium für Bildung und Forschung (BMBF) for their financial support.

Last but certainly not least, I want to especially thank my family for their unwavering support. No matter what I do or what I need, they have always been there to help me or give me a little push in the right direction whenever I needed it.

Volker Daum

Abstract

The aim of image registration is to compute a mapping from one image's frame of reference to another's, such that both images are well aligned. Even when the mapping is assumed to be rigid (only rotation and translation) this can be a quite challenging task to accomplish between different image modalities. Noise and other imaging artifacts like bias fields in magnetic resonance (MR) imaging or streak artifacts in computed tomography (CT) can pose additional problems. In non-rigid image registration these problems are further compounded by the additional degrees of freedom in the transform.

Another problem is that the non-rigid registration problem is usually ambiguous: Different deformation fields can lead to equally well aligned images. Nevertheless, one would prefer deformations that coincide with medical or physiological expectations. For instance, in MR low intensity image values can indicate bones as well as air. We would prefer a registration result that only maps bone to bone and air to air, even though matching air to bone might lead to a visually similar result.

This work strives to address some of these problems. In a first step we provide a solid non-rigid registration algorithm. We compare several optimization algorithms, to ensure that the registration result is at least numerically as good as possible. We also explore how the parameter determining the global stiffness of the computed transform can be specified in a way that yields predictable results. In a second step we want to integrate prior information about the desired deformation into this registration algorithm. Two types of prior information are considered in this work:

The first are known point correspondences that explicitly specify the desired deformation for some parts of the images. This provides a very straightforward way for a user to interact with the registration algorithm. The known correspondences are efficiently integrated into the registration algorithm, which allows the specification of arbitrary number of correspondences and the application of the approach in 2-D and 3-D. As the landmarks are treated as hard constraints it is guaranteed that they are matched exactly. It is shown that this additional information can immensely benefit the registration result, especially in difficult cases like the registration of relatively unrelated imaging modalities like positron emission tomography (PET) and CT.

The second type of information is provided in the form of training deformations reflecting the kinds of deformation usually encountered in an application. These are used to generate a model which can be used to guide the registration to a result that is similar to the training data. We consider two variants of statistical deformation models. Either the model is generated and applied on the deformations themselves or on their Laplacian. The latter has the advantage of being inherently invariant to remaining rigid misalignments in the training data. They are applied in the context of atlas registration for MR/PET attenuation correction. An template CT image is registered with the patient MR to generate a pseudo-CT of the patient that can be used for the PET attenuation correction. However, the different intensity distributions in CT and MR, effects like bias fields and the low inter-slice resolution common in MR imaging, make the multi-modal registration prone to errors. The deformation model, learned from a set of mono-modal registrations, is used to constrain and thus improve the multi-modal registration. The algorithm is evaluated on a set of patient data for which the ground-truth CT scan is available. This allows the evaluation of the atlas registration results through a direct comparison with the ground truth CT data. Our experiments show that the registration employing the statistical deformation models yields generally improved results.

Übersicht

Die Bildregistrierung in der Medizin hat zur Aufgabe eine Abbildung zu berechnen die zwei Bilder in ein gemeinsames Koordinatensystem überführt. Selbst wenn nur eine starre Abbildung gesucht ist, ist dies eine Herausforderung wenn unterschiedliche Bildmodalitäten kombiniert werden sollen. Weiter erschwert wird diese Aufgabe durch die zur Verfügung stehende Bildqualität die oftmals unter Rauschen und anderen Artefakten zu leiden hat. In der nichtstarren Bildregistrierung kommt zusätzlich die Vielzahl der Freiheitsgrade in der Transformation erschwerend hinzu.

Ein weiteres Problem speziell bei der nichtstarren Registrierung ist, dass die Aufgabenstellung meist nicht eindeutig ist: Unterschiedliche Deformationen können zu ähnlich gut aussehenden Ergebnissen führen. Dennoch sind nicht alle diese Deformationen in jeder Anwendung gleich gut, da man in der Regel gewisse medizinisch oder biologisch motivierte Erwartungen an die Art der Abbildung hat. Beispielsweise zeigen viele Datensätze aus der Magnetresonanztomographie (MRT) ähnliche Intensitätswerte für Luft und Knochen. In der Anwendung würde man jedoch Deformationen bevorzugen die Luft auf Luft und Knochen auf Knochen abbilden, obwohl eine Abbildung von Luft auf Knochen visuell das gleiche Ergebnis liefern kann.

Diese Arbeit hat zum Ziel einige dieser Probleme anzugehen und entsprechende Lösungsvorschläge zu präsentieren. Dazu führen wir zunächst einen soliden nichtstarren Registrierungsalgorithmus ein. Wir vergleichen dabei mehrere nichtlineare Optimierungsalgorithmen um sicherzustellen, dass zumindest ein numerisch gutes Ergebnis erzeugt wird. Um die Parametrierung zu vereinfachen wird außerdem untersucht, wie der globale Parameter der die Steifheit der Abbildung steuert so gesetzt werden kann, dass ein vorhersagbares Ergebnis erzeugt wird. In einem weiteren Schritt wird gezeigt wie zusätzliches Vorwissen über die gewünschte Deformation in die Registrierung eingebracht werden kann. In dieser Arbeit werden dabei zwei unterschiedliche Arten von Vorwissen näher betrachtet:

Die erste Variante von Vorwissen sind Punktkorrespondenzen die für einzelne Bildteile die gewünschte Deformation fest vorgeben. Dies erlaubt einem Anwender eine sehr direkte Interaktion mit dem Registrierungsalgorithmus. Die Punktkorrespondenzen werden effizient in den Algorithmus integriert, was die Benutzung von einer beliebigen Anzahl solcher Landmarken sowohl in 2-D wie auch 3-D Bilddaten erlaubt. Da die Landmarken als strikte Bedingungen behandelt werden, kann ihre exakte Abbildung aufeinander garantiert werden. Anhand eines praktischen Beispiels wird gezeigt, dass dieses zusätzliche Wissen besonders in schwierigen Anwendungsfällen, wie der Fusion von Daten aus Positronen-Emissions-Tomographen (PET) und Computertomographen (CT) eine große Verbesserung des Ergebnisses zur Folge haben kann.

Die zweite Art von Vorwissen mit der wir uns beschäftigen sind bekannte Trainingsdeformationen die man bereits in derselben Anwendung beobachtet hat. Aus diesen kann man statistische Modelle erzeugen, mit denen der Registrierungsalgorithmus so gelenkt wird, dass er Ergebnisse erzeugt die den Trainingsdaten ähnlich sind. Die Modelle werden dabei entweder direkt auf den Trainingsdaten oder auf deren zweiten Ableitungen (Laplace) berechnet. Der Zweite Ansatz hat hierbei den Vorteil gegen starre Fehlausrichtungen der Lerndaten robust zu sein. Beide Methoden werden auf das praktische Problem der MRT/PET Schwächungskorrektur mittels Atlasregistrierung angewandt. Dabei wird ein Atlas CT auf einen Patienten MRT Datensatz registriert um ein pseudo CT Bild dieses Patienten zu erstellen. Dieses kann dann in einem weiteren Schritt für eine Schwächungskorrektur einer PET Aufnahme aus einem hybriden MRT/PET Gerät verwendet werden. Diese Art der multimodalen Registrierung ist jedoch Aufgrund der unterschiedlichen Intensitätsverteilungen in den Bildern, Bildartefakten und unterschiedlichen Bildauflösungen, fehleranfällig. Das Deformationsmodell wird in dieser Anwendung mit Trainingsdaten aus monomodalen (CT/CT) Registrierungen generiert. In der multimodalen Registrierung kann es dann die möglichen Deformationen zusätzlich beschränken und so das Endergebnis verbessern. Dieser Ansatz wird auf einem Datenbestand von MRT und dazugehörigen CT Datensätzen evaluiert. Die vorhandenen CT Daten erlauben bei dieser Evaluierung den direkten Vergleich zwischen den generierten pseudo CT Daten mit den tatsächlichen im Anwendungsfall nicht verfügbaren CT Daten. In diesen Experimenten können wir zeigen, dass unser Ansatz gegenüber einem Standardansatz ohne Deformationsmodelle ein deutlich verbessertes Ergebnis liefert.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Over the course of the past forty years medical diagnostics was revolutionized by the invention of multiple modalities that are able to view the human body and its function in full 3-D. Where computed tomography (CT) is able to provide detailed insights into the human anatomy, nuclear medicine methods like positron emission tomography (PET) and single photon emission computed tomography (SPECT) allow to visualize processes like perfusion or enhanced tissue growth due to cancer proliferation. Magnetic resonance (MR) imaging is generally able to provide good soft tissue imaging combined with the capability to do functional imaging. As all of these techniques have their individual advantages and disadvantages their combination has been an area of interest for a long time. This is usually done either by retrospective image registration that tries to compute a transform between two separately acquired images, or by acquiring more than one imaging modality "simultaneously" in the same hybrid scanner. Similarly, combining images from longitudinal studies by means of image registration allows qualitative and quantitative analysis of the progression of various diseases and how they are affected by treatment. Image registration also allows the combination of high quality preoperative images with data acquired intraoperatively.

The number of registration techniques and variants is almost as varied as their applications. Rigid and non-rigid, parametric or non-parametric, numerous distance measures and regularizers, not to name the possible options of optimization schemes, constitute a rich choice of components for performing the registration task. While in the area of rigid registration the quality of the resulting match can be relatively well assessed, this is considerably harder in the non-rigid case, since in virtually all non-trivial real cases the ground truth deformation is not known. All attempts at generating synthetic ground truth datasets are likely to favor registration methods that model the deformation field similarly to the method used for generating the synthetic deformations. Non-rigid registration is also relatively difficult to apply in practice, due to the many degrees of freedom it offers in terms of the deformation that is generated. Usually the user is responsible to select the degree of non-rigidity allowed in the transform, which is often difficult as there is often a lack of intuition about the user set parameters that determine the amount of non-rigidity. Last, but not least, it is often difficult to assess whether a non-rigid registration result is only visually appealing or makes actual medical and physiological sense.

This work is therefore concerned with the incorporation of prior knowledge about the desired deformation into the registration algorithm. Prior knowledge is already input by the choice of distance measure, regularizer and the parameter governing the non-rigidity of the transform, which we will aim to make a more intuitive choice for the user. More advanced forms of prior knowledge we will examine are known point correspondences and especially statistical information about the deformations most often encountered in a particular application scenario.

1.1 Overview

The aim of this work is to examine possibilities to make non-rigid registration easier and more robust to use in practice. To this end current non-rigid, non-parametric registration techniques, distance measures and regularizers and their respective behavior in combination are examined. Furthermore different optimization techniques are examined as the study of existing and the development of new mathematical terms can only make sense if they are sufficiently optimized and perceived problems with a particular term are not just due to a bad optimization. The main focus, however, is on incorporating prior knowledge about the registration subject that makes it more likely that the results are in accordance with the expectation of the user. The types of prior knowledge employed in this context are known point-to-point correspondences (landmarks) and a statistical model of the expected transform based on a principal component analysis (PCA).

The incorporation of landmarks is discussed in 5.1. The approach proposed in this work allows the incorporation of arbitrary numbers of landmarks which are matched exactly, works in 2-D as well as in 3-D and is shown to significantly improve the difficult problem of a retrospective registration of a full body PET scan with a CT scan.

The theory of the second method, which incorporates prior information through statistical deformation models into the registration algorithm, is discussed in Section 5.2. The practical application of these models is then evaluated on the problem of atlas registration for MR/PET attenuation correction. Attenuation correction is a post-processing step in PET and SPECT imaging that corrects for the attenuation the tracer radiation undergoes while it travels from the point of the radiation emission to the detector. The necessary attenuation information is usually provided by a CT or a transmission scan. In a MR/PET hybrid scanner the MR image cannot directly provide such an attenuation map. Instead, an attenuation map can be generated by non-rigidly registering an atlas CT to the MR.

1.2 Contribution

This work brings several contributions to the field of non-rigid image registration. In short these can be summarized as:

- comparison of a set of optimization algorithms for non-rigid, non-parametric, energy regularized registration

- easier user interaction for the parameter governing the stiffness of the deformation, through
  - a physical relation for changes to the parameter
  - a rescaling of the distance measure and regularizer terms of the registration, such that the impact of their choice on the stiffness parameter is lessened
- introduction of prior information into the registration through
  - the specification of additional landmarks
  - PCA based deformation models
  - application of deformation models to MR/PET attenuation correction

The first two contributions aim at simplifying the practical application of the registration algorithm by providing a good automatically parameterized optimization and a good intuition for the remaining non-numeric stiffness parameter. The prior information methods make the registration results more likely to correspond with user expectations, even for difficult registration problems, which are subject to image artifacts (noise, bias fields, streak artifacts etc.) and ambiguities.

In more detail, the first contribution is concerned with the comparison of several optimization algorithms for a non-rigid, non-parametric registration that makes use of an energy regularization term on the deformation field itself to keep it smooth. The optimization of the terms in the registration formulation is rather non-trivial and in practice the result of a registration can differ significantly depending on the optimization algorithm and the resulting quality of the optimization. We therefore compare the standard semi-implicit gradient descent scheme [Mode 04] with several Newton type optimization algorithms, in a single- and multi-level setting, both for a sum of squared differences and a mutual information distance measure.

In addition to the numeric problems we also consider the practical usability of the non-rigid registration algorithm. The major problem here is the choice of parameters. While the numerical parameters like step sizes, numbers of iterations, image quantization etc. can be estimated, the parameter governing the stiffness is a fundamental choice of the user. To make this choice easier we attach a physical model to this parameter that makes it easier for the user to predict the result of a parameter change. Additionally, some rescalings of the distance measure and regularizer terms to a uniform value range are presented. If such a rescaling succeeds it has the benefit that different types of distance measures and regularizers can be exchanged without the need to change the stiffness parameter.

The incorporation of prior information into the registration is also aimed at simplifying the interaction of the user with the algorithm. The work proposes a way to constrain the non-rigid registration by additional landmarks. The registration thus has to compute a deformation that guarantees an exact match at the landmarks. This helps to guide the registration and can help to improve the trustworthiness of the registration result. To add landmarks to an image intensity driven registration approach is not a new idea [Fisc 03a, John 02, Hart 02, Ursc 06]. However, the way we mathematically integrate these constraints is, to our knowledge, novel in the domain of image registration. The point correspondences are treated mathematically as Dirichlet boundary conditions for the computed vector field. This way the numerical registration problem gets computationally smaller instead of larger, the more landmarks are added. This way adding even large sets of landmarks, for example rigidly registered bones or other segmented and separately registered regions, is no problem.

The final contribution of this work is the generation and practical application of PCA based model regularization terms to a non-rigid, non-parametric registration approach. This encompasses a PCA model generated directly on the learning deformations, augmented to be invariant to translations, as well as a model based on the curvature of the deformations, which is invariant to rotations and translations by construction. The different approaches are evaluated on the problem of MR/PET attenuation correction through atlas registration.

Chapter 2

Medical Background

As this work is concerned with the processing of medical images, this chapter will shortly review the most important 3-D medical imaging techniques. This information is needed primarily to further the understanding of the application scenarios referred to in the work. For a more in-depth discussion of these imaging modalities please refer to [Doss 08].

2.1 Computed Tomography (CT)

Wilhelm Conrad Röntgen was the first to extensively study X-ray radiation in 1895. It allowed for the first time the non-invasive imaging of the interior of the human body. X-ray radiation is generated by accelerating electrons from a cathode to an anode in a vacuum tube. Upon hitting the anode the kinetic energy of the electrons is transformed in part into an electromagnetic radiation known as X-rays. This high energy radiation has the property that it can pass through solid materials while undergoing attenuation. The attenuation is due to the radiation interacting with the material in the form of absorption and scattering. The amount of attenuation that takes place is dependent on the density of the material being imaged and the energy of the radiation. Bone for instance is a rather dense material and will therefore attenuate the radiation much more than soft tissue.

The amount of radiation that can be observed at the detector is governed by the attenuation law

$$I = I_0 e^{\int_L \mu(l) \, \mathrm{d}l}, \tag{2.1}$$

where I_0 is the radiation generated by the X-ray tube and I the radiation that can be detected after attenuation. L is the ray from the tube to the detector and μ the function of attenuation coefficients along the ray i. e. how much the tissue at location l attenuates the radiation. Basically, μ is the material property that we want to measure. In computed tomography (CT) one is not interested in the resulting radiation I but rather in the coefficients μ in the volume of interest (i. e. the patient). To compute these values it is necessary to generate many measurements for I for different rays through the patient. To this end the X-ray tube and the detector are rotated around the patient (see Figure 2.1) generating measurements for rays with

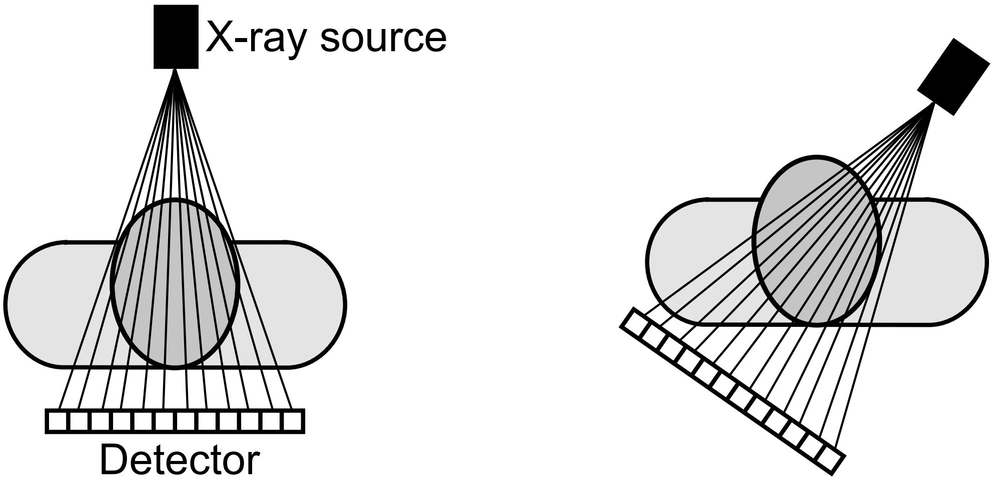


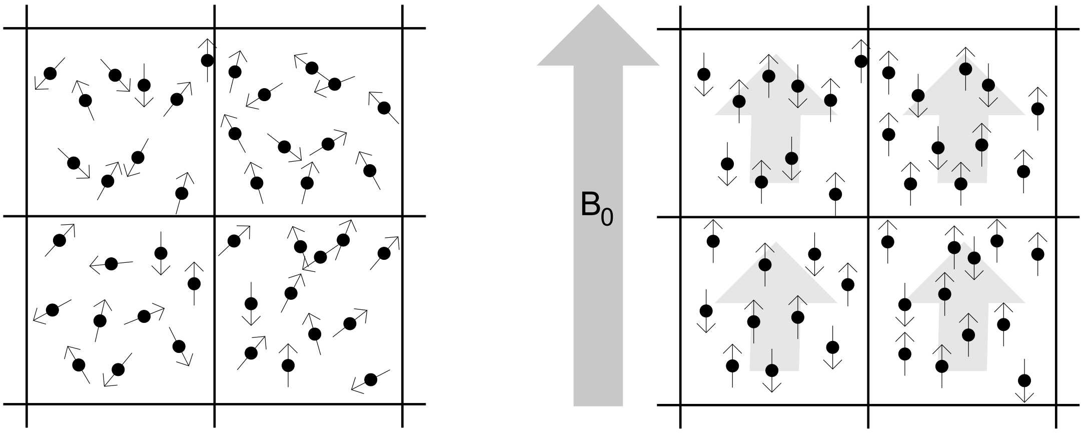
Figure 2.1: Schematic of a cone-beam CT. A source projects X-ray radiation through the patient onto a detector. The detector captures the attenuated radiation. The procedure is repeated from different angles to generate sufficient data for a 3-D reconstruction.

different angles and origins. From these measurements the values for the attenuation coefficients of μ can be calculated by methods like filtered back projection or iterative reconstruction. CT images therefore reflect how much the X-ray radiation is attenuated by the materials in the subject. It, therefore, has excellent imaging of bones, but a relatively weak soft tissue contrast. For some applications like angiography, a contrast agent with high attenuation coefficients is administered to the patient to make blood vessels visible. Contrast agents also allow for studying functional properties like blood flow or perfusion.

2.2 Magnetic Resonance Imaging

Magnetic resonance (MR) imaging is based on a completely different principle than CT. Where CT is based on X-rays, a very high energy radiation, MR works with magnetic fields and radio waves, which do not have any harmful effects on humans. The effects of nuclear magnetic resonance were first studied by Felix Bloch and Edward Purcell, for which they won the Nobel Prize in 1952. The use of this phenomenon as imaging modality is due to Paul Lauterbur and Sir Peter Mansfield, who developed the techniques for spatial localization necessary for imaging. The material presented in this section is based on [Horn 96].

MR imaging is based on the property of nuclear spin. Each electron, neutron and positron has the fundamental property called spin, which describes the continuous rotation of the particle around an axis. Normally, the spin of the particles in a material is random, such that if all the spins are summed up the net spin is zero. When placed in a strong magnetic field, usually denoted B_0, the spins align with or opposed to the field (see Figure 2.2). Statistically, however, more spins will align with the magnetic field than opposed to it. If a group of particles with spin is therefore treated as a macroscopic entity called a spin packet, its net spin is aligned with the



(a) Free spins

(b) Spins in magnetic field

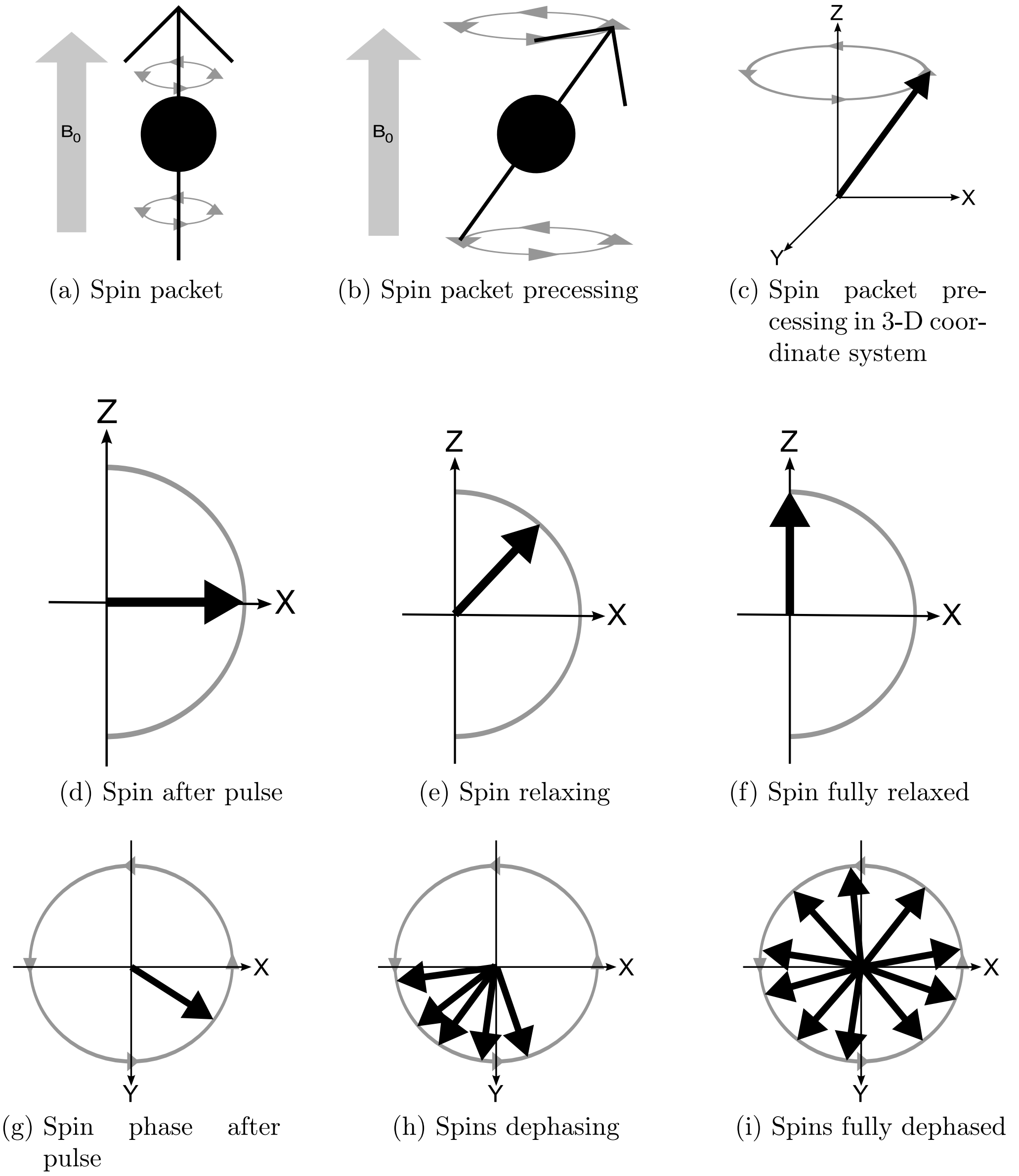
Figure 2.2: (a) shows the unorganized spins at rest. The net magnetization in the medium is 0 as the spins are oriented completely at random and cancel each other out. In (b) the spins are under the influence of a strong magnetic field B_0. In a strong magnetic field the spins align with or opposed to the magnetic field. Statistically more spins align with the magnetic fields than opposed to it. A group of spins (boxes), called a spin packet, therefore has a net spin (sum over all spins in a box) aligned with the magnetic field (gray arrows).

magnetic field. Thus aligned, the particles with spin also have the ability to absorb radio waves of a specific frequency. The frequency which is called Larmor frequency depends on the strength of the magnetic field B_0 and the chemical compound the particle is a part of. Hydrogen has the Larmor frequency of

$$\nu = B_0 \cdot 42.58 \text{ MHz/T}. \tag{2.2}$$

If a spin packet absorbs the energy of a radio wave with this frequency it is knocked out of its alignment with the magnetic field. Due to its spin it starts to precess around the axis defined by the direction of the magnetic field. Over time the spin will return to its alignment with the magnetic field while dispersing the energy it has taken in by emitting radio waves at its Larmor frequency. These radio waves can be detected by a receive coil and are ultimately the base measurement needed for constructing magnetic resonance images. Two properties of the spin returning to its original aligned state are the quantities that determine what an MR image shows. These properties are the T1 and the T2 relaxation time. The T1 time tells how long it takes a particular spin packet to return to its alignment with the magnetic field. The second property, the T2 relaxation time, results from the spins packet not being composed of completely homogeneous spins. All the spins in a spin packet interact and are subject to a very slightly different magnetic field. Once the spins in the spin packet start to precess, they will do so at very slightly differing frequencies, prompting them to go out of phase over time. The dephasing speed of the spins is measured by the T2 relaxation time. For an illustration of the T1 and T2 relaxation times refer to Figure 2.3.

The T1 and T2 times depend on many factors like the chemical composition of



(a) Spin packet

(b) Spin packet precessing

(c) Spin packet precessing in 3-D coordinate system

(d) Spin after pulse

(e) Spin relaxing

(f) Spin fully relaxed

(g) Spin phase after pulse

(h) Spins dephasing

(i) Spins fully dephased

Figure 2.3: Images (a)-(i) show the behavior of spin packets in the presence of a strong magnetic field B_0 aligned with the z-axis.
Top row: (a) Spin packet at rest rotating around its axis aligned with the magnetic field. (b) Spin packet that has been knocked out of the alignment with the vector field. The spin axis is precessing around the direction of the magnetic field B_0. (c) Spin precessing in a 3-D coordinate system, with the magnetic field B_0 aligned with the z-axis.
Middle row: Illustration of T1 decay. Images show xz-coordinate frame of (c) rotating around the z-axis such that it is always aligned with the spin. (d) The spin after a 90° pulse, (e) over time the spin realigns (f) with the magnetic field. During realignment the spin packet has to disperse energy in the form of a radio wave.
Bottom row: Illustration of T2 decay. Images show the precessing spins of a spin packet in the xy-plane of (c). (g) After an initial 90° pulse all spins of a spin packet are aligned at the same phase. (h) Over time the spins dephase, (i) until they are distributed randomly and uniformly.

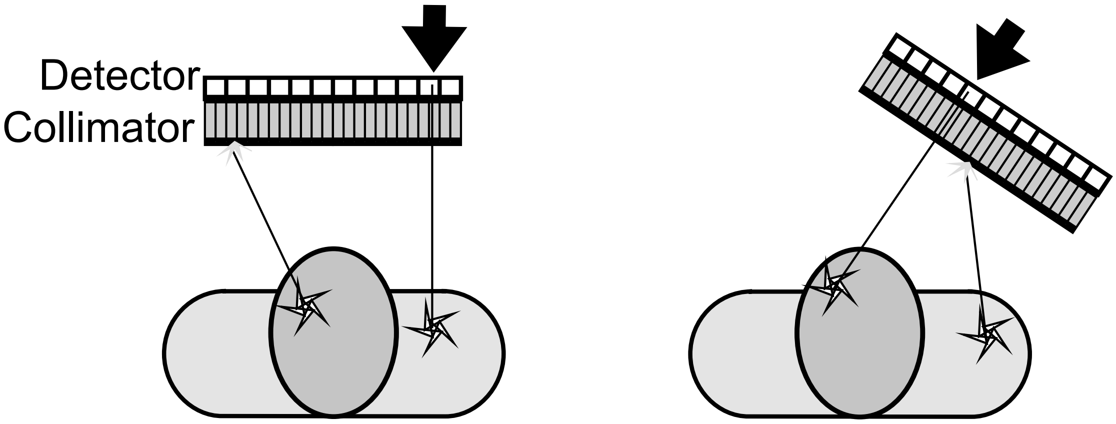


Figure 2.4: SPECT detector with collimator (gray pattern). The collimator ensures that only rays parallel to the sheets of the collimator can reach the detector. In both image examples two events (star) are shown, one which successfully projects a ray onto the detector (black arrow) and one that gets shielded by the collimator.

tissue or the mobility of the particles carrying the spin. They can therefore differ very much for different tissue types, leading to a good soft tissue contrast in MR images. Additionally, there are a lot of choices in how these times are measured and combined to form the actual image, yielding many different looking imaging sequences that can be tailored to fit the application at hand. As discussed above the radio frequency used for the pulses can only influence particles that resonate with that particular frequency. In all practical medical applications this element is Hydrogen, which is a part of any human tissue. However, there is no Hydrogen in bones, which, therefore, give almost no signal and show up pretty similar to air in MR. Accordingly, bones are mostly distinguished by the tissues surrounding them.

2.3 Single Proton Emission Computed Tomography (SPECT)

In contrast to CT and MR the main focus of nuclear medicine imaging modalities is not the imaging of the patient physiology, but rather the imaging of functions in the body. This is achieved by injecting the patient with a tracer substance that contains ligands tailored to take part in certain body functions. For example a glucose analog substance will take part in the body's metabolism and will thus concentrate in regions with a lot of glucose consumption. Other ligands are engineered to bind to certain tumor specific features [Doss 08]. The localization of the tracer inside the body is realized through the radioactive decay of a radionuclide that is chemically bound to the ligand. In single proton emission computed tomography (SPECT) this radionuclide (e. g. technetium-99m, iodine-123 etc.) undergoes a gamma decay, meaning that when the radionuclide decays it emits a gamma ray. This radiation can be detected by a gamma camera.

For the 3-D reconstruction it is necessary to detect not only that a decay has happened but also at least in which direction from the detector it happened. To facilitate this, the detector is equipped with a collimator. A collimator is essentially a block of lead with holes. Only those rays that are aligned with the holes in the collimator are thus able to reach the detector behind it. This way only rays with a clearly defined orientation are detected (compare Figure 2.4). A disadvantage of this approach is that a lot of information, i. e. rays not aligned with the collimator are completely ignored.

The radiation observed at the detector coming from a certain direction is dependent on the amount of tracer in the region and the amount of attenuation the radiation undergoes while traveling through the body. The attenuation observed in this context is related to the attenuation measured in CT, but not exactly the same due to the different energy levels of the radiation. Mathematically the measurement at the detector is thus the result of the attenuation law (2.1). However, in SPECT the quantity we want to reconstruct is I_0, the intensity at the point of origin. The attenuation μ is just a side-effect that should not show up in the reconstructed image. It is therefore necessary to have a μ-map in order to do good reconstruction of the tracer concentration in the patient. One way to acquire this map is to make a tomographic reconstruction with an exterior radiation source, much like this is done in CT. These days, most SPECT (and PET) scanners are hybrid scanners that are integrated with a CT. Although the energy level of the radiation used in CT is not exactly the same as that emitted by the tracer, it can still be used as a good approximation to the needed attenuation map.

To sum up, SPECT has the ability to produce images that visualize processes in the patient like tumor growth, perfusion or metabolism. It does in general not visualize the patient anatomy very well and the image quality is, compared to CT and MR, relatively low. Also, the radioactive tracer and the radiation produced by it, is not completely harmless, such that the minimization of the tracer dose is always a concern.

2.4 Positron Emission Tomography (PET)

Positron Emission Tomography is in many ways very similar to SPECT. The main difference is the radionuclide used for the tracer, which decays not by emitting a gamma ray, but by emitting a proton (e. g. fluorine-18). After traveling a short distance the proton will eventually hit an electron, which results in an annihilation of both while emitting two photons in almost exactly opposite directions. This emission of two photons in opposing directions makes it possible to determine a line on which the event took place by simply connecting the two points at which the photons are detected. A collimator is therefore not necessary and many more events can be detected, resulting in a higher signal yield. As photons traveling in all possible directions have to be detected the PET detector is not planar, but instead a tube all around the imaged volume (see Figure 2.5).

PET is therefore rather similar to SPECT, just with an, in general, better image quality due to the higher signal yield. PET images therefore also rely on the existence of an attenuation map for accurate image reconstruction. Aside from this PET and SPECT also rely on different tracer ligands as not every ligand used in one modality

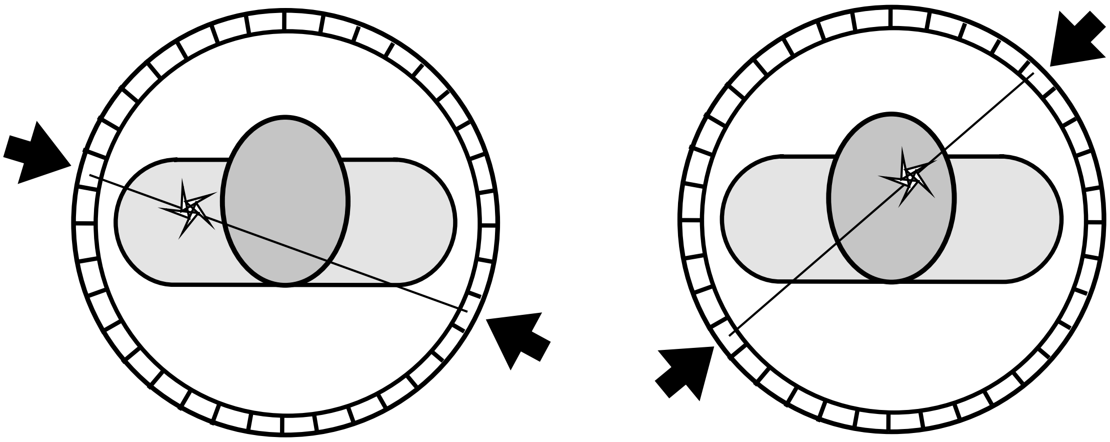


Figure 2.5: A PET detector only acknowledges simultaneously arriving photons (black arrows). The event (star) has to be somewhere on the line between the two points at which the photons were detected. No collimation is necessary.

can be easily combined with the radionuclide used in the other.

Chapter 3

Image Registration

Image registration can be employed in a variety of tasks, such as the combination of data from multiple modalities to combine functional (e. g. Section 2.3 (SPECT) and 2.4) and morphological information (e. g. Section 2.1 (CT) and 2.2 (MR)) or the monitoring of tumor growth in longitudinal studies. There are also more “abstract” applications like facilitating segmentation or classification approaches by registration with an atlas (see Section 6) or studying anatomical variations by analyzing the transforms generated by co-registering multiple datasets [Spie 09]. In some applications even datasets of different dimensionality are combined. This is for example employed in interventional applications where an intra-operative 2-D X-Ray image is registered with a pre-operative 3-D image. In this work, however, we only consider the pair-wise registrations between datasets of the same dimensionality.

The basic concept of this kind of registration is to find a transform $\boldsymbol{\Phi} : \Omega \mapsto \Omega_M$ that maps the frame of reference of the so-called moving image $M : \Omega_M \mapsto \mathbb{R}$, to the frame of reference of the fixed image $F : \Omega_F \mapsto \mathbb{R}$, such that the image content is aligned. The $\Omega_M \subset \mathbb{R}^d$ and $\Omega_F \subset \mathbb{R}^d$ denote the domains over which the moving image and the fixed image are defined, respectively. The variable d in this context specifies the dimensionality of the data, i. e. 2-D, 3-D etc.. The overlap of both images under the current transform, in the frame of reference of the fixed image, is defined as $\Omega = \{\mathbf{x} \mid \mathbf{x} \in \Omega_F \ \wedge \ \boldsymbol{\Phi}(\mathbf{x}) \in \Omega_M\}$. As such the overlap domain is actually dependent on the transform. In this work the dependency of the overlap domain Ω on the current transform $\boldsymbol{\Phi}$ will be ignored, as this would enormously complicate the mathematical formulations and also their numeric solution. To compensate, all terms used in the registration formulations presented in this work are normalized over the size $|\Omega|$ of the overlap domain, such that changes in the overlap have as little influence as possible.

A registration usually aims at minimizing a distance measure $\mathcal{D}$ that serves as the mathematical definition of the quality of the alignment, i. e.

$$\boldsymbol{\Phi}^* = \operatorname*{argmin}_{\boldsymbol{\Phi}} \mathcal{D}(F, M \circ \boldsymbol{\Phi}). \tag{3.1}$$

The choice of distance measure is highly problem dependent, though. It can be mono- or multi-modal, focus on aligning similar gray values, statistically often coinciding gray values, edges or image patterns. Similarly to the choice of the distance measure

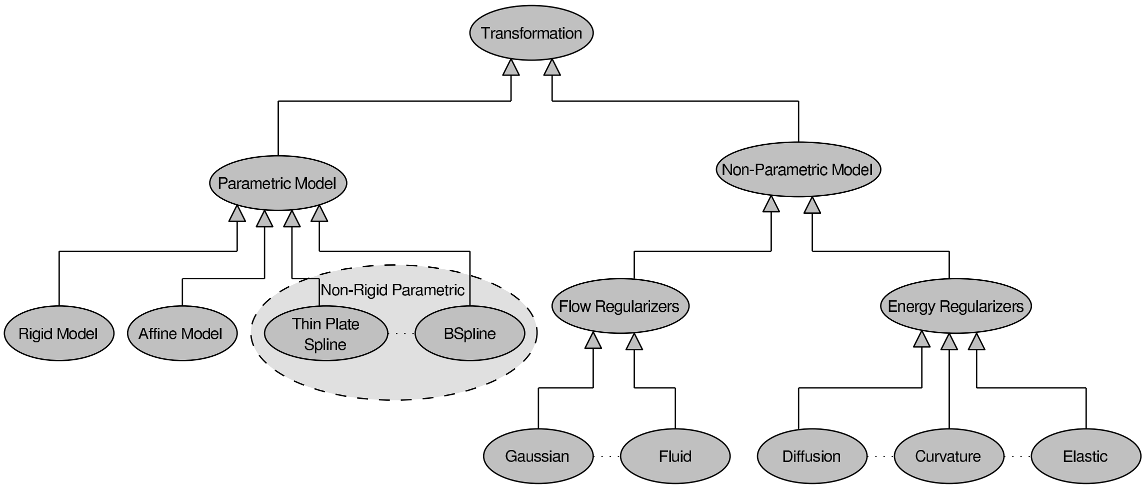


Figure 3.1: Rough categorization of different models used for the transform in popular registration approaches.

there is also a choice in how the transform $\boldsymbol{\Phi}$ should be represented. Naturally, the transform determined by the algorithm should not be completely arbitrary, but conform to some form of regularity to be useful. Usually this regularity is implemented by requiring $\boldsymbol{\Phi}$ to be locally or globally smooth in some sense. Such a restriction can be imposed directly through the way the transform is modeled or through additional regularization terms incorporated into the minimization problem.

The next section is intended to give a rough overview over different registration methods and how they model, constrain and optimize the transform. For further general overviews of different registration methods please refer to [Mode 04, Clar 06]. The remainder of this work will then concentrate on a non-rigid, non-parametric, energy regularized registration technique (compare Section 3.1.3). The components of such an algorithm, namely the distance measure and the regularizer are introduced in Section 3.3 and Section 3.4 respectively, in terms of the mathematical framework introduced in Section 3.2.

3.1 Registration Methods

In order to fulfill the varying requirements of clinical practice, namely robustness with respect to noise and the amount of motion compensated, the accuracy of the achieved result and the computation speed, a multitude of different approaches for image registration have been developed over the years. The different ways to constrain the transform lead to algorithms that differ largely in the number of unknowns that have to be optimized. As a general rule of thumb, more degrees of freedom in a transform model, result in a larger degree of deformations that can be compensated, but also to a reduced robustness with respect to noise and also a more difficult optimization problem. Figure 3.1 provides a schematic overview of the commonly employed registration methods, from the rigid registration with the least degrees of freedom to the non-parametric registration formulation that allows the specification of a transform separately for every pixel in the image domain. This section will briefly outline how most of these methods work.

3.1.1 Rigid and Affine Registration

In rigid registration only the position of the moving image M is changed, but it is not deformed in any way i. e. the transform $\boldsymbol{\Phi}$ is restricted to rotation and translation.

$$\boldsymbol{\Phi}_{\mathbf{a}}(\mathbf{x}) := \mathbf{A}_{\mathbf{a}}\mathbf{x} + \mathbf{t}_{\mathbf{a}} \tag{3.2}$$

The rigid transform $\boldsymbol{\Phi}_{\mathbf{a}}$ is defined through the parameter vector $\mathbf{a}$, which, in the case of a 3-D transform, contains three parameters to define the rotation and three parameters defining the translation along the x-,y- and z-axes. There are several possibilities how the rotation $\mathbf{A}_a$ is defined through the three rotational parameters for instance by a versor (unit quaternion; 3-D only) [Hart 04] or Euler Angles [Hart 04]. The optimization problem (3.1) thus becomes

$$\mathbf{a}^* = \underset{\mathbf{a}}{\operatorname{argmin}}\, \mathcal{D}(F, M \circ \boldsymbol{\Phi}_{\mathbf{a}}). \tag{3.3}$$

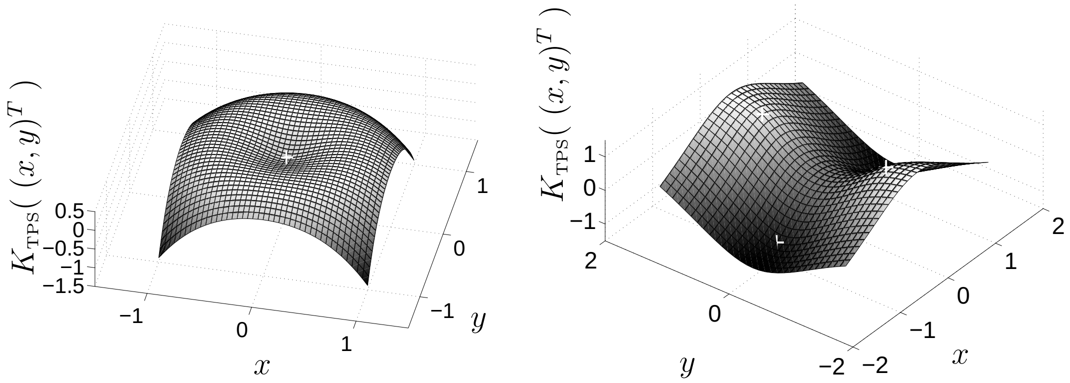
Dealing with a parametric transform simplifies the optimization considerably as the four parameters in 2-D or six parameters in 3-D needed to define such a transform represent a rather manageable search space.

Another interesting aspect is the validation of the registration result. In practical medical applications there is always the problem that the result is hard to check for its correctness. It is therefore a beneficial property of the rigid transform that only the position of the image is changed, but the image itself is not deformed. This way, at least the image itself stays trustworthy. As a downside, there are relatively few applications, like the registration of the skull, in which a genuinely rigid transform is observed. In order to account for that it is possible to relax the rigidity constraint somewhat, by allowing the matrix $\mathbf{A}_{\mathbf{a}}$ to include scalings. Adding shears ultimately leads to a completely affine transform.

As the number of parameters for this transform model is rather low (in 3-D: 6 for rigid, 9 for rigid and scaling, 12 for affine) and therefore require a relatively small search space, it is well suited to a gradient-free optimization. Calculating good gradients can be quite a challenge, especially for multi-modal distance measures. Some of the problems encountered in this context are the susceptibility of any gradient operation to noise, but also less well known effects arising from the problem discretization, as for example gridding artifacts [Plui 00]. Gradient based optimization schemes, while often giving the fastest convergence speed, can therefore be detrimental to the robustness of the algorithm. This robustness combined with the relative trust that can be placed in the resulting image $M \circ \boldsymbol{\Phi}_{\mathbf{a}}$ makes rigid registration the most widespread registration algorithm used in clinical practice.

3.1.2 Parametric Non-rigid Registration

In non-rigid registration the image content is allowed to deform to some extent to allow a better match. Still, the resulting transform should be locally smooth and



(a) TPS kernel function

(b) Interpolating TPS surface

Figure 3.2: (a) shows the TPS kernel. It is dented at the center and passes through $0 = K_{\text{TPS}}(\ (0,0)^T\ )$. It is also 0 on a circle with a radius of $r = 1$ from the origin i. e. $K_{\text{TPS}}(\mathbf{x}) = 0 \quad \forall \|\mathbf{x}\|_2 = 1$. (b) shows a sample application of a TPS interpolating the points $(1, 0)$, $(-1, 0)$, $(0, 1)$, $(0, -1)$, marked by the white crosses, with values of $1, -1, 1, -1$ respectively.

disallow ridges and folds. This can be enforced by representing the deformation as an inherently smooth, parametric function. Usually these functions are composed of kernel functions that are controlled by a set of parameters $\mathbf{a}$. The optimization problem for a parametric non-rigid registration is therefore the same as (3.3). The most common examples are BSplines and thin-plate splines. Although the regularity of the transform is implicitly imposed by the model, not every model can guarantee that unwanted properties of the transform cannot occur. For example it is perfectly possible to get folds in a BSpline transform if no additional measures are taken to prevent this. In practice this is often resolved by constraining the step size or search area in the optimization, such that the control points cannot be moved far enough to allow such results or by imposing additional constraints on the control points.

Thin-Plate Splines

The thin-plate splines (TPS) are based on the mathematical work of [Wahb 90] and [Duch 76]. They were introduced into the context of image registration in [Book 89] and have been used in many publications since (see e. g. [Rohr 01](approximating TPS), [Evan 91](application in 3-D)). TPS and other radial basis-functions are most often used in conjunction with landmarks i. e. known point correspondences $\boldsymbol{\Phi}(\mathbf{x}_{Fi}) = \mathbf{x}_{Mi}$, where $\mathbf{x}_{Fi} \in \Omega_F$ and $\mathbf{x}_{Mi} \in \Omega_M$ denote the corresponding coordinates of the landmark in the moving and fixed image frame of reference respectively. Intuitively a TPS describes the elevation of a metal plate that is deformed by being fixed to a certain number of points. The TPS is defined as the function that minimizes the bending energy of the plate while interpolating those fixed points. In 2-D the TPS kernel function (see also Figure 3.2a) is defined as

$$K_{\text{TPS}}(\mathbf{x}) := \|\mathbf{x}\|_2^2 \log \|\mathbf{x}\|_2, \tag{3.4}$$

where $\|\cdot\|_2$ is the L^2 norm. K_{TPS} is centered around the origin and passes through $0 = K_{\text{TPS}}(\ (0,0)^T\ )$. The complete spline is made up of an affine transform and a set of kernel functions centered around the fixed points. The affine transform and a set of weights for the kernel functions are chosen such that the landmark points $\mathbf{x}_{Mi}$ are interpolated. For an example of a TPS surface see Figure 3.2b. To define a transform by a TPS, the deformation of the plate is interpreted as a displacement in each dimension respectively i. e. one TPS is needed per dimension. The affine transform is lifted straightforwardly to d dimensions (d-D). A transform defined by a TPS is thus written as

$$\boldsymbol{\Phi}_{\text{TPS}}(\mathbf{x}, \mathbf{a}) := \mathbf{A}_{\mathbf{a}}\mathbf{x} + \mathbf{t}_{\mathbf{a}} + \sum_{i=1}^{n} \mathbf{b}_{\mathbf{a},i} K_{\text{TPS}}(\mathbf{x} - \mathbf{x}_{Fi}), \tag{3.5}$$

where $\mathbf{a}$ is the set of parameters defining the TPS by providing the control points $\mathbf{x}_{Fi}$ with $i = 1, \ldots, n$, the weights of the kernel functions $\mathbf{b}_{\mathbf{a},i}$ and the affine components of the transform $\mathbf{A}_{\mathbf{a}}$ and $\mathbf{t}_{\mathbf{a}}$. The definition of the transform through the TPS makes it possible to guarantee a one-to-one match of the landmarks with a smoothly interpolated deformation field everywhere else. It also allows for a closed form solution of the matching problem. One of the disadvantages of this model is that each kernel, and therefore each landmark, has a global influence. Thus moving a landmark influences the transform over the whole domain and requires a recomputation of all parameters.

BSplines

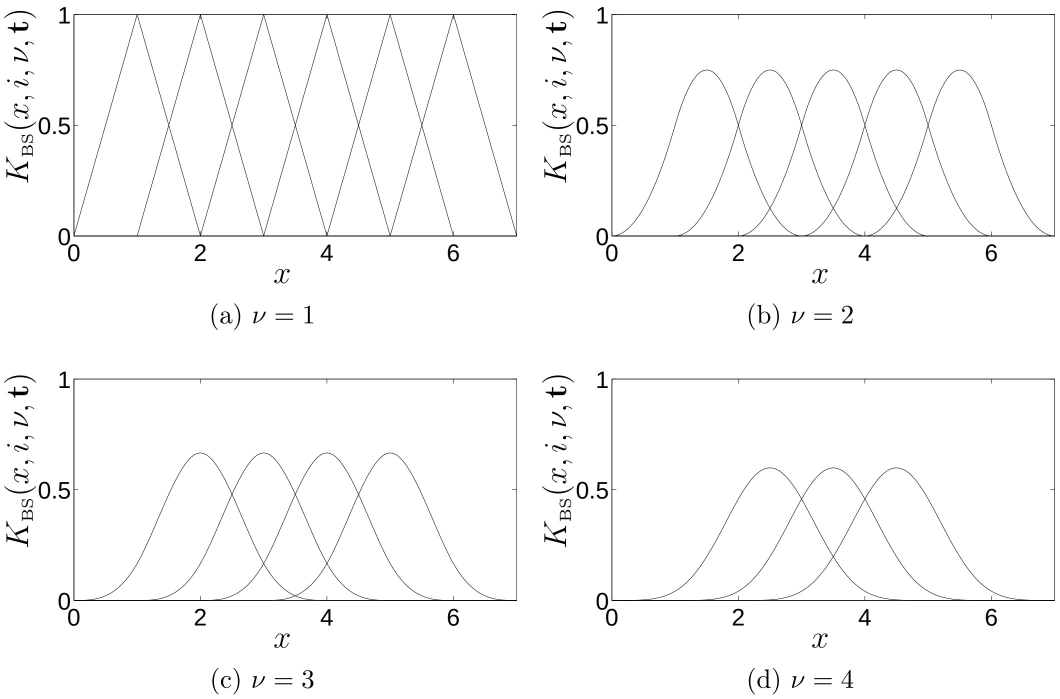
BSplines are piecewise polynomial functions that are often used in image processing to represent smooth data [Lee 97]. In image registration they have been used for instance by Rückert et al. [Ruec 99] and Rohlfing et al.. [Rohl 00] in non-rigid registration. These methods employ BSplines to represent the deformation. Depending on the number of control points used to define a BSpline it can be represented with a relatively low amount of parameters and is inherently smooth. The more control points are used, the more local the inherent smoothness becomes and the more parameters are needed.

In 1-D BSplines are composed of a set of kernel functions that are stitched at given points in their parameter domain. These points are defined by the so-called knot vector $\mathbf{t} \in \mathbb{R}^m$. Usually the knot vector is set uniformly to $\mathbf{t} = (0, 1, 2, \ldots)$. Using this, the individual 1-D BSpline kernel functions are defined as

$$\begin{aligned} K_{\text{BS}}(x, i, \nu, \mathbf{t}) :=& \frac{x - t_i}{t_{i+\nu} - t_i} K_{\text{BS}}(x, i, \nu - 1, \mathbf{t}) \\ &+ \frac{t_{i+\nu+1} - x}{t_{i+\nu+1} - t_{i+1}} K_{\text{BS}}(x, i + 1, \nu - 1, \mathbf{t}) \end{aligned} \tag{3.6}$$

$$K_{\text{BS}}(x, i, 0, \mathbf{t}) := \begin{cases} 1 & \text{if } x \in [t_i, t_{i+1}] \\ 0 & \text{else} \end{cases}, \tag{3.7}$$

In Figure 3.3 the 1-D BSpline kernels for $\nu = 1, \ldots, 4$ are shown. In higher dimensions



(a) $\nu = 1$

(b) $\nu = 2$

(c) $\nu = 3$

(d) $\nu = 4$

Figure 3.3: BSpline kernels for polynomial degree $\nu = 1, \ldots, 4$ and uniform knot vector.

the BSpline basis functions are evaluated for each dimension separately and multiplied to get the final d-dimensional kernel function. This method of applying BSplines to higher dimensions is called a tensor product BSpline. Accordingly a separate knot vector $\mathbf{t}_i \in \mathbb{R}^{m_i} \quad i = 1, \ldots, d$ is needed for the definition of the intervals in each dimension. The BSpline is controlled by a set of control points $\mathbf{P}$ that are arranged on a regular grid. In 2-D, for example, with knot vectors $\mathbf{t}_1 \in \mathbb{R}^{m_1}$ and $\mathbf{t}_2 \in \mathbb{R}^{m_2}$ there would be a control point grid with $(m_1 - \nu) \times (m_2 - \nu)$ control points $\mathbf{P} = (\mathbf{p}_{1,1}, \ldots, \mathbf{p}_{m_1-\nu, m_2-\nu})$. The tensor product BSpline in 2-D is thus written as

$$\begin{aligned}\boldsymbol{\Phi}_{\mathrm{BS}}(\mathbf{x}, \mathbf{a}) &:= \boldsymbol{\Phi}_{\mathrm{BS}}(\mathbf{x}, \nu, \mathbf{P}, \mathbf{t}_1, \mathbf{t}_2) \\ &= \sum_{i=1}^{m_1-\nu} \sum_{j=1}^{m_2-\nu} \mathbf{p}_{i,j} K_{\mathrm{BS}}(x_1, i, \nu, \mathbf{t}_1) K_{\mathrm{BS}}(x_2, j, \nu, \mathbf{t}_2). \end{aligned} \tag{3.8}$$

As the BSpline is stitched together from kernel functions with a limited support a change of a control point also only has a local influence. This is advantageous during the optimization of the registration transform, as the change of a single control point does not necessarily mean that the complete distance measure has to be re-evaluated. A drawback of the stitching of the kernel functions is that the BSpline is only differentiable up to one less than the polynomial degree of the basis functions. Thus for a polynomial degree ν, a BSpline is $C^{\nu-1}$ continuous. Therefore, if one is interested in properties derived from derivatives of the deformation field or intends to use optimization schemes based on derivatives, BSplines with a high enough polynomial degree have to be chosen.

Additionally, registration models employing BSplines often add an affine transform $\boldsymbol{\Phi}_{\text{BS}}$, as the BSpline alone cannot accurately model rotations, for instance. Except for BSplines of degree $\nu = 1$, which is just a polygon connecting the control points, the control points are not interpolated. The smoothness of the model is controlled by the spacing and number of the control points and by the polynomial degree ν. Theoretically there is no constraint on the placement of the control points, and therefore also no guarantee that folds may not occur. Practically, this can be controlled through the optimization algorithm by constraining the step size in a suitable manner or by imposing additional constraints on the control points. For the optimization of the transform gradient-free, as well as different gradient based optimization methods [Kabu 04, Ruec 99] have been used. A review of several optimization methods is presented in [Klei 07].

3.1.3 Non-Parametric Non-Rigid Registration

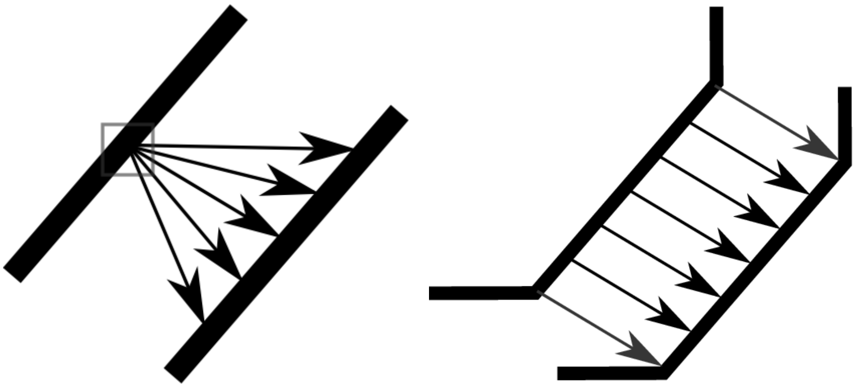
The approach that this work is focused on is the non-parametric, non-rigid registration, which was introduced by Hermosillo et al. [Herm 02b]. The formulation of the algorithm in this work is also heavily based on the work of Modersitzki [Mode 04]. This approach allows deformations that are not constrained by an explicit model, which theoretically makes arbitrary transforms possible. As the location of each pixel $\mathbf{x}$ can be moved by an individual offset $\boldsymbol{u}(\mathbf{x})$ it is helpful to represent the transform in terms of these offsets.

$$\boldsymbol{\Phi}(\mathbf{x}) := \mathbf{x} - \boldsymbol{u}(\mathbf{x}). \tag{3.9}$$

This formulation with an “offset” function $\boldsymbol{u}$ instead of the transform $\boldsymbol{\Phi}$ has the advantage that $\boldsymbol{u}$ directly represents the intuitive notion of a deformation field, which is an offset onto the untransformed index space. It is formulated as a negative offset $\mathbf{x} - \boldsymbol{u}(\mathbf{x})$ as a deformation field is supposed to point from the untransformed index position in the moving image M to the target position in the new frame of reference, i. e. $\mathbf{x} \in \Omega_M \mapsto \mathbf{x} + \boldsymbol{u}(\mathbf{x})$. But as the distance measure is evaluated in the frame of reference of the fixed image $\mathbf{x} \in \Omega_F \mapsto \mathbf{x} - \boldsymbol{u}(\mathbf{x})$ has to be evaluated in order to apply the transform.

In the practical implementation, with discreetly represented vector fields and images, the deformation is also represented in the frame of reference of the fixed image for practical reasons: For the computation of $M_{\boldsymbol{u}}$ the moving image has to be resampled deformed and resampled to the discrete grid on which the fixed image is represented. The deformation locally expands and compresses the vector field which leads to over and undersampling. This is difficult to account for when interpolating the deformed image back onto the regular grid. It is much simpler to represent the deformation in the frame of reference of the fixed image, as this way, one can simply look up the corresponding image intensity for every pixel position $M_{\boldsymbol{u}}$.

In general, if the transform is not further constrained the mathematical problem is not well defined. Consider for example a moving edge (see Figure 3.4). With no additional information it is not possible to determine for a single pixel on the source edge, where it has to be mapped on the target edge. This is known as the



(a) Aperture problem, local (b) Aperture problem, global

Figure 3.4: Local view of a moving edge in an image. (a) For a single pixel it is not possible to determine a unique match on the target edge. (b) A more global view reveals neighboring corners for which the match is clear. By requiring the vector field to be smooth the match for all points in between is also made unique.

aperture problem in the optical flow community (see e. g. [Beau 95]). To overcome this limitation additional information has to be considered for the calculation of the movement of an individual pixel. This is done by adding a so-called regularizer or smoother. It adds information about the deformation of the pixel neighborhood by requiring that the calculated vector field is locally smooth. This way undesirable transforms that include folds or ridges are discouraged. A regularizer can be added as an energy term $\mathcal{R}$ to the optimization problem (3.1):

$$\begin{aligned} \boldsymbol{u}^* &= \underset{\boldsymbol{u}}{\operatorname{argmin}}\, \mathcal{E}(F, M, \boldsymbol{u}) \\ &:= \underset{\boldsymbol{u}}{\operatorname{argmin}}\, \mathcal{D}(F, M_{\boldsymbol{u}}) + \alpha \mathcal{R}(\boldsymbol{u}), \end{aligned} \tag{3.10}$$

$$\text{where} \quad M_{\boldsymbol{u}}(\mathbf{x}) = M(\mathbf{x} - \boldsymbol{u}(\mathbf{x})). \tag{3.11}$$

The weighting parameter $\alpha \in \mathbb{R}, \alpha > 0$ determines how strictly the regularization term has to be adhered to. This energy regularization is a kind of Tikhonov regularization [Clar 06, Tikh 77]. Usually, when such a regularization is used it is desirable to ultimately get rid of the regularization term again if a convergence towards a good result is achieved, as it was not a part of the original problem statement. A standard approach is, for instance, to iteratively decrease α during the optimization. In medical image registration, however, it can be argued that the regularizer represents physical properties of the tissue being deformed and therefore should not be eliminated.

A different way to incorporate the necessary regularization is related to iterative Tikhonov regularization [Clar 06]. Not the energy is regularized, but rather the search direction (usually the gradient) in the optimization. For example gradient flow methods [Strz 04, Dros 05] are basically gradient descent optimization schemes for (3.1) that apply a smoothing operator to the calculated gradient in each step. This smoothing operator can be a Gaussian smoothing or a more complex operation, related to the energy regularizers presented in Section 3.4. This way a "smooth path" to the final result is generated. Overall this also results in smooth transforms, although the result might be a bit more complicated to predict as in the case of energy regularization. In some cases, also both approaches are combined by using a gradient flow optimizer to optimize an energy regularized registration term. Another scheme that regularizes the change in the deformation field is the fluid registration [Bro 96], It defines a partial differential equation governing the "time dependent" behavior of the deformation field. Time here is introduced as an artificial parameter, considering the moving and the fixed image as two instances of the same object at different points in time. As the overall deformation field is not further constrained it is able to match even rather unrelated shapes if necessary. Whether this is a desirable property depends, as usual, on the application in question.

3.2 Mathematical Framework

For the remainder of this work we will focus on a non-rigid, non-parametric, energy regularized registration formulation, as outlined in Section 3.1.3. The components of the algorithm are, therefore, at least a distance measure $\mathcal{D}$ and a regularizer $\mathcal{R}$. In this section we will discuss the mathematical tools and notation needed for the introduction of the components of the registration algorithm. With these tools it is possible to introduce the terms in the continuous, formulate their continuous derivative and the corresponding discretizations necessary for an implementation.

3.2.1 Notation

The following notations are used to help the reader to better distinguish whether a formulation is presented in the continuous or discrete and if vector or scalar valued quantities are treated. As usual, matrices and vectors are denoted in bold notation, such as $\mathbf{A}$ and $\mathbf{a}$ respectively. Many of the functions used like $\boldsymbol{u}$ are also vector valued, as they describe an offset in $\mathbb{R}^d$. To better distinguish between these continuous, vector valued functions, they are denoted in bold, italic notation as $\boldsymbol{u}$, while their discretized counterparts are denoted in standard vector notation as $\mathbf{u}$. We also differentiate different norms by subscripts. If not otherwise indicated $|\cdot|$ denotes the scalar absolute value and $||\cdot||$ the Euclidean or L^2 norm.

3.2.2 Variational Calculus

The usual way to solve continuous optimization problems is to identify extremal points by determining the roots of the derivative of the function that has to be optimized. In the case of equation (3.10), however, one has to deal with an optimization with respect to a function $\boldsymbol{u}$. As a function cannot be varied in the same way as a scalar variable, it is necessary to turn to the calculus of variations [Bron 99] to find a minimizer $\boldsymbol{u}^*$. Let the space of functions over which the minimization is performed be a Hilbert space $\mathcal{U} : \Omega \mapsto \Omega$ of functions over the domain $\Omega \subseteq \mathbb{R}^d$ over which both images F and M are defined. The Hilbert space is defined by function addition, subtraction, scalar multiplication and an inner product. For the inner product on $\mathcal{U}$, which also induces a norm $\|\boldsymbol{u}\|_{\mathcal{U}}$, the following definition is chosen.

$$\forall \boldsymbol{u}, \boldsymbol{v} \in \mathcal{U} \qquad \langle \boldsymbol{u}, \boldsymbol{v} \rangle_{\mathcal{U}} := \frac{1}{|\Omega|} \int_{\Omega} \boldsymbol{u}(\mathrm{x})^T \, \boldsymbol{v}(\mathrm{x}) \; \mathsf{dx} \qquad \|\boldsymbol{u}\|_{\mathcal{U}} := \langle \boldsymbol{u}, \boldsymbol{u} \rangle_{\mathcal{U}} \tag{3.12}$$

Note that this definition of the inner product is normalized over the size of the domain $|\Omega|$. This definition makes the notation of accordingly normalized distance measures and regularizers easier later on, as the normalization term $\frac{1}{\Omega}$ will be already incorporated into the inner product. With respect to a test-function $\boldsymbol{\eta} \in \mathcal{U}$ the Gâteaux derivative [Bron 99] of (3.10) around $\boldsymbol{u}$ is defined as

$$\begin{aligned} \mathsf{d}\mathcal{E}(F, M, \boldsymbol{u}; \boldsymbol{\eta}) &:= \lim_{\epsilon \to 0} \frac{\mathcal{E}(F, M, \boldsymbol{u} + \epsilon \boldsymbol{\eta}) - \mathcal{E}(F, M, \boldsymbol{u})}{\epsilon} \\ &= \left. \frac{\mathsf{d}\mathcal{E}(F, M, \boldsymbol{u} + \epsilon \boldsymbol{\eta})}{\mathsf{d}\epsilon} \right|_{\epsilon = 0} . \end{aligned} \tag{3.13}$$

For the existence of a minimizer for (3.10), it is necessary that the Gâteaux derivative vanishes for all possible choices of the test function $\boldsymbol{\eta}$, i. e.

$$\mathsf{d}\mathcal{E}(F, M, \boldsymbol{u}^*; \boldsymbol{\eta}) = 0 \quad \forall \boldsymbol{\eta} \in \mathcal{U}. \tag{3.14}$$

This is known as the Euler-Lagrange equation. If this equation holds the Gâteaux derivative has reached an extremum. This is very similar to a directional derivative, just with a test function instead of a direction vector. Furthermore, similarly to the definition of an extremum of a function through directional derivatives, we can use the inner product of $\mathcal{U}$ to define for the Euler-Lagrange equation

$$\langle \nabla_{\boldsymbol{u}} \mathcal{E}(F, M, \boldsymbol{u}), \boldsymbol{\eta} \rangle_{\mathcal{U}} := \left. \frac{\mathsf{d}\mathcal{E}(F, M, \boldsymbol{u} + \epsilon \boldsymbol{\eta})}{\mathsf{d}\epsilon} \right|_{\epsilon = 0} = 0 \qquad \forall \boldsymbol{\eta} \in \mathcal{U}. \tag{3.15}$$

As this has to hold for all possible test functions $\boldsymbol{\eta}$, the Euler-Lagrange equations are equivalent to $\nabla_{\boldsymbol{u}} \mathcal{E}(F, M, \boldsymbol{u}^*) = 0$. It is imperative to keep in mind that $\nabla_{\boldsymbol{u}} \mathcal{E}(F, M, \boldsymbol{u})$ is not a derivative in the common sense, but just defined through the above equation. This definition is equivalent to the definition of the gradient of a function through directional derivatives. That this definition is applicable to all the terms encountered in this work is due to the chain rule being applied in the differentiation. This ensures that every term in the Gâteaux derivatives we will derive is an inner product of $\boldsymbol{\eta}$ or any of its derivatives with another term.

Strictly speaking, we would have to derive the Gâteaux derivative for the complete term of $\nabla_{\boldsymbol{u}} \mathcal{E}(F, M, \boldsymbol{u})$ from equation (3.10). However, this would result in very large equations. Instead we will derive the Gâteaux derivatives separately for the regularization (Section 3.4) and distance measure terms (Section 3.3). It is therefore important that this is done in a consistent fashion and that constants are not eliminated, as this would then have to be done analogously in all other terms as well.

3.2.3 Discretization

As it is not possible to represent all possible choices for the continuous functions $\boldsymbol{u}$, the optimization of $\mathcal{E}(F, M, \boldsymbol{u})$ requires a discretization of all terms. The basis for the discretization is the discrete representation of the domain Ω over which all the functions and their derivatives are defined. To this end, the domain Ω is sampled at s discrete positions $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_s)^T$, where each position $\mathbf{x}_i \in \Omega$ is a d-dimensional vector specifying a discrete position. In practice, the sample positions $\mathbf{x}_i$ are chosen on a rectangular grid with uniform spacing h_j and $j = 1, \ldots, d$ in each dimension. As it is easier to represent the sampled functions and gradients as a large vector than as a matrix corresponding to $\mathbf{X}$, the discretized vector fields are arranged as

$$\begin{aligned}\mathbf{u} &= (\mathbf{u}_1^T, \ldots, \mathbf{u}_d^T)^T \\ &= (u_{1,1}, \ldots, u_{1,s}, \ldots, u_{d,1}, \ldots, u_{d,s})^T \\ &= (u_1(\mathbf{x}_1), \ldots, u_1(\mathbf{x}_s), \ldots, u_d(\mathbf{x}_1), \ldots, u_d(\mathbf{x}_s))^T\end{aligned} \quad . \tag{3.16}$$

The discrete representation $\mathbf{u}$ of the deformation function $\boldsymbol{u}$ therefore contains d sequential vectors $\mathbf{u}_i$ of size s, which define the deformation in each dimension separately. The same discretization is also employed for the gradient of the energy $\nabla_{\boldsymbol{u}}\mathcal{E}(F, M, \boldsymbol{u})$ and the terms it is composed of. The inner product (3.12) can thus be discretized as

$$\underbrace{\langle \boldsymbol{\eta}, \boldsymbol{u} \rangle_{\mathcal{U}}}_{\text{continuous}} \approx \underbrace{\langle \boldsymbol{\upeta}, \mathbf{u} \rangle_{\mathcal{U}}}_{\text{discrete}} = \frac{1}{s} \boldsymbol{\upeta}^T \mathbf{u}. \tag{3.17}$$

The discretized Gâteaux derivative of the energy $\mathcal{E}$ is therefore

$$\langle \boldsymbol{\eta}, \nabla_{\boldsymbol{u}}\mathcal{E}(F, M, \boldsymbol{u}) \rangle_{\mathcal{U}} \approx \langle \boldsymbol{\upeta}, \nabla_{\mathbf{u}}\mathcal{E}(F, M, \mathbf{u}) \rangle_{\mathcal{U}} = \frac{1}{s} \boldsymbol{\upeta}^T \; \nabla_{\mathbf{u}}\mathcal{E}(F, M, \mathbf{u}) = 0. \tag{3.18}$$

As the discrete gradient is the vector of all partial derivatives with respect to all parameters, it can be constructed by replacing the discrete test function $\boldsymbol{\upeta}$ by the standard basis vectors. This then constitutes real directional derivatives, defining a d-D gradient vector. The resulting gradient is similar to the one that would be obtained, if it was calculated directly on the discretized energy $\mathcal{E}(F, M, \mathbf{u})$. Note that the partial derivatives for each sample position $\mathbf{x}_i$ get shorter with the overall number of samples s. As the number of sample positions goes to infinity, the change that can be obtained at a single "discrete" position tends to zero, because the region represented by the sample position also tends to zero.

It has also been discussed whether one should "discretize and optimize" or "optimize and discretize", meaning whether the discretization should be performed before or after the gradient derivation (see e. g. [Habe 06]). One of the main differences between these approaches is that if the discretization is performed first, the calculation of the derivative can be performed with differential calculus (vector derivative), which is in general mathematically easier to handle than the derivation in the framework of the calculus of variations (function derivative). In our experience the result is in most

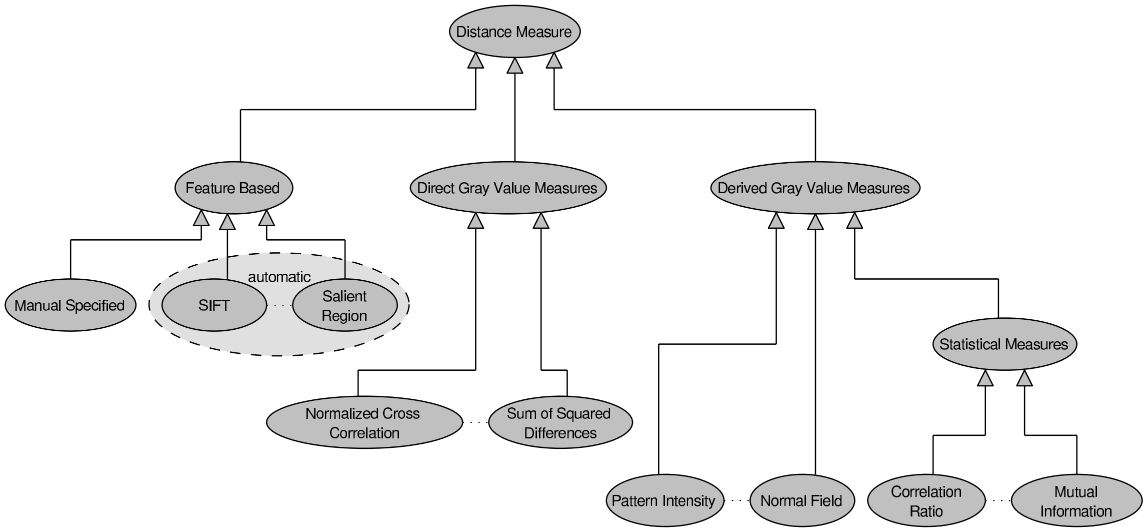


Figure 3.5: Rough categorization of different distance measures in popular registration approaches.

cases pretty much identical with the notable exception of how boundary conditions get incorporated.

With these tools at hand we can now proceed to introduce the formulations of the distance measures and regularizers. These are always introduced as continuous terms, for which the Gâteaux derivative is calculated. Finally, the continuous terms are discretized to get formulations that can be implemented in a computer program.

3.3 Distance Measures

In any registration algorithm the distance measure decides what is considered a good match. In this section we will introduce the two distance measures used in the experiments in this work. To put them into context a rough schematic overview over the different classes of distance measures is depicted in Figure 3.5. The first block shown there are feature based distance measures, which do not compare the images directly, but instead only the distance between a set of (sparse) corresponding features computed on the images. The way that the features are identified on the images ranges from manual specification, to automatic methods like differential operators (detect ridges, corners etc.), SIFT features (compare [Ke 04]) or salient region features [Huan 04, Hahn 06]. A review of several local feature descriptors applicable in this problem domain can be found in [Miko 05].

A more direct approach to compare the two images is to compare the gray values directly. The simplest variant of this approach are the mono-modal measures which assume that corresponding structures in the two images have identical intensities. The most prominent examples of mono-modal distance measures are the sum of absolute differences, the sum of squared differences and the cross-correlation. The cross-correlation already relaxes the assumption of identical image intensities to some extent. The sum of squared differences has gained some popularity as distance measure due to its inherent simplicity which makes it well understood, easy to implement and thus a good subject for experimenting with new registration methods or optimization schemes. As it is also the measure of choice for the mono-modal registrations performed in this work it is discussed in more detail in Section 3.3.1. In order to be able to better match images from multiple modalities, measures have to compare derived quantities such as image patterns (compare [Wees 97] for 2-D/3-D registration), the normal field [Dros 04, Habe 05] or image statistics. Image statistics based distance measures feature some of the most prominent multi-modal distance measures like the mutual information [Well 96, Maes 97] (MI), which is discussed in more detail in Section 3.3.2, the normalized mutual information [Stud 99] or the correlation ratio [Roch 98].

In the following sections the two distance measures used in this work are presented. We will discuss the calculation of their respective derivatives and how a discrete implementation can be realized.

3.3.1 Sum of Squared Differences

The sum of squared differences (SSD) is one of the simplest distance measures available (see e. g. [Mode 04]). It is based on the assumption that the intensities of corresponding tissue within two datasets are equal and is defined as follows:

$$\mathcal{D}_{\text{SSD}}(F, M_{\boldsymbol{u}}) := \frac{1}{|\Omega|} \int_{\Omega} \left(F(\mathbf{x}) - M_{\boldsymbol{u}}(\mathbf{x})\right)^2 \,\mathsf{dx}. \tag{3.19}$$

Note that this denotes a d-D integral. The definition presented here contains a normalization over the size of the computational domain $|\Omega|$. This makes the measure somewhat more predictable when applied to differently sized datasets. The according derivative can be calculated in the framework of the variational calculus as

$$\begin{aligned}
\mathsf{d}\mathcal{D}_{\text{SSD}}(F, M_{\boldsymbol{u}}; \boldsymbol{\eta}) &= \frac{\mathsf{d}}{\mathsf{d}\epsilon} \frac{1}{|\Omega|} \int_{\Omega} \left(F(\mathbf{x}) - M(\mathbf{x} - \boldsymbol{u}(\mathbf{x}) - \epsilon\boldsymbol{\eta}(\mathbf{x}))\right)^2 \,\mathsf{dx}\Big|_{\epsilon=0} \\
&= \frac{1}{|\Omega|} \int_{\Omega} 2\boldsymbol{\eta}(\mathbf{x})^T \left(F(\mathbf{x}) - M(\mathbf{x} - \boldsymbol{u}(\mathbf{x}))\right) \left((\nabla M)(\mathbf{x} - \boldsymbol{u}(\mathbf{x}))\right) \,\mathsf{dx} \\
&= \langle 2(F - M_{\boldsymbol{u}})\nabla M_{\boldsymbol{u}}, \boldsymbol{\eta}\rangle_{\mathcal{U}} \\
&= 0 \quad \forall \boldsymbol{\eta} \in \mathcal{U}
\end{aligned} \tag{3.20}$$

Note that $\nabla M_{\boldsymbol{u}} = (\nabla M)(\mathbf{x} - \boldsymbol{u}(\mathbf{x}))$ denotes the gradient of the untransformed moving image M evaluated at the transformed position $\mathbf{x} - \boldsymbol{u}(\mathbf{x})$. Using (3.15) we can thus define

$$\nabla_{\boldsymbol{u}} \mathcal{D}_{\text{SSD}}(F, M_{\boldsymbol{u}}) = 2(F - M_{\boldsymbol{u}})\nabla M_{\boldsymbol{u}}. \tag{3.21}$$

The discrete approximation of (3.19) and its derivative (3.21) is determined on the grid $\mathbf{X}$ by the application of the discretization of the inner product (3.17), as

$$\begin{aligned}\mathcal{D}_{\text{SSD}}(F, M_{\mathbf{u}}) &= \frac{1}{s}(F(\mathbf{x}) - M_{\mathbf{u}}(\mathbf{x}))^2 \\ &= \frac{1}{s}\sum_{i=1}^{s}(F(\mathbf{x}_i) - M_{\mathbf{u}}(\mathbf{x}_i))^2 \qquad (3.22) \\ \nabla_{\mathbf{u}}\mathcal{D}_{\text{SSD}}(F, M_{\mathbf{u}})(\mathbf{x}_i) &= \frac{2}{s}(F(\mathbf{x}_i) - M_{\mathbf{u}}(\mathbf{x}_i))\nabla M_{\mathbf{u}}(\mathbf{x}_i) \quad \forall i = 1, \ldots, s. \qquad (3.23)\end{aligned}$$

3.3.2 Mutual Information

For the multi-modal registration task, distance measures based on image intensity statistics are widely established. One of the most often used statistical distance measures is the mutual information and its variants. Use of the mutual information (MI), was introduced as distance measure for image registration by Wells et al. [Well 96] and Maes et al. [Maes 97]. The mutual information evaluates how much information is shared between both images in their current alignment. It is defined over the probability density functions (PDF) of gray values in the moving image $p_{M_{\boldsymbol{u}}}$, fixed image p_F and the joint PDF $p_{F,M_{\boldsymbol{u}}}$. The PDFs describe how likely the gray values i_M, i_F and the gray value pair $\mathbf{i} = (i_F, i_M)^T$ can be observed in their respective images under the current deformation $\boldsymbol{u}$.

If F and $M_{\boldsymbol{u}}$ were statistically independent then $p_{F,M_{\boldsymbol{u}}} = p_F \; p_{M_{\boldsymbol{u}}}$. For a good match the intensities of F and $M_{\mathbf{u}}$ should be as statistically dependent as possible. As a joint distribution of $p_{F,M_{\mathbf{u}}} = p_F \, p_{M_{\mathbf{u}}}$ would indicate a statistical independence of the two distributions, the mutual information distance measure therefore aims to maximize the Kullback-Leibler divergence (KL) between $p_{F,M_{\mathbf{u}}}$ and $p_F \, p_{M_{\mathbf{u}}}$. To be useful as a distance measure, which has to be minimized, the mutual information distance measure here is formulated with the negative KL.

$$\begin{aligned}\mathcal{D}_{\text{MI}}(F, M_{\boldsymbol{u}}) &:= -\,\text{KL}(p_{F,M_{\boldsymbol{u}}}, p_F \; p_{M_{\boldsymbol{u}}}) \\ &= -\int_{\mathbb{R}^2} p_{F,M_{\boldsymbol{u}}}(\mathsf{i}) \; \log \frac{p_{F,M_{\boldsymbol{u}}}(\mathsf{i})}{p_F(i_F) \; p_{M_{\boldsymbol{u}}}(i_M)} \; \mathsf{di} \qquad (3.24)\end{aligned}$$

The implementation and derivation of $\mathcal{D}_{\text{MI}}$ presented here is done as outlined in [Herm 02a].

Density Estimation

The first step in computing $\mathcal{D}_{\text{MI}}$ is the calculation of the joint PDF $p_{F,M_{\mathbf{u}}}$ and its marginals p_F and $p_{M_{\mathbf{u}}}$. To estimate a smooth PDF, Parzen estimation is applied. Parzen established in his work [Parz 62] that a PDF can be approximately recovered by sampling the data and smoothing the samples with an appropriate kernel function K. Let $\boldsymbol{\rho} = (\rho_1, \rho_2)^T$ denote the bandwidth parameter of the kernel function then K

is defined as

$$K_\rho(x) := \frac{1}{\rho} K\left(\frac{x}{\rho}\right) \qquad \text{for 1-D} \tag{3.25}$$

$$K_{\boldsymbol{\rho}}(\mathbf{x}) := K_{\rho_1}(x_1)\; K_{\rho_2}(x_2) \qquad \text{for 2-D,} \tag{3.26}$$

If n "joint intensity" samples $\mathbf{j}_k$ are drawn from the image pair $(F, M_{\boldsymbol{u}})$ the Parzen estimate for the joint PDF $p_{F,M_{\boldsymbol{u}}}$ is written as

$$p_{F,M_{\boldsymbol{u}}}(\mathbf{i}) \approx \frac{1}{n} \sum_{k=1}^{n} K_{\boldsymbol{\rho}}(\mathbf{i} - \mathbf{j}_k). \tag{3.27}$$

If the function K satisfies the conditions

$$\int_{-\infty}^{\infty} K(x)\, \mathsf{d}x = 1 \qquad \int_{-\infty}^{\infty} |K(x)|\, \mathsf{d}x < \infty$$

$$\lim_{x \to \infty} |xK(x)| = 0 \qquad \sup |K(x)| < \infty, \tag{3.28}$$

then the estimate (3.27) converges towards the true PDF for $n \to \infty$ and $\boldsymbol{\rho} \to 0$. For a dense sampling of Ω, (3.27) therefore converges to

$$p_{F,M_{\boldsymbol{u}}}(\mathbf{i}) = \frac{1}{|\Omega|} \int_\Omega K_{\boldsymbol{\rho}}(\mathbf{i} - (F(\mathbf{x}), M_{\boldsymbol{u}}(\mathbf{x}))^T)\, \mathsf{d}\mathbf{x}. \tag{3.29}$$

Examples of functions fulfilling the requirements (3.28) are the Gaussian $G(x) := \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2}$ and the BSpline kernel functions as introduced in (3.6).

Open points in the presented density estimation are the choice of the Parzen kernel function K, its bandwidth ρ and the sampling strategy. These points are actually very important to the resulting estimate and have been the subject of much research (see e. g. [Hahn 10]). In the context of this work, we will only shortly touch on some of the subjects mentioned there. As kernel function it was decided to use a discretized Gaussian for this work. The theoretical disadvantage of the unlimited support of the Gaussian is somewhat alleviated in practice, as in a discrete Gaussian kernel representation only a limited support is used anyway. Ideally the choice of the number of samples, bins and bandwidth all depend on each other and on the chosen kernel function. In this work we always use a dense sampling, i. e. each pixel position in the discretized computational domain Ω is sampled once. A sparser sampling would not really speed up the computation much, as each image pixel has to be touched during the calculation of the transformed gradient image anyway. Unless otherwise noted $m = 64$ bins and a bandwidth of ρ equal to twice the bin size was chosen, as those are values reported to work well in literature (see [Knop 06, Hahn 10]).

Derivative

The calculation of the derivative of the mutual information distance measure $\mathcal{D}_{\text{MI}}$ is rather involved. The derivation used in this work is based on the one presented by Hermosillo in [Herm 02a]. As starting point we have to consider the variation of $\mathcal{D}_{\text{MI}}(F, M_{\mathbf{u}})$ with respect to $\epsilon\boldsymbol{\eta}$ again that defines the Gâteaux derivative.

$$\begin{aligned}
\mathsf{d}\mathcal{D}_{\text{MI}}(F, M_{\boldsymbol{u}}; \boldsymbol{\eta}) &= -\frac{\mathsf{d}}{\mathsf{d}\epsilon} \int_{\mathbb{R}^2} p_{F,M_{\boldsymbol{u}+\epsilon\boldsymbol{\eta}}}(\mathbf{i}) \log \frac{p_{F,M_{\boldsymbol{u}+\epsilon\boldsymbol{\eta}}}(\mathbf{i})}{p_F(i_F)p_{M_{\boldsymbol{u}+\epsilon\boldsymbol{\eta}}}(i_M)} \, \mathsf{d}\mathbf{i} \bigg|_{\epsilon=0} \\
&= -\int_{\mathbb{R}^2} \left(\frac{\mathsf{d}p_{F,M_{\boldsymbol{u}+\epsilon\boldsymbol{\eta}}}(\mathbf{i})}{\mathsf{d}\epsilon}\right) \log \frac{p_{F,M_{\boldsymbol{u}+\epsilon\boldsymbol{\eta}}}(\mathbf{i})}{p_F(i_F)p_{M_{\boldsymbol{u}+\epsilon\boldsymbol{\eta}}}(i_M)} \\
&\qquad + p_{F,M_{\boldsymbol{u}+\epsilon\boldsymbol{\eta}}}(\mathbf{i}) \left(\frac{\mathsf{d}}{\mathsf{d}\epsilon} \log \frac{p_{F,M_{\boldsymbol{u}+\epsilon\boldsymbol{\eta}}}(\mathbf{i})}{p_F(i_F)p_{M_{\boldsymbol{u}+\epsilon\boldsymbol{\eta}}}(i_M)}\right) \, \mathsf{d}\mathbf{i} \bigg|_{\epsilon=0} \\
&= -\int_{\mathbb{R}^2} \left(\frac{\mathsf{d}p_{F,M_{\boldsymbol{u}+\epsilon\boldsymbol{\eta}}}(\mathbf{i})}{\mathsf{d}\epsilon}\right) \log \frac{p_{F,M_{\boldsymbol{u}+\epsilon\boldsymbol{\eta}}}(\mathbf{i})}{p_F(i_F)p_{M_{\boldsymbol{u}+\epsilon\boldsymbol{\eta}}}(i_M)} \\
&\qquad + p_F(i_F)p_{M_{\boldsymbol{u}+\epsilon\boldsymbol{\eta}}}(i_M) \left(\frac{\mathsf{d}}{\mathsf{d}\epsilon} \frac{p_{F,M_{\boldsymbol{u}+\epsilon\boldsymbol{\eta}}}(\mathbf{i})}{p_F(i_F)p_{M_{\boldsymbol{u}+\epsilon\boldsymbol{\eta}}}(i_M)}\right) \, \mathsf{d}\mathbf{i} \bigg|_{\epsilon=0} \\
&= -\int_{\mathbb{R}^2} \left(\frac{\mathsf{d}p_{F,M_{\boldsymbol{u}+\epsilon\boldsymbol{\eta}}}(\mathbf{i})}{\mathsf{d}\epsilon}\right) \log \frac{p_{F,M_{\boldsymbol{u}+\epsilon\boldsymbol{\eta}}}(\mathbf{i})}{p_F(i_F)p_{M_{\boldsymbol{u}+\epsilon\boldsymbol{\eta}}}(i_M)} \\
&\qquad + p_F(i_F)p_{M_{\boldsymbol{u}+\epsilon\boldsymbol{\eta}}}(i_M) \left(\frac{1}{p_F(i_F)p_{M_{\boldsymbol{u}+\epsilon\boldsymbol{\eta}}}(i_M)} \frac{\mathsf{d}p_{F,M_{\boldsymbol{u}+\epsilon\boldsymbol{\eta}}}(\mathbf{i})}{\mathsf{d}\epsilon}\right. \\
&\qquad\qquad \left. - \frac{p_{F,M_{\boldsymbol{u}+\epsilon\boldsymbol{\eta}}}(\mathbf{i})}{p_F(i_F)p_{M_{\boldsymbol{u}+\epsilon\boldsymbol{\eta}}}(i_M)^2} \frac{\mathsf{d}p_{M_{\boldsymbol{u}+\epsilon\boldsymbol{\eta}}}(i_M)}{\mathsf{d}\epsilon}\right) \, \mathsf{d}\mathbf{i} \bigg|_{\epsilon=0} \\
&= -\int_{\mathbb{R}^2} \left(1 + \log \frac{p_{F,M_{\boldsymbol{u}}}(\mathbf{i})}{p_F(i_F)p_{M_{\boldsymbol{u}}}(i_M)}\right) \left(\frac{\mathsf{d}p_{F,M_{\boldsymbol{u}+\epsilon\boldsymbol{\eta}}}(\mathbf{i})}{\mathsf{d}\epsilon}\bigg|_{\epsilon=0}\right) \, \mathsf{d}\mathbf{i} \\
&\qquad + \int_{\mathbb{R}^2} \frac{p_{F,M_{\boldsymbol{u}}}(\mathbf{i})}{p_{M_{\boldsymbol{u}}}(i_M)} \left(\frac{\mathsf{d}p_{M_{\boldsymbol{u}+\epsilon\boldsymbol{\eta}}}(i_M)}{\mathsf{d}\epsilon}\bigg|_{\epsilon=0}\right) \, \mathsf{d}\mathbf{i} \qquad (3.30)
\end{aligned}$$

The second term of (3.30) can be shown to be 0, as

$$\int_{\mathbb{R}^2} \frac{p_{F,M_{\boldsymbol{u}+\epsilon\boldsymbol{\eta}}}(\mathbf{i})}{p_{M_{\boldsymbol{u}+\epsilon\boldsymbol{\eta}}}(i_M)} \frac{\mathsf{d}p_{M_{\boldsymbol{u}+\epsilon\boldsymbol{\eta}}}(i_M)}{\mathsf{d}\epsilon} \, \mathsf{d}\mathbf{i} = \int_{\mathbb{R}} \frac{1}{p_{M_{\boldsymbol{u}+\epsilon\boldsymbol{\eta}}}(i_M)} \frac{\mathsf{d}p_{M_{\boldsymbol{u}+\epsilon\boldsymbol{\eta}}}(i_M)}{\mathsf{d}\epsilon} \underbrace{\int_{\mathbb{R}} p_{F,M_{\boldsymbol{u}+\epsilon\boldsymbol{\eta}}}(\mathbf{i}) \, \mathsf{d}i_F}_{=p_{M_{\boldsymbol{u}+\epsilon\boldsymbol{\eta}}}(i_M)} \, \mathsf{d}i_M$$

$$= \frac{\mathsf{d}}{\mathsf{d}\epsilon} \underbrace{\int_{\mathbb{R}} p_{M_{\boldsymbol{u}+\epsilon\boldsymbol{\eta}}}(i_M)\ \mathsf{d}i_M}_{=1} = 0 \tag{3.31}$$

It is thus only necessary to determine $\left.\frac{\mathsf{d}p_{F,M_{\boldsymbol{u}+\epsilon\boldsymbol{\eta}}}(\mathbf{i})}{\mathsf{d}\epsilon}\right|_{\epsilon=0}$ to get to the final derivative. To do so we substitute $p_{F,M_{\boldsymbol{u}+\epsilon\boldsymbol{\eta}}}$ by its Parzen estimation introduced in (3.29).

$$\left.\frac{\mathsf{d}p_{F,M_{\boldsymbol{u}+\epsilon\boldsymbol{\eta}}}(\mathbf{i})}{\mathsf{d}\epsilon}\right|_{\epsilon=0} = \frac{\mathsf{d}}{\mathsf{d}\epsilon}\frac{1}{|\Omega|}\int_{\Omega} K_{\rho}(\mathbf{i} - (F(\mathbf{x}), M_{\boldsymbol{u}+\epsilon\boldsymbol{\eta}}(\mathbf{x}))^T)\ \mathsf{d}\mathbf{x}\Bigg|_{\epsilon=0}$$

$$= \frac{1}{|\Omega|}\int_{\Omega} \frac{\partial K_{\rho}}{\partial i_M}(\mathbf{i} - (F(\mathbf{x}), M_{\boldsymbol{u}}(\mathbf{x}))^T)(\nabla M_{\boldsymbol{u}}(\mathbf{x}))^T \boldsymbol{\eta}(\mathbf{x})\ \mathsf{d}\mathbf{x} \tag{3.32}$$

Similarly to the definition in the sum of squared differences distance measure, $\nabla M_{\boldsymbol{u}}(\mathbf{x}) = (\nabla M)(\mathbf{x} - \boldsymbol{u}(\mathbf{x}))$ denotes the derivative of the untransformed moving image, accessed at the transformed position $\mathbf{x} - \boldsymbol{u}(\mathbf{x})$. With this result (3.30) can be rewritten as

$$\mathsf{d}\mathcal{D}_{\mathrm{MI}}(F, M_{\boldsymbol{u}}; \boldsymbol{\eta}) = -\frac{1}{|\Omega|}\int_{\mathbb{R}^2}\int_{\Omega} E_{\mathrm{MI}}(\mathbf{i})\frac{\partial K_{\rho}}{\partial i_M}(\mathbf{i} - (F(\mathbf{x}), M_{\boldsymbol{u}}(\mathbf{x}))^T)$$
$$(\nabla M_{\boldsymbol{u}}(\mathbf{x}))^T \boldsymbol{\eta}(\mathbf{x})\ \mathsf{d}\mathbf{x}\ \mathsf{d}\mathbf{i} \tag{3.33}$$

$$\text{where} \qquad E_{\mathrm{MI}}(\mathbf{i}) = 1 + \log \frac{p_{F,M_{\boldsymbol{u}}}(\mathbf{i})}{p_F(i_F)p_{M_{\boldsymbol{u}}}(i_M)}. \tag{3.34}$$

Equation (3.33) can be regarded as a convolution with respect to E_{MI}. Denoting the convolution with $\star$ equation (3.33) is transformed to

$$\mathsf{d}\mathcal{D}_{\mathrm{MI}}(F, M_{\boldsymbol{u}}; \boldsymbol{\eta}) = -\frac{1}{|\Omega|}\int_{\Omega}\left(\frac{\partial K_{\rho}}{\partial i_M} \star E_{\mathrm{MI}}(\mathbf{i})\right)(F(\mathbf{x}), M_{\boldsymbol{u}}(\mathbf{x}))^T\ \ (\nabla M_{\boldsymbol{u}}(\mathbf{x}))^T \boldsymbol{\eta}(\mathbf{x})\ \mathsf{d}\mathbf{x}$$

$$= \left\langle -\left(\frac{\partial K_{\rho}}{\partial i_M} \star E_{\mathrm{MI}}(\mathbf{i})\right)(F(\mathbf{x}), M_{\boldsymbol{u}}(\mathbf{x}))^T\ \ (\nabla M_{\boldsymbol{u}}(\mathbf{x})), \boldsymbol{\eta}\right\rangle_{\mathcal{U}}. \tag{3.35}$$

From (3.15) it is thus possible to identify

$$\nabla_{\boldsymbol{u}}\mathcal{D}_{\mathrm{MI}}(F, M_{\boldsymbol{u}}) = -\left(\frac{\partial K_{\rho}}{\partial i_M} \star E_{\mathrm{MI}}\right)(F(\mathbf{x}), M_{\boldsymbol{u}}(\mathbf{x}))^T\ \ (\nabla M_{\boldsymbol{u}}(\mathbf{x})). \tag{3.36}$$

Note that in contrast to the definition in [Herm 02a] the factor $\frac{1}{|\Omega|}$ disappears in our formulation of $\nabla_{\boldsymbol{u}}\mathcal{D}_{\mathrm{MI}}$ due to the choice of the inner product. Hermosillo also points out that the partial derivative commutes with the convolution, such that (3.36) can be reformulated as

$$\nabla_{\boldsymbol{u}}\mathcal{D}_{\text{MI}}(F, M_{\boldsymbol{u}}) = -\left(K_{\boldsymbol{\rho}} \star \frac{\partial E_{\text{MI}}}{\partial i_M}\right)(F(\mathbf{x}), M_{\boldsymbol{u}}(\mathbf{x}))^T \ (\nabla M_{\boldsymbol{u}}(\mathbf{x})) \tag{3.37}$$

$$\text{where} \quad \frac{\partial E_{\text{MI}}}{\partial i_M}(\mathbf{i}) = \frac{\frac{\partial p_{F,M_{\boldsymbol{u}}}(\mathbf{i})}{\partial i_M}}{p_{F,M_{\boldsymbol{u}}}(\mathbf{i})} - \frac{\frac{\partial p_{M_{\boldsymbol{u}}}}{\partial i_M}}{p_{M_{\boldsymbol{u}}}}. \tag{3.38}$$

For a practical implementation there is no clear advantage of using equation (3.36) or (3.37) as the basis. In (3.36) the partial derivative of the kernel $K_{\boldsymbol{\rho}}$ can be calculated analytically, such that only one discrete filter operation is necessary. In (3.37) something similar is possible, as the joint and moving image PDFs are calculated from Parzen estimation. The same Kernel $K_{\boldsymbol{\rho}}$ is applied there, although not as a convolution. It is, therefore, possible to incorporate the partial derivatives in (3.37) into the application of the Parzen density estimation, such that also just one discretized kernel has to be applied.

Discretization

In the discretization of the mutual information distance measure the calculation of the densities takes a central role again. Evaluating (3.27) is computationally very costly as the sum over all samples is usually quite large. To reduce the computational costs the samples are not used directly but instead binned into a histogram. Equation (3.27) can be rewritten in terms of the histogram entries as

$$\begin{aligned} p_{F,M_{\boldsymbol{u}}}(\mathbf{i}) &= \frac{1}{n}\sum_{k=1}^{m} b_k K_{\boldsymbol{\rho}}(\mathbf{i} - \mathbf{c}_k) \\ &= \left(K_{\boldsymbol{\rho}} \star \frac{\mathbf{b}}{n}\right)(\mathbf{i}), \end{aligned} \tag{3.39}$$

where m is the number of bins in the histogram, $\mathbf{c}_k$ denotes the k-th bin center and $\mathbf{b} = (b_1, \dots, b_m)$ the number of samples in the respective bins. The resulting formulation can be calculated by a discrete convolution of the histogram entries $\mathbf{b}$, normalized to $\sum_{k=1}^{n} b_k = 1$, with the discretized Parzen kernel function $K_{\boldsymbol{\rho}}$. Once $p_{F,M_{\boldsymbol{u}}}$ has been successfully estimated, p_F and $p_{M_{\boldsymbol{u}}}$ can be determined by marginalization. The same is true for the estimation of $\frac{\partial p_{F,M_{\boldsymbol{u}}}}{\partial \mathbf{i}}$ and $\frac{\partial p_{M_{\boldsymbol{u}}}}{\partial i_M}$ as

$$\frac{\partial p_{M_{\boldsymbol{u}}}}{\partial i_M} = \int_{\Omega} \frac{\partial p_{F,M_{\boldsymbol{u}}}}{\partial i_M} \,\mathsf{d}i_F.$$

All in all the discrete version of (3.36) is written as

$$\nabla_{\mathbf{u}}\mathcal{D}_{\text{MI}}(F, M_{\boldsymbol{u}})(\mathbf{x}_i) = -\frac{1}{s}\left(\frac{\partial K_{\boldsymbol{\rho}}}{\partial i_M} \star E_{\text{MI}}\right)(F(\mathbf{x}_i), M_{\mathbf{u}}(\mathbf{x}_i))^T \ (\nabla M_{\mathbf{u}}(\mathbf{x}_i)). \tag{3.40}$$

In practice the values for $E_{\text{MI}}(\mathbf{i})$ are calculated at the same joint intensity values $\mathbf{i}$ that were chosen as bin centers for the histogram in the joint density estimation. This way, the calculation of the mutual information consists of the sampling of the joint images to generate the histogram followed by the application of the Parzen kernel. In order to not get into any boundary handling issues during the computations on the histogram values the histogram is padded with a number of zeros equal to the radius of the local support of the Parzen kernel used for the density estimation. The first convolution with the Parzen kernel for the density estimation of the joint PDF and the marginal PDF of the moving image is then performed on the whole padded histogram, using a 0 Dirichlet boundary condition. From the result the values of E_{MI} can be straightforwardly calculated for every bin according to (3.34).

The second convolution with the partial derivative of the Parzen kernel $\frac{\partial K_{\boldsymbol{\rho}}}{\partial i_M}$ is then executed only inside the original, unpadded histogram region, not in the padded boundary, as the result of the convolution will only be accessed at intensities of $(F(\mathbf{x}_i), M_{\mathbf{u}}(\mathbf{x}_i))^T$. This way no additional boundary handling is necessary as for the region, for which the convolution is applied the local support of the Parzen kernel is fully contained within the padding. For the same reason it is also not necessary to handle any calculation of E_{MI} where the densities are 0. If either $p_F = 0$ or $p_{M_{\mathbf{u}}} =$ then $p_{F,M_{\mathbf{u}}} = 0$. And if $p_{F,M_{\mathbf{u}}} = 0$ then there simply was no sample for that specific intensity combination in the data that was in reach of the support of the Parzen kernel. Thus, no bin within the support of the Parzen kernel around this value will ever be accessed.

Finally, for every discrete position $\mathbf{x}_i$ the image gradient of the moving image is multiplied with the energy term resulting from $\frac{\partial K_{\boldsymbol{\rho}}}{\partial i_M} \star E_{\text{MI}}$ accessed at the according gray value index $\mathbf{i}$. To get a smoothly varying gradient, the values that were only calculated on the histogram bins are linearly interpolated for gray value indices $\mathbf{i}$ that lie in between histogram bin centers.

3.4 Regularizers

In non-parametric, non-rigid registration the regularity of the computed deformation field relies on the regularization that is applied. It also ensures that the problem is well posed (compare Section 3.1.3). Most regularizers have in common that they require the vector field to be locally smooth, penalizing sharp changes like ridges and folds. Accordingly, common regularization techniques are often based on derivatives of the vector field. The diffusion regularizer (Section 3.4.1) quadratically penalizes the first derivative of the vector field, the curvature regularizer (Section 3.4.2) the second. A notable advantage of the curvature regularizer is that it is invariant to affine transforms in the deformation, making the registration less dependent on a good rigid pre-registration.

In the following we will introduce the diffusion and curvature regularizers, with their respective derivatives and discrete approximations, as they are used in the experiments presented in this work.

3.4.1 Diffusion Regularizer

A straightforward way to ensure a smooth vector field is to quadratically penalize any variation of the vector field, i. e. the gradient of the vector field should be minimized. This approach is known as a diffusion regularizer (see [Fisc 99]) as the resulting term is equivalent to the steady state solution of a diffusion problem on the vector field, with the gradient of the distance measure acting as the driving force. It is defined as

$$\mathcal{R}_{\text{Diff}}(\boldsymbol{u}) = \frac{1}{|\Omega|}\int_{\Omega} \|\nabla \boldsymbol{u}(\mathbf{x})\|_F^2 \,\mathsf{dx} = \frac{1}{|\Omega|}\int_{\Omega} \sum_{i=1}^{d} \|\nabla u_i(\mathbf{x})\|^2 \,\mathsf{dx}, \tag{3.41}$$

where $\|,\|_F$ is the Frobenius matrix norm. In the derivative calculation, it can be a bit confusing which dimensionality the respective terms have. The reader is therefore reminded that $\boldsymbol{u} \in \mathbb{R}^d$ is a vector valued function and therefore the gradient is the Jacobian matrix $\nabla \boldsymbol{u} = (\nabla u_1, \ldots, \nabla u_d) \in \mathbb{R}^{d \times d}$. Finally, the Laplacian of a vector field is again a vector valued quantity $\Delta \boldsymbol{u} \in \mathbb{R}^d$. Applying the calculus of variations as outlined in Section 3.2 leads to the following Gâteaux derivative.

$$\begin{aligned}
\mathsf{d}\mathcal{R}_{\text{Diff}}(\boldsymbol{u};\boldsymbol{\eta}) &= \frac{\mathsf{d}}{\mathsf{d}\epsilon}\frac{1}{|\Omega|}\int_{\Omega} \sum_{i=1}^{d} \|\nabla u_i(\mathbf{x}) + \epsilon \nabla \eta_i(\mathbf{x})\|^2 \,\mathsf{dx}\Bigg|_{\epsilon=0} \\
&= \frac{1}{|\Omega|}\int_{\Omega} 2\sum_{i=1}^{d} \nabla \eta_i(\mathbf{x})\; \nabla u_i(\mathbf{x}) \,\mathsf{dx} \qquad (3.42) \\
&= \frac{1}{|\Omega|}\int_{\Omega} 2(\nabla \boldsymbol{\eta}(\mathbf{x}))^T (\nabla \boldsymbol{u}(\mathbf{x})) \,\mathsf{dx}
\end{aligned}$$

In order to get rid of the derivative of the test function we have to make use of boundary conditions imposed on the function $\boldsymbol{u}$. In this case either von Neumann (fixed first derivative across the domain boundary) or Dirichlet (fixed known function on the domain boundary) boundary condition can be chosen. The boundary condition imposed on $\boldsymbol{u}$ also implicitly impose constraints on the test functions $\boldsymbol{\eta}$ as these may not change $\boldsymbol{u}$ in a way that would lead $\boldsymbol{u} + \epsilon\boldsymbol{\eta}$ to violate the boundary condition. The Dirichlet boundary condition is formally defined as

$$\boldsymbol{u}(\mathbf{x}) = \boldsymbol{c}(\mathbf{x}) \qquad \text{implies} \quad \boldsymbol{\eta}(\mathbf{x}) = 0 \qquad \forall \mathbf{x} \in \partial\Omega, \tag{3.43}$$

where $\boldsymbol{c}(\mathbf{x})$ is the known function of boundary values for $\boldsymbol{u}$ and $\partial\Omega$ denotes the boundary of the domain Ω. For the definition of the von Neumann boundary condition it is necessary to introduce $\boldsymbol{n}$ which is a function that for every coordinate $\mathbf{x} \in \partial\Omega$ is orthogonal to the domain boundary and has unit length. Formally, if $\boldsymbol{b}(t) : \mathbb{R} \mapsto \mathbb{R}^d$ is a piecewise differentiable curve that defines the domain boundary $\partial\Omega$ then $\boldsymbol{n}(\boldsymbol{b}(t))^T \nabla \boldsymbol{b}(t) = 0$ and $\|\boldsymbol{n}(\mathbf{x})\| = 1 \quad \forall \mathbf{x} in \partial\Omega$. The von Neumann boundary conditions on a vector valued function $\boldsymbol{u}$ then specify for the Jacobian $\nabla \boldsymbol{u}$ and therefore for each component ∇u_i that

$$\begin{aligned} \boldsymbol{n}(\mathbf{x})^T \nabla \boldsymbol{u}(\mathbf{x}) &= \left(\boldsymbol{n}(\mathbf{x})^T \nabla u_1(\mathbf{x}), \dots, \boldsymbol{n}(\mathbf{x})^T \nabla u_d(\mathbf{x})\right)^T = \mathbf{0} \quad \forall \mathbf{x} \in \partial\Omega \\ &\text{implies} \\ \boldsymbol{n}(\mathbf{x})^T \nabla \boldsymbol{\eta}(\mathbf{x}) &= \left(\boldsymbol{n}(\mathbf{x})^T \nabla \eta_1(\mathbf{x}), \dots, \boldsymbol{n}(\mathbf{x})^T \nabla \eta_d(\mathbf{x})\right) = \mathbf{0} \quad \forall \mathbf{x} \in \partial\Omega. \end{aligned} \tag{3.44}$$

These boundary conditions can be used in conjunction with Green's theorem [Bron 99] to simplify equation (3.42). Green's theorem is the multi-dimensional extension of integration by parts and is used here to eliminate the derivative of the test function $\nabla \boldsymbol{\eta}$.

$$\begin{aligned} \mathsf{d}\mathcal{R}_{\text{Diff}}(\boldsymbol{u}; \boldsymbol{\eta}) &= \underbrace{\frac{2}{|\Omega|} \int_{\partial\Omega} \boldsymbol{\eta}(\mathbf{x})^T (\nabla \boldsymbol{u}(\mathbf{x}))\, \mathsf{dx}}_{=0 \text{ by (3.43) or (3.44)}} - \frac{2}{|\Omega|} \int_{\Omega} \boldsymbol{\eta}(\mathbf{x})^T (\Delta \boldsymbol{u}(\mathbf{x}))\, \mathsf{dx} \\ &= \langle -2\Delta \boldsymbol{u}, \boldsymbol{\eta} \rangle_{\mathcal{U}} \\ &= 0 \quad \forall \boldsymbol{\eta} \in \mathcal{U} \end{aligned} \tag{3.45}$$

In accordance with (3.15), we thus define

$$\nabla_{\boldsymbol{u}} \mathcal{R}_{\text{Diff}}(\boldsymbol{u}) = -2\Delta \boldsymbol{u}. \tag{3.46}$$

The regularizer and its gradient formulation is discretized by replacing the differential operator Δ with its discretized equivalent $\mathbf{A}_\Delta$, resulting from the use of second order central differences.

$$\mathcal{R}_{\text{Diff}}(\mathbf{u}) = \frac{1}{s} \mathbf{u}^T \mathbf{A}_\Delta \mathbf{u} \tag{3.47}$$

$$\nabla_{\mathbf{u}} \mathcal{R}_{\text{Diff}}(\mathbf{u}) = -\frac{2}{s} \mathbf{A}_\Delta \mathbf{u} \tag{3.48}$$

$\mathbf{A}_\Delta$ is a block structured matrix of the form

$$\begin{aligned} \mathbf{A}_\Delta &= \operatorname{diag}\left(\mathbf{A}_{\Delta,b}, \dots, \mathbf{A}_{\Delta,b}\right) \\ &= \begin{pmatrix} \mathbf{A}_{\Delta,b} & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \mathbf{A}_{\Delta,b} \end{pmatrix} \end{aligned} \tag{3.49}$$

where each block matrix $\mathbf{A}_{\Delta,b}$ represents a diffusion matrix In stencil notation (see Appendix (B)) the block matrices $\mathbf{A}_{\Delta,b}$ can be denoted as

$$\frac{1}{h^2} \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \tag{3.50}$$

for 2-D and a homogeneous image spacing h. More generally the stencil entries e for a diffusion stencil in d dimensions can be given as

$$e_{i_1,\dots,i_d} = \frac{1}{h^2}\begin{cases} -2d & \text{if } i_k = 0 \quad \forall 1 \le k \le d \\ 1 & \text{if } |i_l| = 1 \quad \wedge \quad i_k = 0 \quad \forall 1 \le k \le d \quad \wedge \quad k \ne l \\ 0 & \text{otherwise} \end{cases}, \tag{3.51}$$

where the subscripts i_k indicate the offset in the according dimension from the stencil center. The full matrix $\mathbf{A}_\Delta$ has therefore a size of $ds \times ds$, where d is the dimensionality and s the number of pixels/voxels with which the data has been discretized. Additionally, the used boundary condition has to be incorporated into ${\mathbf{A}_\Delta}^{(k,k)}$. In the discrete we make the simplification that the domain boundaries are always aligned with one of the dimensions, due to the rectangular nature of the discrete pixels.

In the following we will make a few simplifications for an easier notation. We only consider the matrix ${\mathbf{A}_\Delta}^{k,k}$ so only one component of the vector field at a time. We will also only consider the domain boundary of the l-th dimension and therefore omit the dimension index, for the other dimensions. The value $u_{i_1,\dots,i_d}$ is therefore simply written as u_j with $j = i_l$. Furthermore, only the lower domain boundary is considered i. e. entry u_1 is the pixel right adjacent to the boundary and u_0 is already outside the computational domain.

The Dirichlet boundary condition (3.43) just require all values outside of the computational domain to be equal to 0, and thus $u_0 = 0$. This behavior can be achieved by deforming the matrix stencil at the boundary such that in the changed stencil entry $e_{i_1,\dots,i_d} = 0$ for $i_j < 0$.

For the von Neumann boundary condition (3.44) it has to be ensured that derivatives over the domain boundary are 0. Using a backward difference we can conclude that

$$\begin{aligned} \frac{u_1 - u_0}{h} &= 0 \\ \mathbf{u}_0 &= \mathbf{u}_1. \end{aligned} \tag{3.52}$$

The stencil thus has to be modified thus that any access across the boundary is mapped to the adjacent pixel inside the domain, i. e. for the diffusion stencil from (3.51) placed on a boundary pixel this results in

$$e_{i_1,\dots,i_d} = \frac{1}{h^2}\begin{cases} -2d+1 & \text{if } i_k = 0 \quad \forall 1 \le k \le d \\ 1 & \text{if } |i_l| = 1 \quad \wedge \quad i_k = 0 \quad \wedge i_j \ge 0 \quad \forall 1 \le k \le d \quad \wedge \quad k \ne l \\ 0 & \text{otherwise} \end{cases}. \tag{3.53}$$

3.4.2 Curvature Regularizer

The so-called curvature regularization minimizes second order derivatives, more specifically the Laplacian Δ of the vector field $\boldsymbol{u}$ to constrain the non-rigidity of the deformation. This approach was used by Horn and Schunck for optical flow in [Horn 81] and was introduced as registration regularizer by Fischer et al. in [Fisc 03b]. It is defined as

$$\mathcal{R}_{\text{Curv}}(\boldsymbol{u}) = \|\Delta \boldsymbol{u}\|_{\mathcal{U}}^2 = \frac{1}{|\Omega|} \int_\Omega \|\Delta \boldsymbol{u}(\mathbf{x})\|^2 \, \mathsf{dx}. \tag{3.54}$$

An important property of $\mathcal{R}_{\text{Curv}}(\boldsymbol{u})$ is that affine transforms are not penalized by this regularizer, as $\Delta(\mathbf{Ax} + \mathbf{t}) = \mathbf{0}$. As such the quality of the rigid registration that usually precedes the application of a non-rigid registration loses some importance. It is also of some advantage for inter patient registrations that global scalings are not penalized. The Gâteaux derivative of the curvature regularizer is

$$\begin{aligned} \mathsf{d}\mathcal{R}_{\text{Curv}}(\boldsymbol{u}; \boldsymbol{\eta}) &= \frac{\mathsf{d}}{\mathsf{d}\epsilon} \frac{1}{|\Omega|} \int_\Omega \|\Delta \boldsymbol{u}(\mathbf{x}) + \epsilon \Delta \boldsymbol{\eta}(\mathbf{x})\|^2 \, \mathsf{dx} \bigg|_{\epsilon=0} \\ &= \frac{1}{|\Omega|} \int_\Omega 2(\Delta \boldsymbol{\eta}(\mathbf{x}))^T (\Delta \boldsymbol{u}(\mathbf{x})) \, \mathsf{dx}. \end{aligned} \tag{3.55}$$

Similarly to the derivation of the derivative of the diffusion regularizer we have to make use of boundary conditions. The used boundary conditions in this case are of von Neumann type, i. e. (3.44) and additionally

$$\boldsymbol{n}(\mathbf{x})^T \, \nabla \Delta \boldsymbol{u}(\mathbf{x}) = 0 \quad \text{implies} \qquad \boldsymbol{n}(\mathbf{x})^T \, \nabla \Delta \boldsymbol{\eta}(\mathbf{x}) = 0 \qquad \forall \mathbf{x} \in \partial\Omega. \tag{3.56}$$

By the application of Green's theorem (3.55) can be simplified to

$$\begin{aligned} \mathsf{d}\mathcal{R}_{\text{Curv}}(\boldsymbol{u}; \boldsymbol{\eta}) &= \frac{2}{|\Omega|} \int_\Omega (\Delta \boldsymbol{\eta}(\mathbf{x}))^T (\Delta \boldsymbol{u}(\mathbf{x})) \, \mathsf{dx} \\ &= \frac{2}{|\Omega|} \sum_{=1}^{d} \int_\Omega (\Delta \eta_i(\mathbf{x}))^T (\Delta u_i(\mathbf{x})) \, \mathsf{dx} \\ &= \frac{2}{|\Omega|} \sum_{=1}^{d} \int_{\partial\Omega} \underbrace{(\boldsymbol{n}(\mathbf{x})^T \nabla \eta_i(\mathbf{x}))}_{=0 \text{ by } (3.44)} \Delta u_i(\mathbf{x}) \, \mathsf{dx} - \int_\Omega (\nabla \eta_i(\mathbf{x}))^T (\nabla \Delta u_i(\mathbf{x})) \, \mathsf{dx} \\ &= \frac{2}{|\Omega|} \sum_{=1}^{d} - \int_{\partial\Omega} \underbrace{(\boldsymbol{n}(\mathbf{x})^T \nabla \Delta u_i(\mathbf{x}))}_{=0 \text{ by } (3.56)} \eta_i(\mathbf{x}) \, \mathsf{dx}_\Omega + \int_\Omega (\Delta^2 u_i(\mathbf{x})) \eta_i(\mathbf{x}) \, \mathsf{dx} \\ &= \frac{2}{|\Omega|} \int_\Omega \boldsymbol{\eta}(\mathbf{x})^T (\Delta^2 \boldsymbol{u}(\mathbf{x})) \, \mathsf{dx} \\ &= \left\langle 2\Delta^2 \boldsymbol{u}, \boldsymbol{\eta} \right\rangle_{\mathcal{U}} \\ &= 0 \quad \forall \boldsymbol{\eta} \in \mathcal{U} \end{aligned} \tag{3.57}$$

$$. \tag{3.58}$$

We can therefore define that

$$\nabla_{\boldsymbol{u}} \mathcal{R}_{\text{Curv}}(\boldsymbol{u}) = \Delta^2 \boldsymbol{u}. \tag{3.59}$$

In the discrete the curvature regularizer and its derivative is represented by a linear operator

$$\mathcal{R}_{\text{Curv}}(\mathbf{u}) = \frac{1}{s} \mathbf{u}^T \mathbf{A}_{\Delta^2} \mathbf{u} \tag{3.60}$$

$$\nabla_{\mathbf{u}} \mathcal{R}_{\text{Curv}}(\mathbf{u}) = \frac{2}{s} \mathbf{A}_{\Delta^2} \mathbf{u}. \tag{3.61}$$

There are two feasible ways to discretize the operator Δ^2 to the matrix $\mathbf{A}_{\Delta^2}$. The first is based on the observation that $\Delta^2 = \Delta^T \Delta$. In the previous section it was shown that Δ can be discretized with the matrix $\mathbf{A}_{\Delta}$. The matrix resulting from the discretization of Δ^2 can therefore be expressed as $\mathbf{A}_{\Delta^2} = \mathbf{A}_{\Delta}{}^T \mathbf{A}_{\Delta}$, with discretized von Neumann boundary conditions.

Another way to arrive at a discretized operator is to directly discretize Δ^2, with finite differences. The resulting matrix $\mathbf{A}_{\Delta^2}$ has the same block structure as the diffusion regularizer (see (3.49)), only with a matrix representing the discretized second order derivatives.

$$\mathbf{A}_{\Delta^2} = \operatorname{diag}(\mathbf{A}_{\Delta^2}{}^{(1,1)}, \dots, \mathbf{A}_{\Delta^2}{}^{(d,d)}) \tag{3.62}$$

The matrices $\mathbf{A}_{\Delta^2}{}^{(i,i)}$ acting on the individual dimensions of the vector field are discretized using finite differences. In stencil notation, with a homogeneous spacing h of the data, they can be denoted as

$$\frac{1}{h^4} \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 2 & -8 & 2 & 0 \\ 1 & -8 & 20 & -8 & 1 \\ 0 & 2 & -8 & 2 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}, \tag{3.63}$$

in 2-D. The matrix stencil in 3-D can be given as

$$e_{i_1,i_2,i_3} = \frac{1}{h^4} \begin{cases} 42 & \text{if } i_1 = i_2 = i_3 = 0 \\ -12 & \text{if } \sum_{l=1}^{3} |i_l| = 1 \\ 2 & \text{if } |i_k| = 1 \quad \wedge \quad \sum_{l=1}^{3} |i_l| = 2 \\ 1 & \text{if } |i_k| = 2 \quad \wedge \quad \sum_{l=1}^{3} |i_l| = 2 \\ 0 & \text{otherwise} \end{cases}. \tag{3.64}$$

For the incorporation of the boundary condition we employ the same simplifications for the notation as described for the diffusion regularizer above. For the curvature regularizer we have to incorporate the von Neumann boundary condition on the vector field (3.44), which was already discussed in (3.52) and on the vector fields

Laplacian (3.52). Looking again at the domain boundary in one dimension, this second condition can be discretized using finite differences as

$$\begin{aligned}
\frac{(\Delta u)_1 - (\Delta u)_0}{h} &= 0 \\
(\Delta u)_1 &= (\Delta u)_0 \\
\frac{u_0 - 2u_1 + u_2}{h^2} &= \frac{u_{-1} - 2u_0 + u_1}{h^2} && \text{using finite differences} \\
u_2 - u_1 &= u_{-1} - u_0 \\
u_2 &= u_{-1} && \text{using (3.52) } (u_1 = u_0). \qquad (3.65)
\end{aligned}$$

We thus end up with a kind of mirroring boundary handling. In 2-D these boundary conditions will therefore result in the following deformed stencils, which are respectively one pixel off of the image boundary and directly adjacent to it.

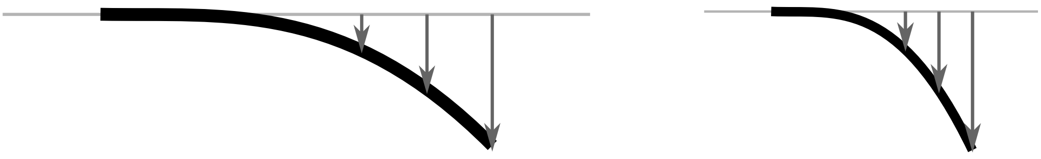
$$\underset{\text{one pixel removed from the boundary}}{\frac{1}{h^4}\begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 2 & -8 & 2 & 0 \\ 0 & -7 & 20 & -8 & 1 \\ 0 & 2 & -8 & 2 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}} \quad \underset{\text{directly on the boundary}}{\frac{1}{h^4}\begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -6 & 2 & 0 \\ 0 & 0 & 12 & -7 & 1 \\ 0 & 0 & -6 & 2 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}} \qquad (3.66)$$

3.5 Parameter Selection

The registration formulation discussed this far has only one parameter that is a fundamental choice of the user: The weighting parameter α that balances the regularizer against the distance measure. All other parameters discussed, like the number of bins or the kernel width in the mutual information implementation are only numerical parameters that can be automatically estimated (compare [Hahn 10]). The α parameter, however, is a user choice that decides how much deformation should be allowed to get a good match. The problem is therefore how to allow the user to choose a value for α that will yield a result that matches his expectations. Ideally the parameter governing the stiffness of the calculated deformation should therefore have an intuitive meaning attached to it. In practice, however, the values that have to be chosen for α to get a desired result depend on the used distance measures and regularizers and on the images that are registered. It is not even well defined what a change in the value for α will change in the registration result. Doubling α will not necessarily result in a "twice as rigid" deformation. It is unclear how one would define a "twice as rigid" deformation to begin with. We therefore examine in the following how the parameter specifying the stiffness of the desired transform can be specified in a way that is at least somewhat intuitive and behaves in a predictable fashion.

3.5.1 Stiffness

The term stiffness or rigidity of a deformation is intended to describe how much the deformation field is allowed to change over a certain distance. This stiffness is



(a) (b)

Figure 3.6: The same deformation (gray arrows) applied to a (a) "normal" and in (b) shorter, "compressed" stick. Intuition tells us that bending the shorter stick by the same amount as the long one takes more force.

measured in the algorithms used in this work by the regularization terms. For instance let us consider a 1-D object, i. e. a stick in a 2-D world. If some force is applied to one end of this idealized stick, while the other is fixed, it will deform according to the strength of the force and the physical characteristics of the stick. If the same forces are applied to a stick twice as long we would expect it to be bended to a much larger extent. Or, if we wanted to bend a shorter stick by the same amount, we would expect to need much more force (compare Figure 3.6). This example is the intuitive model we will rely on to set the parameter for the stiffness of the deformation. In respect to our regularization terms the change in length of the stick corresponds to a change in size of the problem domain Ω in which the deformation is represented, while the deformation itself i. e. the length of the offsets remains unchanged. If the same deformation is scaled (compressed or expanded) to another problem domain the value of the regularizer will change accordingly. In order to evaluate this mathematically we define the domain scaled by κ on which the regularizer is evaluated as Ω_κ, and the quantities represented in it, as

$$\Omega_\kappa := \left\{\mathbf{x}^{(\kappa)} \mid \mathbf{x}^{(\kappa)} = \frac{\mathbf{x}}{\kappa} \wedge \mathbf{x} \in \Omega\right\} \quad |\Omega_\kappa| = \frac{|\Omega|}{\kappa^d} \quad \boldsymbol{u}^{(\kappa)}(\mathbf{x}^{(\kappa)}) := \boldsymbol{u}^{(\kappa)}\left(\frac{\mathbf{x}}{\kappa}\right) := \boldsymbol{u}(\mathbf{x}). \tag{3.67}$$

We will now use this to take a look at the change in the regularization energy when the problem domain is scaled. First, let us consider the derivative $\nabla\boldsymbol{u}^{(\kappa)}(\mathbf{x})$. Substituting $\boldsymbol{u}$ for $\boldsymbol{u}^{(\kappa)}$ in the derivative yields

$$\nabla\boldsymbol{u}^{(\kappa)}(\mathbf{x}) = \nabla_{\mathbf{x}}\boldsymbol{u}(\kappa\mathbf{x}) = \kappa\nabla\boldsymbol{u}(\kappa\mathbf{x}). \tag{3.68}$$

It is imperative to keep in mind here to apply the chain rule when substituting $\boldsymbol{u}(\kappa\mathbf{x})$ for $\boldsymbol{u}^{(\kappa)}(\mathbf{x})$. Evaluating the diffusion regularizer from Section 3.4.1 on the scaled problem domain Ω_κ yields

$$\begin{aligned}\mathcal{R}_{\text{Diff}}(\boldsymbol{u}^{(\kappa)}) &= \frac{1}{|\Omega_\kappa|}\int_{\Omega_\kappa}\sum_{i=1}^{d}\|(\nabla u_i^{(\kappa)})(\mathbf{x}^{(\kappa)})\|^2 \,\mathsf{d}\mathbf{x}^{(\kappa)} \\ &= \frac{\kappa^d}{|\Omega|}\int_{\Omega_\kappa}\sum_{i=1}^{d}\|\kappa\ \nabla u_i(\kappa\mathbf{x}^{(\kappa)})\|^2 \,\mathsf{d}\mathbf{x}^{(\kappa)}\end{aligned}$$

$$
\begin{aligned}
&= \frac{\kappa^{d+1}}{|\Omega|} \int_\Omega \sum_{i=1}^{d} \|\nabla u_i(\mathbf{x})\|^2 \frac{1}{\kappa^d} \,\mathsf{d}\mathbf{x} && \text{by substitution } \mathbf{x}^{(\kappa)} = \frac{\mathbf{x}}{\kappa} \\
&= \frac{\kappa}{|\Omega|} \int_\Omega \sum_{i=1}^{d} \|\nabla u_i(\mathbf{x})\|^2 \,\mathsf{d}\mathbf{x}. && (3.69)
\end{aligned}
$$

Doing the same for the curvature regularizer gives

$$
\begin{aligned}
\mathcal{R}_{\text{Curv}}(u) &= \frac{1}{|\Omega_\kappa|} \int_{\Omega_\kappa} \|\Delta \boldsymbol{u}^{(\kappa)}(\mathbf{x}^{(\kappa)})\|^2 \,\mathsf{d}\mathbf{x}^{(\kappa)} \\
&= \frac{\kappa^d}{|\Omega|} \int_{\Omega_\kappa} \|\nabla_{\mathbf{x}}^T(\kappa \, \nabla \boldsymbol{u}(\kappa \mathbf{x}^{(\kappa)}))\|^2 \,\mathsf{d}\mathbf{x}^{(\kappa)} \\
&= \frac{\kappa^d}{|\Omega|} \int_{\Omega_\kappa} \|\kappa^2 \, \Delta \boldsymbol{u}(\kappa \mathbf{x}^{(\kappa)})\|^2 \,\mathsf{d}\mathbf{x}^{(\kappa)} \\
&= \frac{\kappa^{d+2}}{|\Omega|} \int_{\Omega} \|\Delta \boldsymbol{u}(\mathbf{x})\|^2 \frac{1}{\kappa^d} \,\mathsf{d}\mathbf{x} && \text{by substitution } \mathbf{x}^{(\kappa)} = \frac{\mathbf{x}}{\kappa} \\
&= \frac{\kappa^2}{|\Omega|} \int_{\Omega} \|\Delta \boldsymbol{u}(\mathbf{x})\|^2 \,\mathsf{d}\mathbf{x}. && (3.70)
\end{aligned}
$$

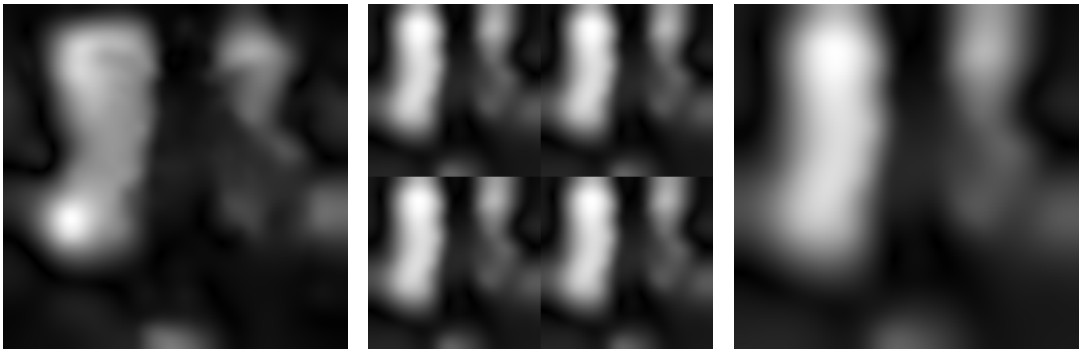
Therefore, while a change in the size of the computational domain by a factor of κ induces the same amount of scaling in the diffusion regularizer, it is scaled quadratically i. e. by κ^2 in the curvature regularizer. Therefore α has to be varied differently for those two regularizers in order to get an (intuitively) similar increase in the stiffness of the transform.

To get a more intuitively behaving parameter governing the stiffness of the transform we propose to use the parameter κ introduced in this discussion instead of α. We thus consider the regularizer in an artificially scaled computational domain Ω_κ. A high value for κ leads to a compressed domain, which generates higher values in the regularizer as the vector field appears to vary more rapidly. Thus the parameter setting for κ has an intuitive meaning attached that allows the user to better predict the consequences of a specific increase or decrease of κ. Doubling the stiffness κ leads to domain Ω_κ scaled down by a factor of 2 and thus a "twice as smooth" deformation. That this intuition actually makes sense is illustrated in Figure 3.7 (compare also the results presented in Figure 6.2). The same approach would also be applicable to other standard regularizers known from literature, as for instance, elastic.

Computationally the use of κ instead of α to weight the regularization term, does not change anything, as for a given regularizer a parameter setting of κ can be readily translated into a setting for α, for the diffusion and curvature regularizers, respectively.

$$
\alpha_{\text{Diff}}(\kappa) := \kappa \qquad\qquad \alpha_{\text{Curv}}(\kappa) := \kappa^2. \tag{3.71}
$$

In the discrete formulation the same effect can be achieved in our discretized regularization terms (3.48) and (3.61) by simply scaling the image spacing h by the stiffness parameter as $h_\kappa = \frac{h}{\kappa}$.



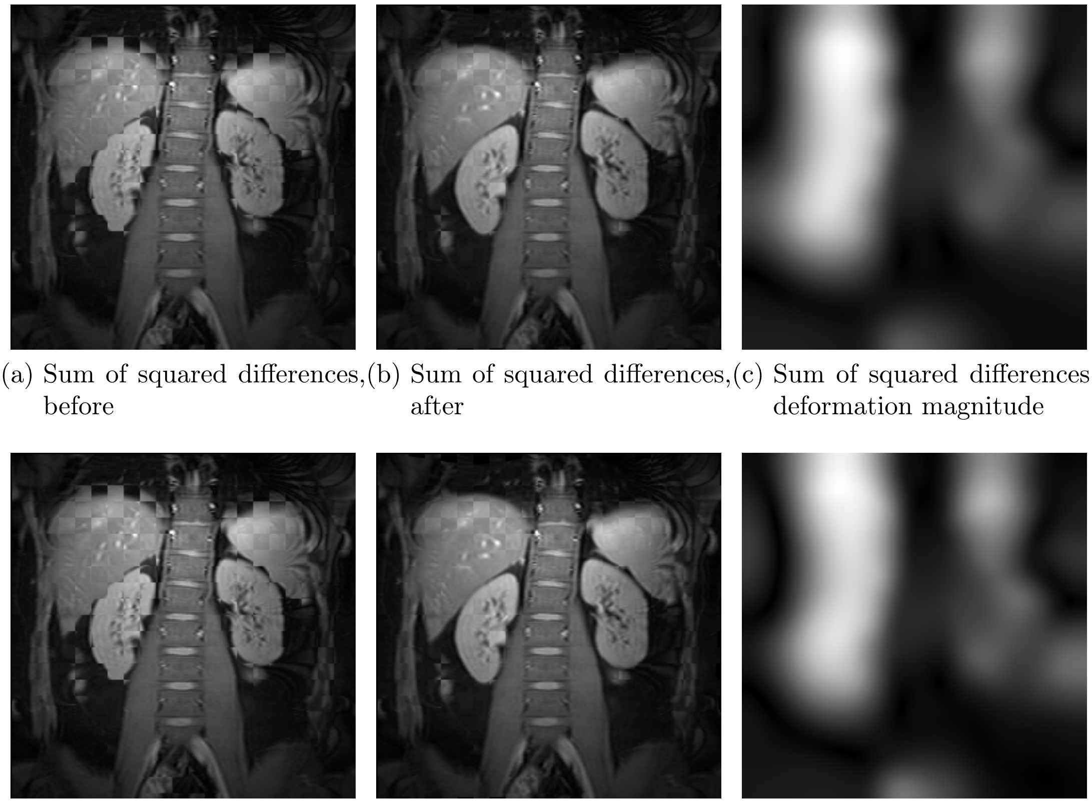
(a) $\kappa = 20$ (b) $\kappa = 40$, scaled and tiled (c) $\kappa = 40$

Figure 3.7: Example registration (see Section 4.7.1 for a description of the dataset) with a curvature regularizer and two different values for the stiffness κ. As κ is doubled between (a) and (c) the deformation should be "twice as smooth". This is illustrated by (b) which is the same result as (c), but scaled down by a factor of 2 and tiled to yield an image that should have roughly the same amount of variation and similarly sharp edges as (a).

3.5.2 Distance Measure Scaling

Even though there is now an intuition how a change in the parameter κ changes the registration result, there is still no information about how a given κ will act on a new, unknown dataset. The first part of this problem is due to the different distance measures, which can evaluate to quite different values when applied to the same dataset. As the different distance measures have been designed with different goals for a good match in mind, it is certainly not possible to map them in a way that they will produce directly comparable values. However, it should be possible to rescale them at least into a common value range defined by a best case and worst case match. To this end we assume two identical images as the theoretical best case match. As a "realistic" worst case match we will work with a random mapping of the pixels in the two images to each other. Naturally, there will often be even worse matches available for a specific distance measure but to give meaning to the rescaling the assumed worst case match should be the same for all distance measures. A linear rescaling is applied to map the values produced by the distance measure into a range of size of 1. This way the possible variation between different distance measures is at least limited, which will allow, to a certain extent, to generate similar results with different distance measures, while keeping the stiffness κ constant. Even though this rescaling cannot change anything about the differing non-linear behavior of the distance measures, it worked surprisingly well in practice, as exemplified in Figure 3.8. Nonetheless, this kind of rescaling is not applied during any of the other experiments in this work, to allow for an easier reproducibility of the results.

In the following the details of the linear rescaling are introduced for the distance measures used in this work (sum of squared differences and mutual information).



(a) Sum of squared differences, before (b) Sum of squared differences, after (c) Sum of squared differences, deformation magnitude

(d) Mutual information, before (e) Mutual information, after (f) Mutual information, deformation magnitude

Figure 3.8: Result of two registrations on a mono-modal dataset (see Section 4.7.1 for a description of the dataset) using the rescaled mutual information and sum of squared differences distance measures with the same stiffness parameter $\kappa = 4$. First column (a)(d): checkerboard overlay before registration; Second column (b)(e): checkerboard overlay after registration; Third column (c)(f): gradient magnitude image of deformation $\mathbf{u}$

Sum of Squared Differences

In the case of the sum of squared differences distance measure the best possible match, i. e. two identical images will result in a distance of $\mathcal{D}_{\text{SSD}} = 0$. For the worst case match, a random association of pixels with each other i. e. they are assumed to be independent, we have to estimate the values in the difference image created by this match. As a simplification we will assume that the input image intensities i_F and i_M can be described by normal distributions i. e. their corresponding means and variances. Using these we can also model the image intensities of the difference image as a normal distribution with

$$\begin{aligned} i_{\text{diff}} &:= i_F - i_M \\ \mathrm{E}\left[i_{\text{diff}}\right] &= \mathrm{E}\left[i_F\right] - \mathrm{E}\left[i_M\right] && (3.72) \\ \mathrm{Var}\left[i_{\text{diff}}\right] &= \mathrm{Var}\left[i_F\right] + \mathrm{Var}\left[i_M\right]. && (3.73) \end{aligned}$$

The identity for the variance is based on the given worst case assumption that the distributions for i_F and i_M are independent. In the discrete the sum of squared differences distance measure computes the arithmetic mean of the squared differences in the domain. As the arithmetic mean is the same as the discrete expectation, we can write this as

$$\mathcal{D}_{\text{SSD}}(F, M) = \mathrm{E}\left[i_{\text{diff}}^2\right] \tag{3.74}$$

This can be reformulated by using the following equivalence.

$$\begin{aligned} \mathrm{Var}\left[i_{\text{diff}}\right] &= \mathrm{E}\left[\left(i_{\text{diff}} - \mathrm{E}\left[i_{\text{diff}}\right]\right)^2\right] \\ &= \mathrm{E}\left[i_{\text{diff}}^2 - 2\mathrm{E}\left[i_{\text{diff}}\right] i_{\text{diff}} + \mathrm{E}\left[i_{\text{diff}}\right]^2\right] \\ &= \mathrm{E}\left[i_{\text{diff}}^2\right] - 2\mathrm{E}\left[i_{\text{diff}}\right]^2 + \mathrm{E}\left[i_{\text{diff}}\right]^2 \\ &= \mathrm{E}\left[i_{\text{diff}}^2\right] - \mathrm{E}\left[i_{\text{diff}}\right]^2 \end{aligned} \tag{3.75}$$

Solving the above for $\mathrm{E}\left[i_{\text{diff}}^2\right]$ we can conclude that

$$\begin{aligned} \mathcal{D}_{\text{SSD}}(F, M) &= \mathrm{E}\left[i_{\text{diff}}^2\right] \\ &= \mathrm{Var}\left[i_{\text{diff}}\right] + \mathrm{E}\left[i_{\text{diff}}\right]^2 \\ &= \mathrm{Var}\left[i_F\right] + \mathrm{Var}\left[i_M\right] + \left(\mathrm{E}\left[i_F\right] - \mathrm{E}\left[i_M\right]\right)^2 \quad \text{by eq. (3.72) and (3.73)}\,. \end{aligned} \tag{3.76}$$

The sum of squared differences distance measure is therefore scaled with the inverse of $\mathrm{E}\left[i_{\text{diff}}^2\right]$, which can be computed from the expectation and variance of the input images. The rescaled distance measures will then yield results in the range $[0, 1]$ most of the time. As the upper bound is based on the assumption that the worst case match is a random matching of gray values, which is not the actual worst case possible, it is not guaranteed that it will always be contained within this range.

Mutual Information

For the mutual information the assumed worst case match of an independence of the distributions of i_F and i_M means that $p_{F,M_{\boldsymbol{u}}}(\mathbf{i}) = p_F(i_F)p_M(i_M)$. The mutual information distance measure will evaluate to

$$\mathcal{D}_{\text{MI}}(F, M_{\boldsymbol{u}}) = -\int_{\mathbb{R}^2} p_{F,M}(\mathbf{i}) \log \frac{p_{F,M}(\mathbf{i})}{p_F(i_F)p_M(i_M)} \,\mathsf{d}\mathbf{i} = 0. \tag{3.77}$$

Please note that the distance measures in this work have to be minimized and the mutual information distance measure is defined as the negative mutual information. This is therefore the largest value the mutual information distance measure will take.

The best possible match $F(\mathbf{x}) = M(\mathbf{x})$ is characterized by $p_F(i) = p_M(i)$. For the joint probability it has to hold that $p_{F,M}(\mathbf{i})\,\mathsf{d}\mathbf{i} = p_F(i_F)\,\mathsf{d}i_F = p_M(i_M)\,\mathsf{d}i_M \iff i_F = i_M$ and $p_{F,M}(\mathbf{i}) = 0 \iff i_F \neq i_M$.

$$\begin{aligned}
\mathcal{D}_{\text{MI}}(F, M) &= -\int_{\mathbb{R}^2} p_{F,M}(\mathbf{i}) \log \frac{p_{F,M}(\mathbf{i})}{p_F(i_F)p_M(i_M)} \,\mathsf{d}\mathbf{i} \\
&= -\int_{\mathbb{R}} p_F(i_F) \log \frac{\mathsf{d}i_M}{p_F(i_F)} \,\mathsf{d}i_F \\
&= \int_{\mathbb{R}} p_F(i_F) \log p_F(i_F) \,\mathsf{d}i_F - \int_{\mathbb{R}} p_F(i_F)(\log \,\mathsf{d}i_M) \,\mathsf{d}i_F \\
&= -\mathcal{H}(p_F) - \log \,\mathsf{d}i_M \qquad (3.78) \\
&= -\mathcal{H}(p_M) - \log \,\mathsf{d}i_F \qquad (3.79)
\end{aligned}$$

Here, $\mathcal{H}$ denotes the continuous entropy. In the continuous case the value $\log \,\mathsf{d}i_M$ is equal to infinity. In practice, however, we never have to deal with this problem, as $\mathsf{d}i_M$ simply corresponds to the quantization of our input images in the mutual information calculation i. e. the resolution used during the histogram binning. In reality p_M and p_F will rarely be identical. We therefore use the maximum absolute rescaling factor that can be achieved by the formulation. The mutual information distance measure can thus be rescaled by a constant value to yield values in the interval $[-1, 0]$ which has the same range as the rescaled sum of squared differences distance measure described above. In contrast to the range we chose for rescaling the sum of squared differences distance measure, the rescaled mutual information distance measure will actually not be able to generate results outside of this range.

Another possibility would certainly be to use any of the normalized mutual information variants available. However, all of these make use of a division by a non-constant factor, which leads to more complicated derivatives of the measure.

3.5.3 Regularizers

A similar problem is posed by the regularizer. The values of the regularizer are usually not standardized and can describe rather different things. A similar approach to the rescaling of the distance measures is difficult as a worst case deformation is rather impossible to define. Instead of the worst case deformation one might choose to define a “standard” deformation for which all regularizers should yield the same value. Such a “standard” deformation would have to be similar to deformations one would expect to see in practice. The choice that we explore in this context is based on a sine function. The advantage of using a trigonometric function for the specification of the vector field is that it yields a smooth variation that does not vanish in the higher order derivatives. Additionally, in the cases described here the regularizers can be calculated explicitly for the chosen “standard” deformation. The specific deformation used is

$$\boldsymbol{u}_{\text{std}}(\mathbf{x}) := \frac{\mathbf{1}}{d}\sum_{i=1}^{d}\sin(x_i). \tag{3.80}$$

The regularizer is evaluated on $\boldsymbol{u}_{\text{std}}$ over the domain $\Omega = \{\mathbf{x} \mid -\pi \leq x_i \leq \pi \quad i = 1,\ldots,d\}$ to obtain the standardization value. We thus derive for the diffusion regularizer that

$$\begin{aligned}\mathcal{R}_{\text{Diff}}(\boldsymbol{u}_{\text{std}}) &= \frac{1}{|\Omega|}\int_\Omega \|\nabla \boldsymbol{u}_{\text{std}}(\mathbf{x})\|_F^2 \;\mathsf{dx}\\ &= \frac{1}{|\Omega|}\int_\Omega \|\nabla\frac{\mathbf{1}}{d}\sum_{i=1}^{d}\sin(x_i)\|^2 \;\mathsf{dx}\\ &= \frac{1}{|\Omega|d}\int_\Omega \sum_{i=1}^{d}\cos^2(x_i)\;\mathsf{dx}\\ &= \frac{1}{|\Omega|d}\sum_{k=1}^{d}\frac{|\Omega|}{2}\;\mathsf{dx}\\ &= \frac{1}{2},\end{aligned} \tag{3.81}$$

The elimination of the integral is done by performing partial integration as follows

$$\begin{aligned}\int_\Omega \cos^2(x_i)\;\mathsf{dx} &= [\sin(x_i)\cos(x_i)]_\Omega + \int_\Omega \sin^2(x_i)\;\mathsf{dx}\\ &= \int_\Omega 1\;\mathsf{dx} - \int_\Omega \cos^2(x_i)\;\mathsf{dx}\\ \int_\Omega \cos^2(x_i)\;\mathsf{dx} &= \frac{|\Omega|}{2}.\end{aligned}$$

Analogously we can calculate for the curvature regularizer

$$\begin{aligned}\mathcal{R}_{\text{Curv}}(\boldsymbol{u}_{\text{std}}) &= \|\Delta \boldsymbol{u}_{\text{std}}\|_{\mathcal{U}}^2\\ &= \frac{1}{|\Omega|}\int_\Omega \|\Delta\frac{\mathbf{1}}{d}\sum_{i=1}^{d}\sin(x_i)\|^2\;\mathsf{dx}\end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{|\Omega|} \int_\Omega \| - \frac{\mathbf{1}}{d} \sum_{i=1}^{d} \sin(x_i) \|^2 \ \mathsf{dx} \\
&= \frac{1}{|\Omega| d} \int_\Omega \sum_{i=1}^{d} \sum_{j=1}^{d} \sin(x_i) \sin(x_j) \ \mathsf{dx} \\
&= \frac{1}{2|\Omega| d} \sum_{i=1}^{d} \sum_{j=1}^{d} \underbrace{\int_\Omega \cos(x_i - x_j) - \cos(x_i + x_j) \ \mathsf{dx}}_{=0} \qquad (3.82)
\end{aligned}
$$

The integrals under the double sum are for $i = j$

$$\int_\Omega \cos(0) - \cos(2x_i) \ \mathsf{dx} = |\Omega|.$$

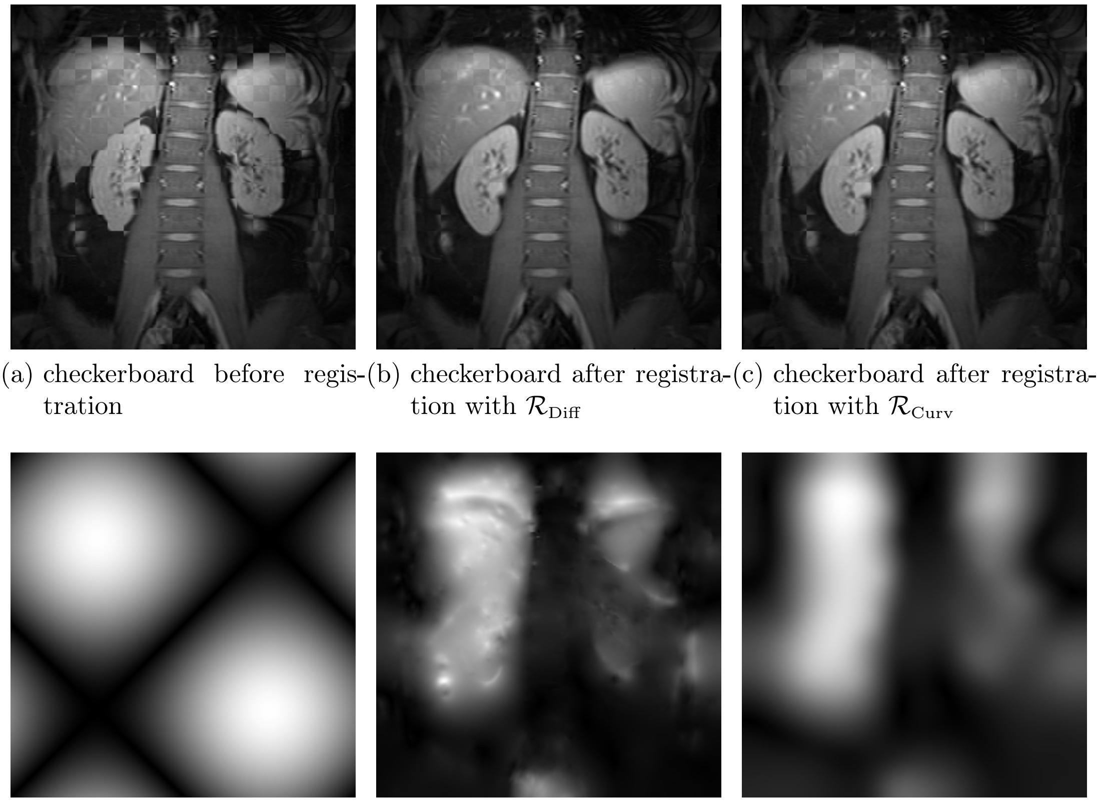
For $i \neq j$ we can again use partial integration to show that

$$
\begin{aligned}
&\int_{\Omega_{x_j}} \int_{\Omega_{x_i}} \cos(x_i - x_j) - \cos(x_i + x_j) \ \mathsf{d}x_i \ \mathsf{d}x_j \\
&= \int_{\Omega_{x_j}} \sin(\pi - x_j) - \sin(-\pi - x_j) - \sin(\pi + x_j) + \sin(-\pi + x_j) \ \mathsf{d}x_j \\
&= -\cos(2\pi) + \cos(0) + \cos(-2\pi) - \cos(0) + \cos(2\pi) - \cos(0) - \cos(0) + \cos(-2\pi) \\
&= 0.
\end{aligned}
$$

Inserting these identities into (3.82) we get

$$
\begin{aligned}
\mathcal{R}_{\text{Curv}}(\boldsymbol{u}_{\text{std}}) &= \|\Delta \boldsymbol{u}_{\text{std}}\|_{\mathcal{U}}^2 \\
&= \frac{1}{2|\Omega| d} \sum_{i=1}^{d} \sum_{j=1}^{d} \underbrace{\int_\Omega \cos(x_i - x_j) - \cos(x_i + x_j) \ \mathsf{dx}}_{=0} \\
&= \frac{1}{2|\Omega| d} d|\Omega| \\
&= \frac{1}{2}. \qquad (3.83)
\end{aligned}
$$

For this specific test function there is accordingly no rescaling necessary. But this changes quickly if the test function is changed. For example if the sin function is varied with twice the angular velocity i. e. $\sin 2x_i$ then $\mathcal{R}_{\text{Diff}}(\boldsymbol{u}_{\text{std}}) = 4.5$ and $\mathcal{R}_{\text{Curv}}(\boldsymbol{u}_{\text{std}}) = 40.5$, due to the application of the chain rule in the derivatives. This will therefore only work to a very limited extent (compare Figure 3.9). Due to this, and also to keep the results reproducible, no rescaling of this kind was applied during any of the other experiments contained in this work.



(a) checkerboard before registration

(b) checkerboard after registration with $\mathcal{R}_{\text{Diff}}$

(c) checkerboard after registration with $\mathcal{R}_{\text{Curv}}$

(d) deformation magnitude $\boldsymbol{u}_{\text{std}}$

(e) deformation magnitude after registration with $\mathcal{R}_{\text{Diff}}$

(f) deformation magnitude after registration with $\mathcal{R}_{\text{Diff}}$

Figure 3.9: Result of two registrations on a mono-modal dataset (see Section 4.7.1 for a description of the dataset) using the sum of squared differences distance measure. Compared is the use of the diffusion and curvature regularizer weighted with a stiffness parameter $\kappa = 40$. First row, checkerboard overlays: (a) before registration, (b) after registration with diffusion regularization, (c) after registration with curvature regularization; Second row, magnitude deformation fields: (d) reference deformation $\boldsymbol{u}_{\text{std}}$ (e) deformation after registration with diffusion regularization, (f) deformation after registration with curvature regularization

Chapter 4

Optimization in Non-rigid, Non-parametric Registration

An important part of any registration approach is the optimization of the chosen energy terms. Experiments show that the result of the registration can strongly vary with the amount of optimization applied and also with the optimization algorithm used. This is mostly due to the complicated energy landscape represented by the target energy functional $\mathcal{E}$, which usually has several local minima. In itself this is not at all surprising, but it demonstrates that before specialized smoother and matching energies are designed and evaluated it is necessary to ensure that they are properly optimized. Otherwise, any evaluation will not only judge a specific registration formulation, but also the quality of the optimizer employed. Vice versa the choice of the distance measure and regularizer plays an important role in judging the performance of the optimization algorithm, as some measures are, for example due to a higher non-linearity, easier to optimize than others. The numerical implementation of the optimization also has a significant impact on the practical usability of the registration in terms of runtime, memory requirements and numerical parameters needed for a specific algorithm.

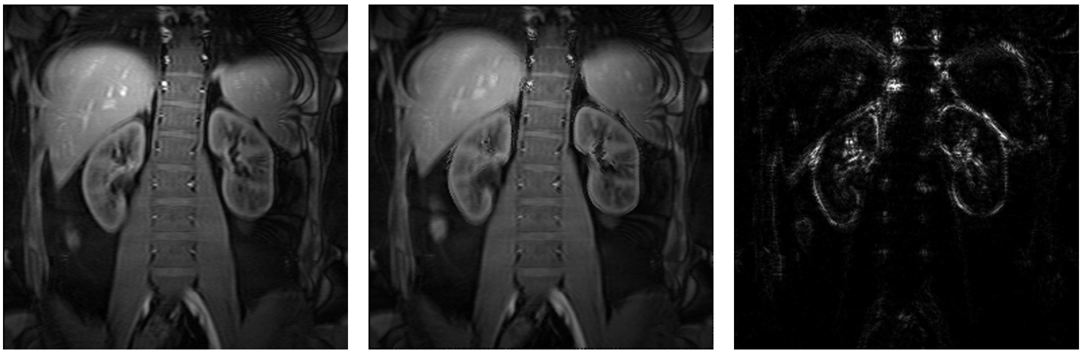
All optimization algorithms discussed in the following are based on solving the Euler Lagrange equations arising from $\nabla_{\boldsymbol{u}}\mathcal{E}(F, M, \boldsymbol{u}) = 0$, or rather the discretized equivalent $\nabla_{\mathbf{u}}\mathcal{E}(F, M, \mathbf{u}) = \mathbf{0}$.

4.1 Gradient Descent

The simplest approach to minimize the energy functional $\mathcal{E}$ is to employ a gradient descent optimization scheme. For the registration problem this is written as

$$\begin{aligned}\mathbf{u}^{(t+1)} &= \mathbf{u}^{(t)} - \tau\nabla_{\mathbf{u}}\mathcal{E}(F, M, \mathbf{u}) \\ &= \mathbf{u}^{(t)} - \tau\left(\nabla_{\mathbf{u}}\mathcal{D}(F, M_{\mathbf{u}^{(t)}}) + \alpha\nabla_{\mathbf{u}}\mathcal{R}(\mathbf{u})\right) \\ &= \mathbf{u}^{(t)} - \tau\left(\nabla_{\mathbf{u}}\mathcal{D}(F, M_{\mathbf{u}^{(t)}}) + \alpha\mathbf{A}\mathbf{u}^{(t)}\right),\end{aligned} \tag{4.1}$$

where the matrix $\mathbf{A}$ is the linear operator resulting from the discretization of $\nabla_{\mathbf{u}}\mathcal{R}$ i. e. $\mathbf{A}_{\triangle}$ (3.48) or $\mathbf{A}_{\triangle^2}$ (3.61). The superscripts in brackets in this formulation indicate



(a) Fixed image F (b) Moving image $M_{\mathbf{u}}$ (c) Magnitude deformation $\|\mathbf{u}\|$

Figure 4.1: Result of an explicit optimization scheme using the sum of squared differences as distance measure and diffusion as regularizer after 20 iterations. In order to get any significant steps at all, line searching was disabled. (a) is the reference image, (b) the deformed moving image after the registration. It is clearly visible that the algorithm did not regularize the deformation sufficiently such that only at the edges in the image changes were performed. This is also reflected in the magnitude image of the deformation (c).

the index of the variable over the course of the non-linear iteration. Theoretically gradient descent is guaranteed to eventually converge to a local minimum, provided that $\mathcal{E}$ is sufficiently smooth and that the step size τ is chosen sufficiently small. In practice however, the calculated descent direction is often so bad that a step size resulting in an actual decrease of the energy is so small that no real improvement is achievable. This problem occurred most pronounced in our experiments when using the mutual information as distance measure.

A reason for these problems can be observed when looking at the vector field after a view iterations of a registration using the sum of squared differences distance measure and a diffusion regularizer. Figure (c) shows the gradient magnitude of the calculated deformation. It exhibits strong edges and thus a high variance of the image gradient which are properties of a very non-smooth deformation field. This result can be better understood by considering the regularizer which is supposed to keep the deformation smooth. An application of, for instance, the diffusion regularizer is similar to diffusion filtering on the gradient of the distance measure. From diffusion filtering it is known that in an explicit scheme small step sizes and many iterations are necessary (compare e. g. [Weic 98]) to solve this problem. But in a gradient descent based registration scheme, the diffusion problem only gets one iteration step applied before the driving force, the gradient of the distance measure changes again. As a result, the deformation is not sufficiently smoothed during each iteration step which results in the poor performance of the gradient descent scheme.

4.2 Semi-Implicit Gradient Descent

As consequence to the observations made above, the regularizer has to be treated differently. As all the regularizing terms we are dealing with are implemented as linear operators, it is possible to solve for them implicitly. We therefore consider the deformation $\mathbf{u}$ that the regularizer is applied to already in the next time step, i. e. the regularizer is applied to $\mathbf{u}^{(t+1)}$ instead of $\mathbf{u}^{(t)}$. Equation (4.1) thus becomes

$$\begin{aligned} \mathbf{u}^{(t+1)} &= \mathbf{u}^{(t)} - \tau \nabla_{\mathbf{u}} \mathcal{D}(F, M_{\mathbf{u}^{(t)}}) - \tau\alpha \mathbf{A}\mathbf{u}^{(t+1)} \\ (\mathbf{I} + \tau\alpha\mathbf{A})\mathbf{u}^{(t+1)} &= \mathbf{u}^{(t)} - \tau \nabla_{\mathbf{u}} \mathcal{D}(F, M_{\mathbf{u}^{(t)}}). \end{aligned} \tag{4.2}$$

An implicit formulation is known to be much more stable, allowing the use of larger step sizes, which offsets the added computational cost. In this case the formulation is only semi-implicit, but as the regularizer, which was identified as one of the main problem sources, is handled implicitly this is still a big improvement and the convergence is much more stable [Schw 06]. In practice this semi-implicit scheme is relatively well behaved and delivers good results after a sufficient number of iterations [Schw 06].

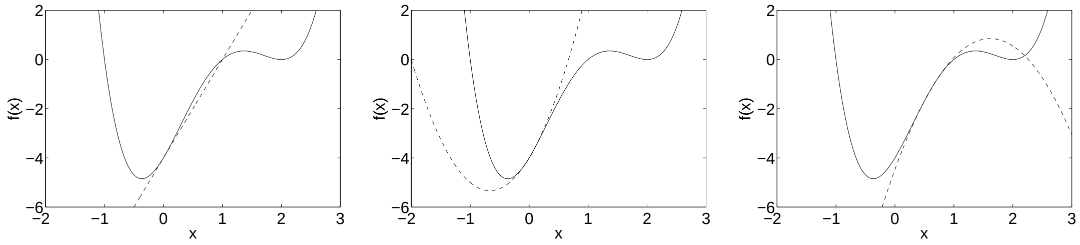
In exchange it is now necessary to solve for a large sparse linear system. As the system matrix $\mathbf{I} + \tau\alpha\mathbf{A}$ is positive definite (see Appendix A), it is possible to apply iterative sparse matrix solvers like Gauss-Seidel, Jacobi, Krylov subspace methods (like conjugate gradient (CG)) (see [Schw 06]) or Multigrid (see [Brig 00]). For the regularizers used in this work there is also a very efficient direct solver based on the Fast Fourier Transform (FFT) (see [Fisc 99, Mode 04]). Another downside of this formulation is that it complicates line searching. Whenever the step size τ is changed the linear system has to be solved again, which is computationally quite costly.

4.3 Inexact Newton

In the previous sections, the regularizing term was identified as a major problem in the optimization. As it is known that in the original energy $\mathcal{E}$ the regularizer is only a second order term, a second order method like Newton's method that makes use of the known Hessian $\mathbf{H}$ will not make any "errors" as far as the regularizer is concerned. The corresponding Newton type optimization will then look like

$$\begin{aligned} \mathbf{u}^{(t+1)} &= \mathbf{u}^{(t)} - \tau \mathbf{H}^{-1}_{\mathcal{E}(F,M,\mathbf{u}^{(t)})} \nabla_{\mathbf{u}} \mathcal{E}(F, M, \mathbf{u}^{(t)}) \\ &= \mathbf{u}^{(t)} - \tau \left(\mathbf{H}_{\mathcal{D}(F,M_{\mathbf{u}^{(t)}})} + \alpha \mathbf{H}_{\mathcal{R}} \right)^{-1} \left(\nabla_{\mathbf{u}} \mathcal{D}(F, M_{\mathbf{u}^{(t)}}) + \alpha \nabla_{\mathbf{u}} \mathcal{R}(\mathbf{u}^{(t)}) \right) \\ &= \mathbf{u}^{(t)} - \tau \left(\mathbf{H}_{\mathcal{D}(F,M_{\mathbf{u}^{(t)}})} + \alpha \mathbf{A} \right)^{-1} \left(\nabla_{\mathbf{u}} \mathcal{D}(F, M_{\mathbf{u}^{(t)}}) + \alpha \mathbf{A}\mathbf{u}^{(t)} \right). \end{aligned} \tag{4.3}$$

Additionally, the Hessian provides valuable information about how much the energy changes with respect to each variable, and thus how much change should be applied at each discrete position in the vector field. The gradient alone does not contain this kind of information. While the gradient does only indicate in which direction the method has to step in order to minimize the measure, the Hessian used in Newton's method also provides an approximation of how long this step has to be. In 1-D (see Figure 4.2) this corresponds to gradient descent fitting a tangent to the function, while Newton's method fits a parabola, with its minimum marking the natural, indicated step length. This also explains one of the weaknesses of Newton's method: If the



(a) Gradient descent $x = 0$ (b) Newton $x = 0$ (c) Newton $x = 0.6$

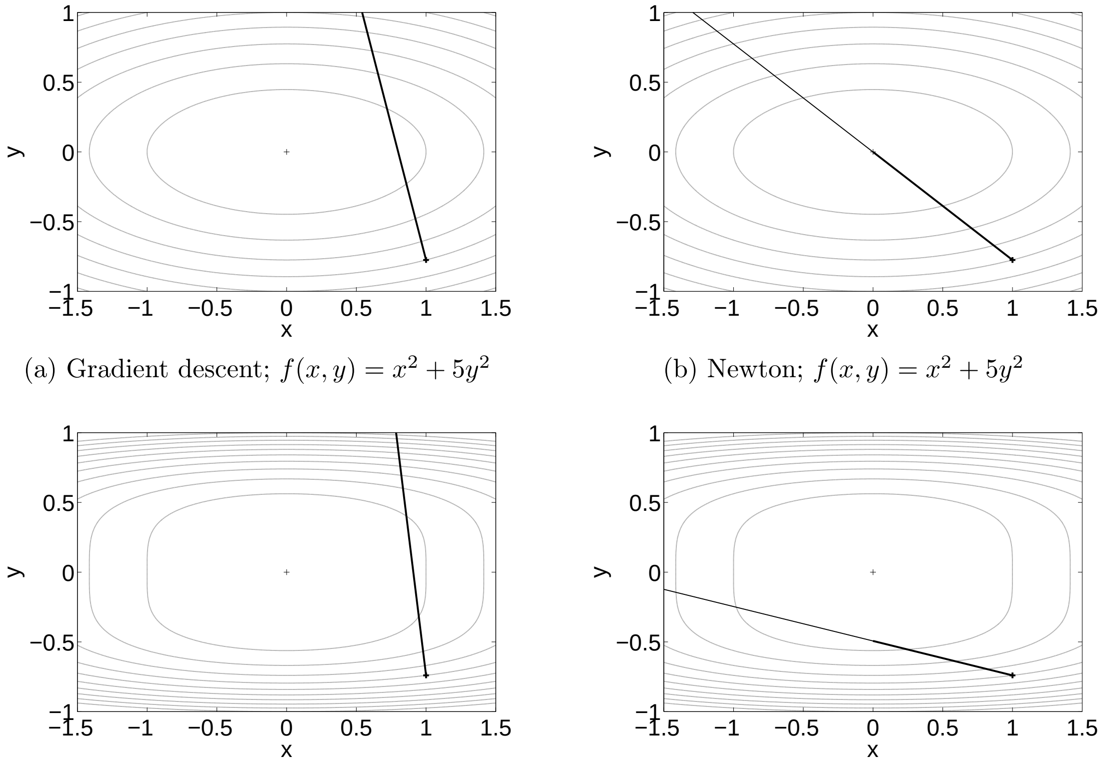
Figure 4.2: Illustration of one step of a (a) gradient descent and (b) Newton's method. The black curve is the function $f(x) = (x-1)(x+1)(x-2)^2$. The blue curve depicts the tangent or respectively the parabola the methods fit to the function. (c) shows Newton's method in a concave area of the function f, where it turns into a maximization method.

function is not convex at the current position $\mathbf{x}$ (i. e. positive definite in d-D) then Newton's method is not a minimization method anymore, as the fitted parabola will flip upside down (see 4.2c). If the function is concave (negative definite in d-D), it will therefore maximize the function and if it is indefinite (only possible in d-D), it will converge towards a saddle point.

If the advantage of Newton's method was limited to just the global step length, this would not be a big deal as we calculate a good step length in the line search procedure anyway. The real advantage is that this behavior is exhibited in each variable of a vector valued function, thus allowing the method to scale the gradient vector individually in each component (compare Figure 4.3).

The benefit of this behavior can be illustrated by taking a look at the derivative of the mutual information distance measure (3.36). It depends on the image gradient $\nabla M_{\boldsymbol{u}}$. If the image statistics are the same for two image regions the overall mutual information energy will depend on both image regions in a similar way. For example in Figure 4.3 two circles have to be matched onto two squares. Due to the image gradient present in the gradient of the mutual information energy, the gradient strength differs for the two circles. Practically this means that over the course of the iteration the white area will be deformed much quicker than the gray area. In extreme cases, this can mean that an area with high gradients completely dominates the optimization process and an area with weaker gradients gets almost ignored. A very common example where this might happen is in registering computed tomography images where the bone and contrast agent to soft tissue contrast can generate such high gradients in the distance measure. In practice, a good step size control will try to find a balance between both image regions. The better solution, however, would be a Newton type method that can use its Hessian (or an approximation thereof) to rescale the gradient appropriately.

The challenge in applying Newton's method is to calculate the Hessian of the distance measure $\mathbf{H}_{\mathcal{D}(F,M_{\mathbf{u}^{(t)}})}$ and to solve the large linear system, arising from the inverse of the Hessian of the energy $\mathbf{H}_{\mathcal{E}(F,M,\mathbf{u}^{(t)})}$. For the sum of squared differences distance measure (see Section 3.3.1) the discrete Hessian can be calculated from the



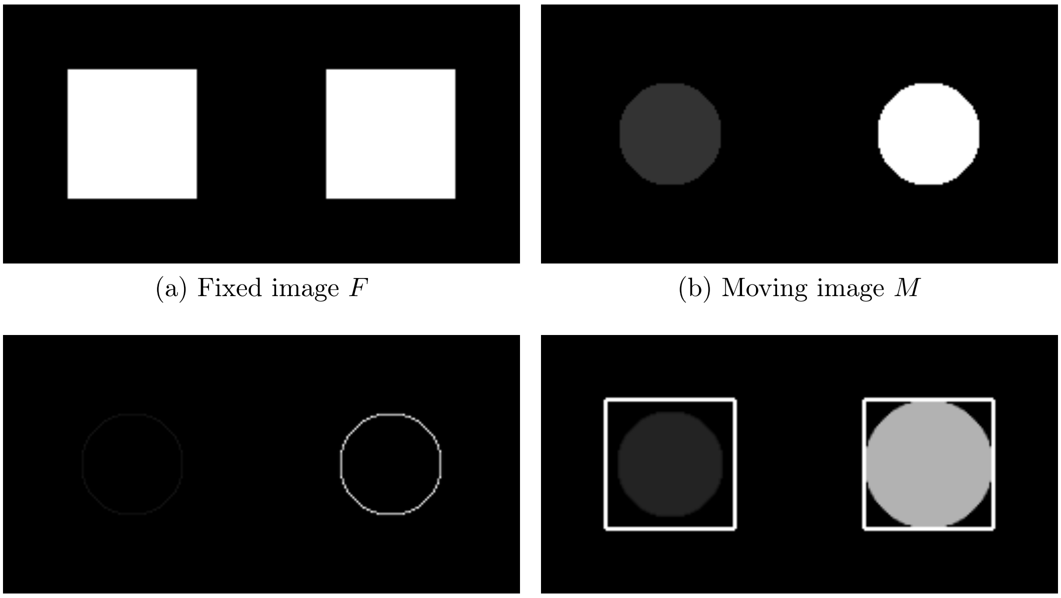
(a) Gradient descent; $f(x, y) = x^2 + 5y^2$

(b) Newton; $f(x, y) = x^2 + 5y^2$

(c) Gradient descent; $f(x, y) = x^2 + 10y^4$

(d) Newton; $f(x, y) = x^2 + 10y^4$

Figure 4.3: Illustration of one step of (a)(c) gradient descent and (b)(d) Newton's method. The fat blue line is the step the methods would take with a step length of 1, the thin blue line shows the range that a line search could pick from. The 2-D functions (a)(b) $f(x, y) = x^2 + 5y^2$ and (c)(d) $f(x, y) = x^2 + 10y^4$, are visualized by isolines at $f(x, y) = 1, \ldots$. In the purely quadratic function, Newton's method is obviously superior as the target function can be correctly fitted. In the mixed quadratic and quartic function, Newton's method is not able to converge in one step, but still it is visible that the search direction is superior to that given by gradient descent and also the indicated step size is reasonable.



(a) Fixed image F

(b) Moving image M

(c) Mutual information gradient magnitude $\|\boldsymbol{u}(\mathbf{x})\|$

(d) Overlay $M_{\boldsymbol{u}}$ and $\|\boldsymbol{u}(\mathbf{x})\|$

Figure 4.4: Behavior of the gradient of the mutual information distance measure. (a) fixed image, (b) moving image, (c) mutual information gradient magnitude, (d) moving image after one step of semi-implicit gradient descent, overlaid with fixed image contour

discrete derivative $\nabla_{\mathbf{u}}\mathcal{D}_{\mathrm{SSD}}$ (3.23) as

$$\begin{aligned}\mathbf{H}_{\mathcal{D}_{\mathrm{SSD}}(F,M_{\mathbf{u}(t)})}(\mathbf{x}_i) &= \nabla^2_{\mathbf{u}}\mathcal{D}_{\mathrm{SSD}}(F, M_{\boldsymbol{u}})(\mathbf{x}_i)\\ &= \nabla_{\mathbf{u}}\left(2(F(\mathbf{x}_i) - M_{\mathbf{u}}(\mathbf{x}_i))\nabla M_{\mathbf{u}}(\mathbf{x}_i)\right)\\ &= \frac{2}{s}(\nabla M_{\mathbf{u}}(\mathbf{x}_i))(\nabla M_{\mathbf{u}}(\mathbf{x}_i))^T + \frac{2}{s}(F(\mathbf{x}_i) - M_{\mathbf{u}}(\mathbf{x}_i))\Delta M_{\mathbf{u}}(\mathbf{x}_i). \quad (4.4)\end{aligned}$$

The according discrete Hessian matrix $\mathbf{H}_{\mathcal{D}(F,M_{\mathbf{u}})}$ therefore has on its main diagonal the squared first order partial derivatives and the second order partial derivatives of $M_{\mathbf{u}}$. On the off-diagonals there are the mixed products of the first order partial derivatives and the mixed second order partial derivatives coupling the dimensions. Theoretically one could just use this term, calculate it and use it in Newton's iteration. In practice, however, there are several problems with this approach. One problem is that on image data that is often degraded by noise, it is pretty hard to calculate a good second derivative of the image $\Delta M_{\mathbf{u}}$. Another problem is that Newton's method is only guaranteed to converge to a minimum on convex functions i. e. the Hessian of the function is required to be positive definite. For the energy $\mathcal{E}(F, M, \mathbf{u})$ this is in general not the case, due to the dependence on the image content. As far as the Hessian is concerned, it is again the part with the second derivatives $\Delta M_{\mathbf{u}}$ which causes the problem. The summand with the first derivatives is inherently positive definite, as is the regularizer (see Appendix A).

A possible solution to this problem is to just drop the term $\frac{2}{s}(F - M_{\mathbf{u}})\Delta M_{\mathbf{u}}$,

$$\left(\begin{array}{c|c|c}
\ddots & \ddots & \ddots \\
(\nabla_{x_1}M_{\mathbf{u}})^2 & (\nabla_{x_1}M_{\mathbf{u}})(\nabla_{x_2}M_{\mathbf{u}}) & (\nabla_{x_1}M_{\mathbf{u}})(\nabla_{x_3}M_{\mathbf{u}}) \\
\ddots & \ddots & \ddots \\
\hline
\ddots & \ddots & \ddots \\
(\nabla_{x_1}M_{\mathbf{u}})(\nabla_{x_2}M_{\mathbf{u}}) & (\nabla_{x_2}M_{\mathbf{u}})^2 & (\nabla_{x_2}M_{\mathbf{u}})(\nabla_{x_3}M_{\mathbf{u}}) \\
\ddots & \ddots & \ddots \\
\hline
\ddots & \ddots & \ddots \\
(\nabla_{x_1}M_{\mathbf{u}})(\nabla_{x_3}M_{\mathbf{u}}) & (\nabla_{x_2}M_{\mathbf{u}})(\nabla_{x_3}M_{\mathbf{u}}) & (\nabla_{x_3}M_{\mathbf{u}})^2 \\
\ddots & \ddots & \ddots
\end{array}\right)$$

Figure 4.5: General structure of the reduced Hessian $\mathbf{H}_{\mathcal{D}(F,M_{\mathbf{u}})}$ for the sum of squared differences distance measure. The delimited boxes indicate the block matrices for the dimension (here 3-D). On the main diagonal there are the quadratic first order partial derivatives, on the off-diagonal the corresponding mixed products of the first order partial derivatives.

arguing that close to the correct solution $F - M_{\mathbf{u}}$ will be small and ultimately tend to 0. The Hessian of the sum of squared distance measure is therefore approximated just by

$$\mathbf{H}_{\mathcal{D}_{\mathrm{SSD}}(F,M_{\mathbf{u}(t)})}(\mathbf{x}_i) \approx \frac{2}{s}(\nabla M_{\mathbf{u}})(\nabla M_{\mathbf{u}})^T. \tag{4.5}$$

Only the first order partial derivatives are retained. This results in a matrix as depicted in Figure 4.3). Hömke et al. proposed in [Homk 06] an approach using this approximation for the Hessian. The work makes clear that the main problem using this approximation of the Hessian is to solve for the resulting inverse problem. The system matrix has, depending on the image content strongly varying stencil entries and adds additional matrix couplings between the dimensions, such that solving for the diffusion or curvature regularizer cannot be broken down into a sub-problem for each dimension anymore. This not only makes the application of an FFT based solver scheme impossible, it also makes the application of an iterative numeric solver rather complicated. This is also shown in [Kalm 03]. In [Homk 06] a multigrid solver which has to employ line smoothing in order to obtain acceptable convergence rates is used. All of this means that solving the linear problem with this approximation of the Hessian introduces a lot of overhead and is computationally quite costly. Henn et al. introduced a closely related approach in [Henn 03], with the main difference that they use an iterative regularization, by applying the regularizing term only in the Hessian and not in the energy functional itself. They circumvented the problem of the linear solver by working with a direct sparse matrix solver. But direct sparse matrix solvers are limited in their applicability mostly to 2-D, due to memory constraints. Also they are usually much slower for large problems than good iterative solvers.

In order to alleviate the problem with the linear solver, Haber et al. [Habe 06] use only the main diagonal of $(\nabla M_{\mathbf{u}})(\nabla M_{\mathbf{u}})^T$ instead of the full matrix. This still does not make an FFT based solver applicable, but it reduces matrix coupling and makes the implementation of a Multigrid solver for this problem easier.

$$\mathbf{H}_{\mathcal{D}_{\mathrm{SSD}}(F,M_{\mathbf{u}})} \approx 2\,\mathrm{diag}\left(\left(\frac{\partial M_{\mathbf{u}}(x)}{\partial x_{1,1}}\right)^2, \ldots, \left(\frac{\partial M_{\mathbf{u}}(x)}{\partial x_{d,s}}\right)^2\right). \tag{4.6}$$

The good convergence of a method only employing a diagonal matrix as an approximation of the Hessian of the distance measure and the observation that actually the regularizer and not the distance measure was the cause of most problems in the optimizations, prompted us to simplify even further. We therefore examined approaches that employ only a scaled identity matrix $\epsilon\mathbf{I}$ as approximation to the Hessian of the distance measure $\mathbf{H}_{\mathcal{D}_{\mathrm{SSD}}(F,M_{\mathbf{u}})}$.

$$\mathbf{H}_{\mathcal{D}_{\mathrm{SSD}}(F,M_{\mathbf{u}})} \approx \epsilon\mathbf{I} \tag{4.7}$$

This eliminates most of the advantages discussed above that Newton's method has in approximating the distance measure. On the other hand it makes it possible again to employ FFT based solvers for the solution of the linear problem and also greatly simplifies any iterative solvers. The convergence rate of the non-linear scheme, however, can be expected to be degraded as the approximation of the distance measure is basically reduced to the same quality as in pure gradient descent. Naturally the real performance of such methods depends on the choice of ϵ which should reflect the scaling of the gradient of the distance measure. A straightforward choice for the sum of squared difference distance measure is to just use the arithmetic average over the squared main diagonal of the part of the Hessian that was used in (4.5).

$$\epsilon = \frac{\sum_{i=1}^{sd} \left(\nabla M_{\mathbf{u}}\right)_i^2}{sd} \tag{4.8}$$

Even with this very crude approximation the convergence of the algorithm is rather good, as shown in Section 4.7.

4.4 Quasi-Newton

So far all the Newton type methods presented were restricted to sum of squared differences as distance measure. This is due to the complicated nature of analytically calculating higher order derivatives for mutual information. As an alternative it is possible to try to approximate $\mathbf{H}_{\mathcal{D}_{\mathrm{MI}}(F,M_{\mathbf{u}})}$ numerically. Newton-type algorithms that work with numerically approximated Hessians are generally known as Quasi-Newton methods. Quasi-Newton methods are essentially the multidimensional extension of the secant method which replaces the gradient i. e. the tangent of the function by a secant through the function values of the last steps. Quasi-Newton methods are derived from the so-called secant condition, which requires the current approximation of the Hessian $\mathbf{H}_{i+1}$ to at least be valid for the last two steps of the optimization algorithm.

$$\begin{aligned} \mathbf{H}_{i+1}\mathbf{s}_i &= \mathbf{t}_i \\ \text{where } \mathbf{s}_i &= \mathbf{u}^{(t+1)} - \mathbf{u}^{(t)} \\ \mathbf{t}_i &= \nabla_{\mathbf{u}}\mathcal{D}(F, M_{\mathbf{u}^{(t+1)}}) - \nabla_{\mathbf{u}}\mathcal{D}(F, M_{\mathbf{u}^{(t)}}) \end{aligned} \tag{4.9}$$

The secant condition constitutes a linearization of the Hessian around the current values of $\mathbf{u}^{(t+1)}$ and $\mathbf{u}^{(t)}$. As a starting point, this can be used to determine a numerical value for ϵ from equation (4.7), by requiring that ϵ be a least-squares solution to (4.9).

$$\begin{aligned} \mathbf{H}_{i+1} &= \epsilon_{i+1}\mathbf{I} \\ \epsilon^*_{i+1} &= \underset{\epsilon_{i+1}}{\operatorname{argmin}}(\epsilon_{i+1}\mathbf{s}_i - \mathbf{t}_i)^2 \\ \epsilon^*_{i+1} &= \frac{\mathbf{s}_i^T\mathbf{t}_i}{\mathbf{s}_i^T\mathbf{s}_i} \end{aligned} \tag{4.10}$$

Interestingly this is essentially a finite difference scheme projected down onto $\mathbf{s}_i$.

There are two immediate problems with this approach. The first is how to secure the positive definiteness of $\mathbf{H}_{i+1}$ i. e. $\epsilon > 0$. In order for $\epsilon > 0$ it has to hold that $\mathbf{s}_i^T\mathbf{t}_i > 0$. This could for instance be ensured in the line search by employing the strong Wolfe line search condition described in Section 4.5. However, the Wolfe conditions add quite some computational cost. Other more heuristic ways to deal with this problem are to just ignore a negative value for ϵ_{i+1} and continue with the last valid estimate or to use its absolute value. In our implementation, the second alternative was chosen, along with a lower threshold for ϵ in order to prevent it from getting too close to 0. The second problem is that the formula requires an "old" $\mathbf{u}^{(t)}$ and a "new" $\mathbf{u}^{(t+1)}$ position to compute ϵ_{i+1}. In other words, it is assumed that a step has already been performed.

In most Quasi-Newton algorithms this first step is simply a gradient descent step i. e. $\mathbf{H}_1 = \mathbf{I}$. In the case of registration, this is not a good idea due to the bad performance of the gradient descent. A better way is to perform a single step of the semi-implicit gradient descent scheme. In this case a small initial step size for this scheme is sufficient as only any kind of first step is needed.

Thus far we started out with an algorithm that uses a rather good approximation of the Hessian of the sum of squared differences distance measure and progressively simplified it. The advantage of a simple representation of the Hessian is that the linear solver becomes simpler and the calculation of the Hessian can be done numerically. During the simplification the second order accuracy for the regularizer is retained. The regularizer is therefore much less of a problem. What is sacrificed is the accuracy in the distance measure, which leads to a degradation in the convergence rate of the non-linear iteration.

In order to recover some of the information that the Hessian provides, we now employ numerical approximations similar to what is used in (4.10) to determine the factor ϵ. In Quasi-Newton methods, this is done by updating a current approximation of the Hessian with low rank matrix updates which are numerically calculated from the last iteration steps. In this work, we will only consider the scheme named after Broyden, Fletcher, Goldfarb and Shanno or BFGS for short (see e. g. [Noce 99] for a more in detail explanation of the BFGS and related Quasi-Newton schemes). The BFGS formula is based on updating, not the Hessian $\mathbf{H}$, but instead the inverse of the Hessian $\mathbf{B} = \mathbf{H}^{-1}$. In Newton's algorithm, this eliminates the need to solve for the updated matrix. The actual update rule can be derived from the inverse secant condition (compare (4.9))

$$\mathbf{B}_{i+1}\mathbf{t}_i = \mathbf{s}_i. \tag{4.11}$$

In addition to the secant condition Quasi-Newton methods employ additional constraints to arrive at the actual update rule. These constraints comprise a low rank update (rank 1 or 2), the preservation of the symmetry of the Hessian and the requirement for the update to be minimal in some sense. Different choices of these constraints, for example for the minimality of the update will lead to different methods. In the case of the BFGS update rule a symmetric rank 2 update of the form

$$\mathbf{B}_{i+1} - \mathbf{B}_i = \mathbf{a}\mathbf{a}^T + \mathbf{b}\mathbf{b}^T \tag{4.12}$$

is chosen, which ensures the preservation of the symmetry of the Hessian by construction. The unknowns $\mathbf{a}$ and $\mathbf{b}$ can be derived from the inverse secant condition (4.11) and from the requirement of a minimal update i. e. that

$$\mathbf{B}_{i+1} = \underset{\mathbf{B}}{\operatorname{argmin}} \|\mathbf{B} - \mathbf{B}_i\|_{\mathrm{WF}} \tag{4.13}$$

where $\|.\|_{\mathrm{WF}}$ denotes a weighted Frobenius norm. For details about the derivation of the update rule please refer to [Noce 99]. The update rule resulting from the application of these requirements is then

$$\begin{aligned} \mathbf{B}_{i+1} &= \mathbf{V}_i^T \mathbf{B}_i \mathbf{V}_i + \rho_i \mathbf{s}_i \mathbf{s}_i^T \\ \text{where} \quad \rho_i &= \frac{1}{\mathbf{t}_i^T \mathbf{s}_i} \\ \mathbf{V}_i &= \mathbf{I} - \rho_i \mathbf{t}_i \mathbf{s}_i^T. \end{aligned} \tag{4.14}$$

The matrix $\mathbf{B}_{i+1}$ is obviously not sparse, which would be a problem in a non-rigid registration application where the full size of the matrix can be huge. The solution to this is to not actually calculate the new matrix $\mathbf{B}_{i+1}$ but instead to store the vectors $\mathbf{s}_i$ and $\mathbf{t}_i$ and perform the update on the fly during the multiplication with $\mathbf{B}_{i+1}$. As it is also impractical to keep all the updates accumulated over the course of an optimization in memory, only a limited number is actually retained. This type of update scheme is therefore called limited memory BFGS or L-BFGS, as introduced by Nocedal in [Noce 80]. Nocedal also describes an efficient recursive evaluation scheme for (4.14) that is originally due to Matthies and Strang [Matt 79]. The BFGS update suffers from similar problems as the numerical schemes used for the calculation of the scaling factor ϵ. The vectors $\mathbf{t}$ and $\mathbf{s}$ have to fulfill certain conditions in order to keep $\mathbf{B}_{i+1}$ positive definite. Similarly, to the numeric calculation of ϵ_{i+1} for the inexact Newton method, these conditions can be ensured by employing the strong Wolfe conditions in the line search. As an alternative, if we do not want to perform these computationally expensive line search checks, we opted to simply skip a BFGS update for which the conditions are not met.

4.5 Automatic Step Size Control

A rather important aspect in the application of optimization algorithms is the step size control that automatically determines the length of the step τ that the algorithm takes. In algorithms like the explicit gradient descent or the Newton type methods this is also known as line search, as the step size τ determines how far along the computed descent direction the algorithm advances. In the semi-implicit gradient descent formulation, however, the step size is contained within the linear problem that has to be solved, which leads to a search on a curve rather than on a line. This has the disadvantage that for every choice of τ the algorithm tries, the linear problem has to be solved again. This check, the line search condition, makes sure that there is a “sufficient” decrease of the energy relative to the given step size τ. The most basic line search condition that can be employed, is to simply check whether there is any improvement at all, i. e.

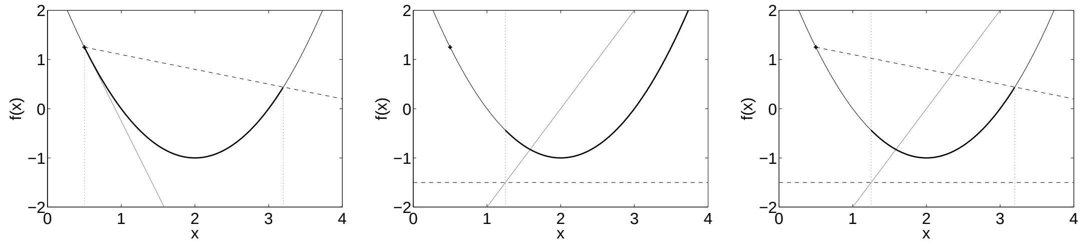
$$\mathcal{E}(F, M, \mathbf{u}^{(t+1)}) < \mathcal{E}(F, M, \mathbf{u}^{(t)}). \tag{4.15}$$

This is the least that any algorithm should check during iteration. The Armijo line search condition is an extension to this that ensures that the decrease in the energy is sufficient related to the length of the step taken. If the decrease in energy is rather small, shorter steps are favored. Let $\mathbf{d}$ denote the descent direction and $c_1 \in]0, 1[$ a user set parameter that controls how strictly the check for a “sufficient” decrease is performed, then the Armijo line search condition is defined as

$$\begin{aligned} \mathcal{E}(F, M, \mathbf{u}^{(t+1)}) &= \mathcal{E}(F, M, \mathbf{u}^{(t)} + \tau\mathbf{d}) \\ &\leq \mathcal{E}(F, M, \mathbf{u}^{(t)}) + c_1\tau\mathbf{d}^T\nabla_{\mathbf{u}}\mathcal{E}(F, M, \mathbf{u}^{(t)}). \end{aligned} \tag{4.16}$$

The application of this formula is illustrated in Figure 4.6a. As in the semi-implicit gradient descent there is no real descent direction we define that $\mathbf{d} := \mathbf{u}^{(t+1)} - \mathbf{u}^{(t)}$ with an artificial step length $\tau = 1$ for the check of the condition. Equation (4.16) is just the first order Taylor expansion around $\mathcal{E}(F, M, \mathbf{u}^{(t)} + \tau\mathbf{d})$ with respect to τ. The parameter c_1 that is added into this Taylor expansion reduces the slope of the linear approximation, thus relaxing the constraint. As, for a sufficiently smooth function, the Taylor expansion becomes more and more accurate as the step size τ is reduced, it has to be possible to fulfill (4.16) for a sufficiently small τ. The smaller c_1 is chosen, the closer this condition becomes to (4.15). In practice rather small values are used e. g. $c_1 \approx 10^{-4}$ (compare [Noce 99]), in order to not get stuck in the line search too long or generate unnecessarily short steps.

The Armijo condition can always be fulfilled with a sufficiently small step size, but for the algorithm to actually advance it should not be chosen too small. One way to



(a) Armijo condition (b) Curvature condition (c) Wolfe conditions

Figure 4.6: All figures show a shifted parabola $f(x) = (x-2)^2 - 1$ as the function to be optimized (black curve). The current position $x = 0.5$ is marked by a black '+', the fat black line identifies the points on the function accepted by the line search condition, the dotted black lines highlight the accepted region boundaries.
(a) Armijo condition; blue line: tangent; red line "tangent" with scaled slope (scaled by $c_1 = 0.1$) defining the Armijo condition (everything below is accepted)
(b) curvature condition; any point of ∇f (blue curve) above the red line depicting $c_2 \nabla f(x)$ is accepted ($c_2 = 0.5$)
(c) both conditions applied in combination (Wolfe conditions)

achieve this is to use a backtracking line search. A backtracking line search starts with a relatively large step size and successively reduces it by a fixed factor $0 < \gamma < 1$ until the line search condition is satisfied.

```
Require: $\tau > 0$;   $0 < \gamma < 1$
while not Armijo do
    $\tau = \gamma\tau$
end while
return $\tau$
```

In all the applications presented in this work the reduction factor is always chosen as $\gamma = 0.5$.

As any small enough step will satisfy the Armijo condition one can add to this a constraint that ensures a sufficiently large step size in order to speed up the convergence. The following constraint with the constant $c_2 \in]0, 1[$ is called the curvature condition (see also Figure (b)).

$$\mathbf{d}^T \nabla_{\mathbf{u}} \mathcal{E}(F, M, \mathbf{u}^{(t)} + \tau \mathbf{d}) \geq c_2 \mathbf{d}^T \nabla_{\mathbf{u}} \mathcal{E}(F, M, \mathbf{u}^{(t)}). \tag{4.17}$$

In contrast to (4.15) and (4.16) this condition ensures that the step taken is not too small. If used together with (4.16) and $0 < c_1 < c_2 < 1$ these two conditions are known as the Wolfe conditions. This condition compares the derivatives of the target function with respect to the step size. The right hand side is, for an actual descent direction $\mathbf{d}$ always negative. If the left hand side of the condition is "more negative" than that, and the condition therefore not fulfilled, it is an indication that $\mathbf{d}$ is still a good descent direction at the new position $\mathbf{u}^{(t)} + \tau\mathbf{d}$ and a longer step should be taken. If it is less negative, or even positive $\mathbf{d}$ is no longer a good descent direction and the line search can be stopped. The condition can be made more strict by comparing the absolute curvature values instead.

$$|\mathbf{d}^T \nabla_{\mathbf{u}} \mathcal{E}(F, M, \mathbf{u}^{(t)} + \tau \mathbf{d})| \leq c_2 |\mathbf{d}^T \nabla_{\mathbf{u}} \mathcal{E}(F, M, \mathbf{u}^{(t)})|. \tag{4.18}$$

The main difference to (4.17) is that in this formulation, too strongly positive values of $\mathbf{d}^T \nabla_{\mathbf{u}} \mathcal{E}(F, M, \mathbf{u}^{(t)} + \tau \mathbf{d})$ are also rejected. Together with the Armijo condition (4.16) this is known as the strong Wolfe conditions. If this condition fails the step size has to be increased. It is therefore necessary to employ a different line search strategy that allows for increasing and decreasing of the step size. In this work we used a strategy based on the bisection method.

```
Require: $\tau > 0$
Require: $l == 0$;   $u == 0$;   $b ==$ false
while not $b$ do
    if not Armijo then
        $u = \tau$
        $\tau = 0.5(l + \tau)$
    else if not Curvature then
        if u == 0 then
            $\tau = 2\tau$
        else
            $l = \tau$
            $\tau = 0.5(u + \tau)$
        end if
    else
        $b =$ true
    end if
end while
return $\tau$
```

Even though it is guaranteed that there is a value for the step size satisfying the Wolfe conditions, we chose to terminate the `while`-loop after a maximum of 5 iterations, if at least the Armijo condition is satisfied. This is done to safe computations and as a safeguard in situations where numerical errors in the evaluation of the measures can lead to difficulties in satisfying the conditions. A disadvantage of the Wolfe conditions are the added computations for evaluating the energy gradient at the new positions $\mathbf{u}^{(t)} + \tau \mathbf{d}$. These evaluations have to be performed for every step size candidate. In exchange the Wolfe conditions should provide the best estimate and additionally ensure in combination with the BFGS method that the approximated Hessian stays positive definite.

The last thing that has to be determined is, how the line search is initialized in the first iteration of the optimization algorithm. In the case of the Newton based methods, an initial step size of $\tau = 1$ is a possible choice, as that corresponds to the optimum step size if the quadratic approximation on which Newton's method relies, is correct. A better approach in practice is to specify the initial step size depending on the length of the initial step. Especially when using a backtracking line search the initial step size has to be chosen large enough or the iteration progress will be small. If it is chosen too large such that considerable parts of the images are not overlaid anymore, the registration will fail completely. For calculating a good initial step size, we choose a sensible length r, like 5% of the shortest edge of the image domain Ω as a reference length for the desired average deformation in each pixel. The initial step size is then computed as

$$\tau_{\text{init}} := r \, \frac{\sum_{i=1}^{s} \|\nabla_{\mathbf{u}} \mathcal{E}(F, M, \mathbf{u})(\mathbf{x}_i)\|}{s}. \tag{4.19}$$

In most experiments, this has proven to be a good initial guess for the backtracking line search. In all successive iterations, the step size of the last successful step is used.

In the case of the semi-implicit gradient descent, the gradient of the distance measure is used instead of the descent direction. This is based on the assumption that the solution to the linear system $\mathbf{I} + \tau\alpha\mathbf{A}$ acts only as a smoothing and does not change the overall length of the gradient vector too much.

4.6 Multilevel

In addition to the various choices of linear and non-linear solvers and line search methods, a multi-level continuation can be added to speed up the optimization process and to alleviate problems with local minima. The basic idea of a multi-level approach is to downsample the input images to a lower resolution, solve the registration problem there and upsample the result again. The upsampled result can then be used as an initial guess for the original registration problem and thus be refined. The optimization of the low resolution registration problem has much lower computational cost than the original problem. Additionally, if a downsampling scheme that incorporates some smoothing is employed, high frequency structures will disappear in the low resolution images, which will lead to fewer local minima in the optimization.

This approach can be iterated, thus creating a pyramid of levels of increasingly lower resolution. Generally a downsampling by a factor of 2 along each dimension is often employed in practice, and is also the choice in each application of a multi-level scheme in this work. In a 2-D application, the number of pixels in the lower resolution problem and therefore the computational cost decreases by a factor of 4, in 3-D it decreases by a factor of 8. It can therefore be beneficial to perform more iterations on the lower levels than on the finer ones, as the cost is low and the gain might be a reduced number of iterations on the next finer level. On the other hand, the low resolution images can never fully capture the original problem. Most importantly, during the downsampling there is usually some averaging of pixel values performed. This leads to gray values in the low resolution image that were not present in the original images. This can impair the registration result on the low resolution images, especially when statistical measures like the mutual information are used. Due to this, the effectiveness of the multi-level technique can vary depending on the image content.

4.7 Comparison of Optimization Methods

The presented optimization techniques and choices for the line search will be evaluated in the following on a set of 2-D example datasets. The evaluated optimization techniques and their respective designations are

- semi-implicit – semi-implicit gradient descent (4.2)
- Newton secant – Newton's method (4.3) with $\mathbf{H}_{\mathcal{D}(F,M_{\mathbf{u}(t)})}$ estimated according to (4.7) and (4.10)
- Newton SSD average – Newton's method (4.3) with $\mathbf{H}_{\mathcal{D}(F,M_{\mathbf{u}(t)})}$ estimated for the sum of squared differences (SSD) distance measure according to (4.7) and (4.8)
- Newton SSD diagonal – Newton's method (4.3) with $\mathbf{H}_{\mathcal{D}(F,M_{\mathbf{u}(t)})}$ estimated for the sum of squared differences (SSD) distance measure according to (4.6)
- L-BFGS secant – limited memory BFGS method (4.3)(4.14) with $\mathbf{B}_0$ estimated according to (4.7) and (4.10) (similar to „Newton secant")

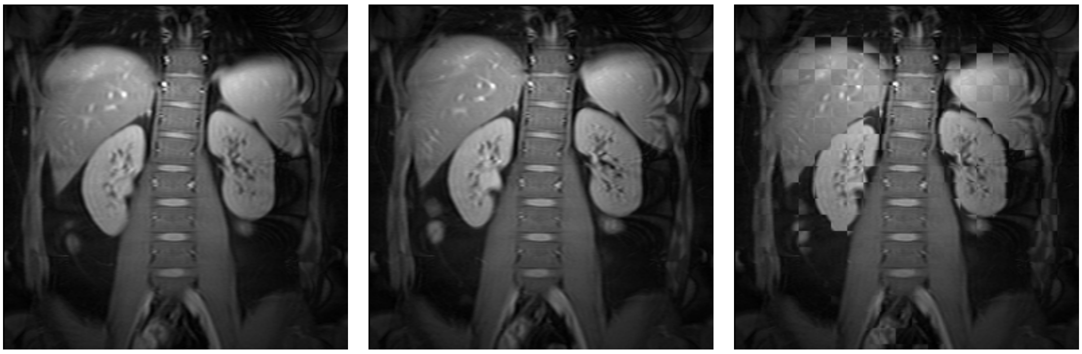
For the line search the used conditions and names are

- descent – simple condition checking if the measure improved at all (4.15)
- Armijo – Armijo rule (4.16) with $c_1 = 0.1$
- Wolfe – Wolfe rules consisting of the Armijo rule (4.16) with $c_1 = 0.1$ and the strong curvature condition (4.18) with $c_2 = 0.9$

The values chosen for the line search parameters c_1 and c_2 have been experimentally determined, but are also in agreement with the values recommended by Nocedal [Noce 99]. For the Armijo condition parameter c_1, which is chosen relatively high, lower values lead to results that are pretty much identical to the „descent" condition and even with this value the differences are slim. In some cases the results for the line search condition "descent" is therefore omitted to reduce clutter in the plots. As the semi-implicit gradient descent method does not really allow to search on a line, the application of the Wolfe rules for this optimization algorithm is questionable and therefore omitted.

The methods are compared in their respective single level convergence. Additionally the final result of a single and a multi-level application employing the respective optimization methods is also presented. For the single level applications, 50 iterations are performed. In the multi-level case, 10 iterations are performed on the finest level and the maximum number of iterations is doubled for each step down in the multi-level pyramid. Finally plots are provided that allow a comparison of the influence of the different line search conditions on the optimization algorithms. The L-BFGS method is run in all examples with 5 updates to the estimated Hessian matrix, as a good compromise of convergence improvement and memory consumption.

The example datasets are introduced in the following. All of them are taken from real world applications that make use of the registration algorithms implemented as a part of this work.



(a) Fixed image F (b) Moving image M (c) Checkerboard overlay

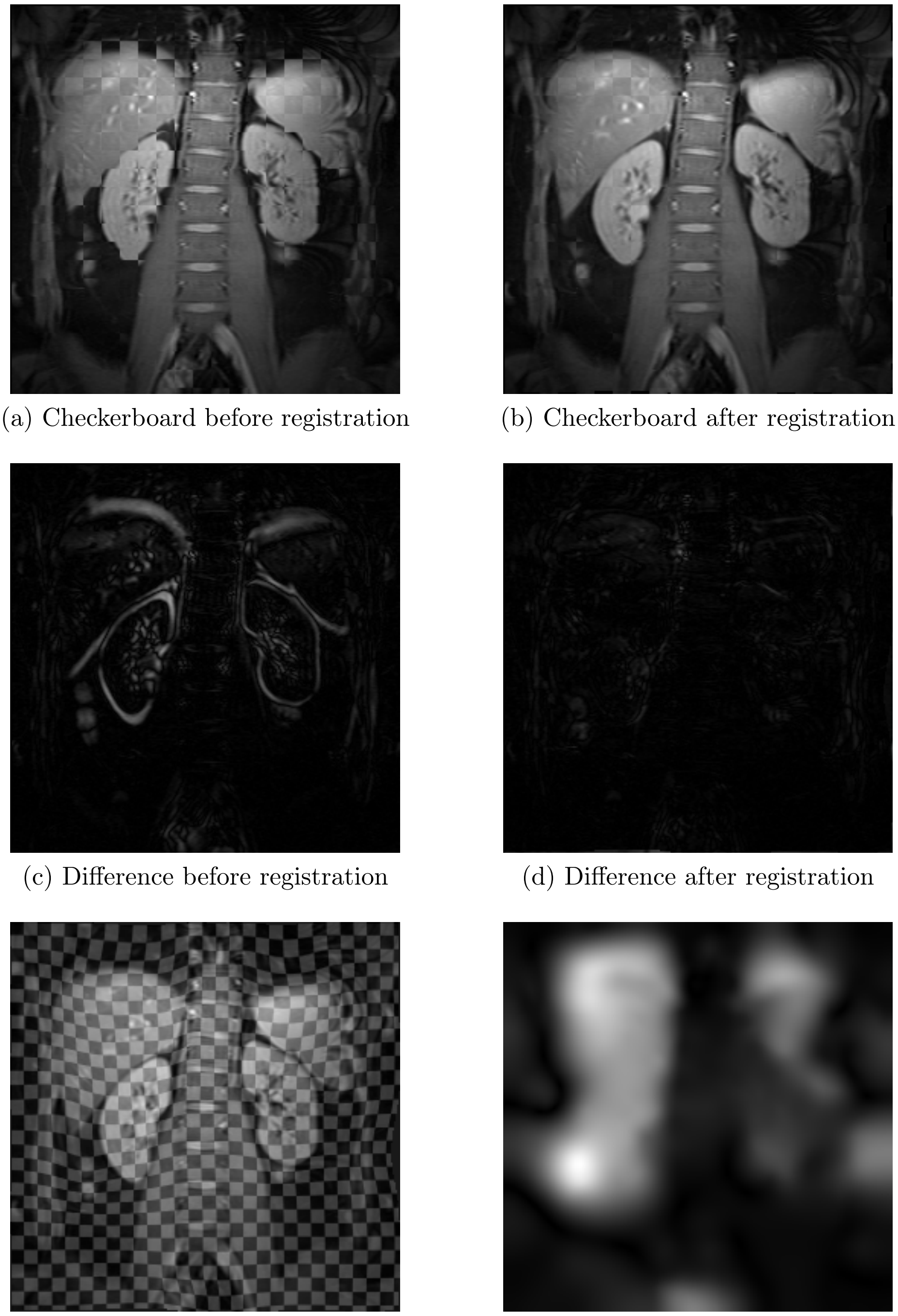
Figure 4.7: The kidney perfusion dataset fixed image F, moving image M and a checkerboard overlay of both, showing the initial mismatch.

4.7.1 Kidney Perfusion

The first application example is a mono-modal registration of abdominal MR images. The aim of this application is to calculate the perfusion in the kidneys from sequentially acquired MR images without the use of contrast agents (compare [Ritt 10]). If the kidneys are correctly registered, the perfusion can be computed by comparison of the pixel gray values in two successive images. As the images are acquired in a relatively short time period no rigid pre-registration is necessary and the images can be regarded as mono-modal. We therefore chose the sum of squared differences distance measure (see Section 3.3.1) in combination with the curvature regularizer (see Section 3.4.2) for this application case. The images have a size of 256×256 with a pixel spacing of $\mathbf{h} = (1.4, 1.4)$ mm and gray values in the range [0 550] stored as unsigned short, 12-bit dicom data. The used stiffness parameter is $\kappa = 20$ and the multi-level registrations use 4 levels. The dataset and an example result are depicted in Figure 4.7 and 4.8 respectively.

As this a mono-modal registration problem, the used distance measure is the sum of squared differences. This application example allows the comparison of all the different presented methods for estimating $\mathbf{H}_{\mathcal{D}(F,M_{\mathbf{u}(t)})}$. These are tested in conjunction with the inexact Newton's method. The L-BFGS method is only applied with the numerical estimator based on the secant condition that is also applicable in a multi-modal scenario. This allows the best comparison of the different estimators and how the L-BFGS method which is also applicable in a multi-modal scenario, compares to Newton's method employing the best, sum of squared differences specific estimate for the Hessian of the distance measure, “Newton SSD diagonal”.

The performance with respect to the different optimization algorithms (Figure 4.9) shows that, not surprisingly, the Newton method employing the most accurate estimate of the Hessian “Newton SSD diagonal” also exhibits the best performance. But it is encouraging to see that the exclusively numerical working L-BFGS method is about as good with the Wolfe line search conditions and even trumps the “Newton SSD diagonal” performance when both use the simpler Armijo line search condition. It is also visible from Figure 4.10 that “Newton SSD diagonal” is pretty much the only optimization algorithm in this application example that significantly benefits



(a) Checkerboard before registration

(b) Checkerboard after registration

(c) Difference before registration

(d) Difference after registration

(e) Deformed image with checkerboard

(f) Magnitude image of deformation

Figure 4.8: Sample results for the registration of the kidney perfusion dataset. The used optimization algorithm is the multi-level “Newton diag” with the Wolfe line search conditions. First row: checkerboard overlays of fixed and moving image (a) before and (b) after registration. Second row: difference images (c) before and (d) after registration. Last row: (e) moving image overlaid with a checkerboard image and deformed; (f) magnitude image of the deformation.

	single level	multilevel
initial	$9.242142 \cdot 10^2$	$9.242142 \cdot 10^2$
semi-implicit, descent	$2.621928 \cdot 10^2$	$1.949514 \cdot 10^2$
semi-implicit, armijo	$2.733949 \cdot 10^2$	$1.977111 \cdot 10^2$
Newton secant, descent	$2.383151 \cdot 10^2$	$1.885831 \cdot 10^2$
Newton secant, armijo	$2.425350 \cdot 10^2$	$1.907033 \cdot 10^2$
Newton secant, wolfe	$2.361874 \cdot 10^2$	$1.868363 \cdot 10^2$
Newton SSD average, descent	$2.398196 \cdot 10^2$	$1.897096 \cdot 10^2$
Newton SSD average, armijo	$2.398196 \cdot 10^2$	$1.906924 \cdot 10^2$
Newton SSD average, wolfe	$2.279642 \cdot 10^2$	$1.881680 \cdot 10^2$
Newton SSD diagonal, descent	$2.047015 \cdot 10^2$	$1.854769 \cdot 10^2$
Newton SSD diagonal, armijo	$2.047015 \cdot 10^2$	$1.854725 \cdot 10^2$
Newton SSD diagonal, wolfe	$1.869532 \cdot 10^2$	$1.851277 \cdot 10^2$
L-BFGS secant, descent	$1.926995 \cdot 10^2$	$1.864865 \cdot 10^2$
L-BFGS secant, armijo	$2.000681 \cdot 10^2$	$1.864668 \cdot 10^2$
L-BFGS secant, wolfe	$1.943582 \cdot 10^2$	$1.860350 \cdot 10^2$

Table 4.1: Final energies after optimization with the different algorithm and line search combinations, applied to the kidney perfusion dataset.

from the improved line search due to the application of the Wolfe conditions. In particular the “descent” line search condition yields no significantly different results from the “Armijo” condition. Finally the optimization results for all the applied methods in the single and the multi-level scheme are given in Table 4.1 and 4.2. It is clearly visible that all methods improve a lot when used in the multi-level framework. However L-BFGS seems to benefit the least of all the methods. The different types of line search seem to only have a rather marginal effect on the end result of a multi-level based optimization.

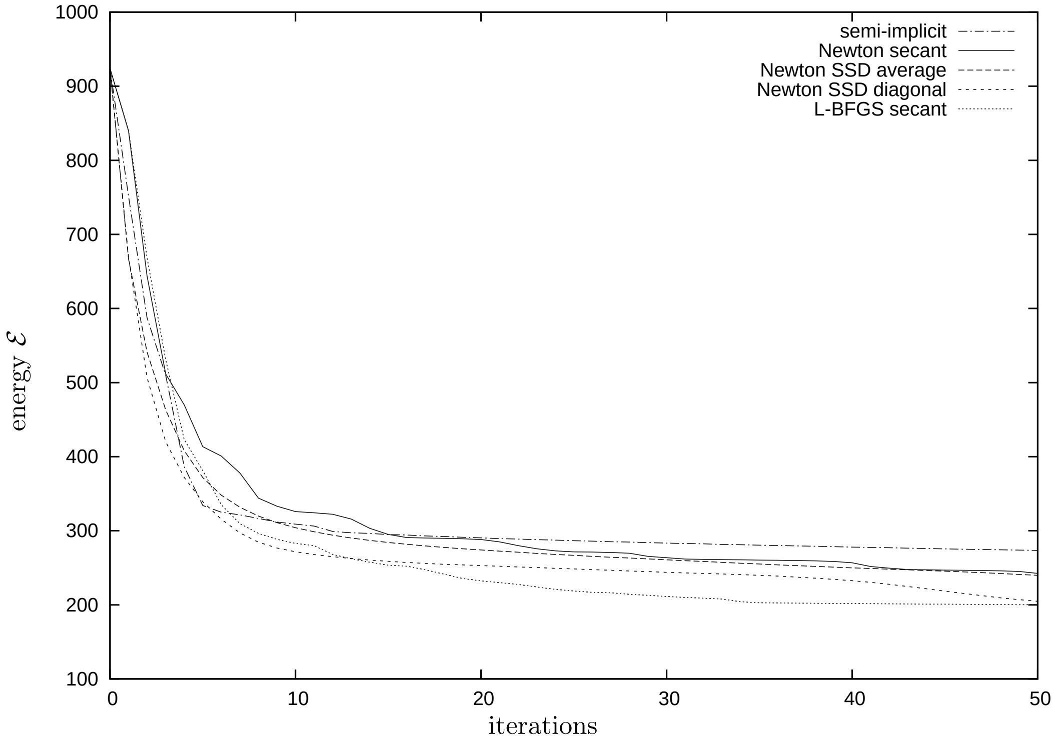
	single level	multilevel
initial	$9.242142 \cdot 10^2$	$9.242142 \cdot 10^2$
semi-implicit, descent	$2.241489 \cdot 10^2$	$1.662688 \cdot 10^2$
semi-implicit, armijo	$2.332077 \cdot 10^2$	$1.689916 \cdot 10^2$
Newton secant, descent	$2.047475 \cdot 10^2$	$1.599432 \cdot 10^2$
Newton secant, armijo	$2.086677 \cdot 10^2$	$1.625634 \cdot 10^2$
Newton secant, wolfe	$2.012050 \cdot 10^2$	$1.574586 \cdot 10^2$
Newton SSD average, descent	$2.066580 \cdot 10^2$	$1.614456 \cdot 10^2$
Newton SSD average, armijo	$2.066580 \cdot 10^2$	$1.623398 \cdot 10^2$
Newton SSD average, wolfe	$1.949253 \cdot 10^2$	$1.598922 \cdot 10^2$
Newton SSD diagonal, descent	$1.745574 \cdot 10^2$	$1.563084 \cdot 10^2$
Newton SSD diagonal, armijo	$1.745574 \cdot 10^2$	$1.562825 \cdot 10^2$
Newton SSD diagonal, wolfe	$1.578066 \cdot 10^2$	$1.557602 \cdot 10^2$
L-BFGS secant, descent	$1.637812 \cdot 10^2$	$1.573520 \cdot 10^2$
L-BFGS secant, armijo	$1.691247 \cdot 10^2$	$1.573811 \cdot 10^2$
L-BFGS secant, wolfe	$1.642830 \cdot 10^2$	$1.565525 \cdot 10^2$

(a) Distance

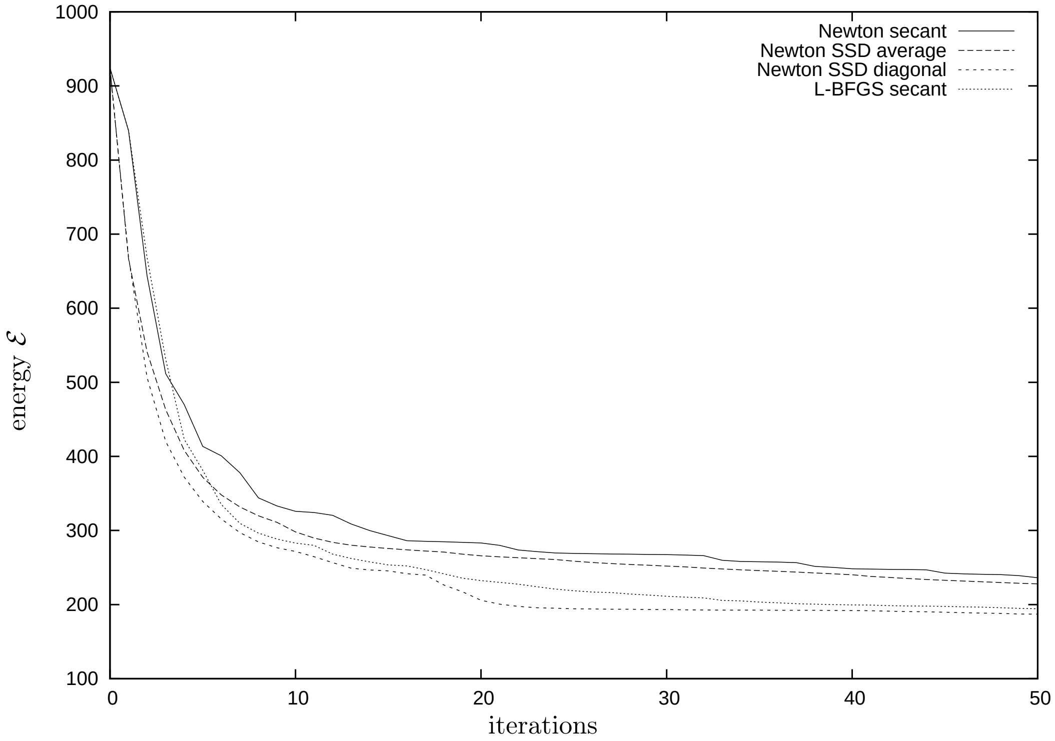
	single level	multilevel
initial	$1.135443 \cdot 10^2$	$1.135443 \cdot 10^2$
semi-implicit, descent	$3.804390 \cdot 10^1$	$2.868262 \cdot 10^1$
semi-implicit, armijo	$4.018726 \cdot 10^1$	$2.871942 \cdot 10^1$
Newton secant, descent	$3.356765 \cdot 10^1$	$2.863991 \cdot 10^1$
Newton secant, armijo	$3.386730 \cdot 10^1$	$2.813989 \cdot 10^1$
Newton secant, wolfe	$3.498234 \cdot 10^1$	$2.937774 \cdot 10^1$
Newton SSD average, descent	$3.316158 \cdot 10^1$	$2.826396 \cdot 10^1$
Newton SSD average, armijo	$3.316158 \cdot 10^1$	$2.835265 \cdot 10^1$
Newton SSD average, wolfe	$3.303892 \cdot 10^1$	$2.827580 \cdot 10^1$
Newton SSD diagonal, descent	$3.014413 \cdot 10^1$	$2.916844 \cdot 10^1$
Newton SSD diagonal, armijo	$3.014413 \cdot 10^1$	$2.919000 \cdot 10^1$
Newton SSD diagonal, wolfe	$2.914661 \cdot 10^1$	$2.936754 \cdot 10^1$
L-BFGS secant, descent	$2.891833 \cdot 10^1$	$2.913450 \cdot 10^1$
L-BFGS secant, armijo	$3.094344 \cdot 10^1$	$2.908572 \cdot 10^1$
L-BFGS secant, wolfe	$3.007523 \cdot 10^1$	$2.948253 \cdot 10^1$

(b) Regularizer

Table 4.2: Final energies after optimization with the different algorithm and line search combinations, applied to the kidney perfusion dataset.

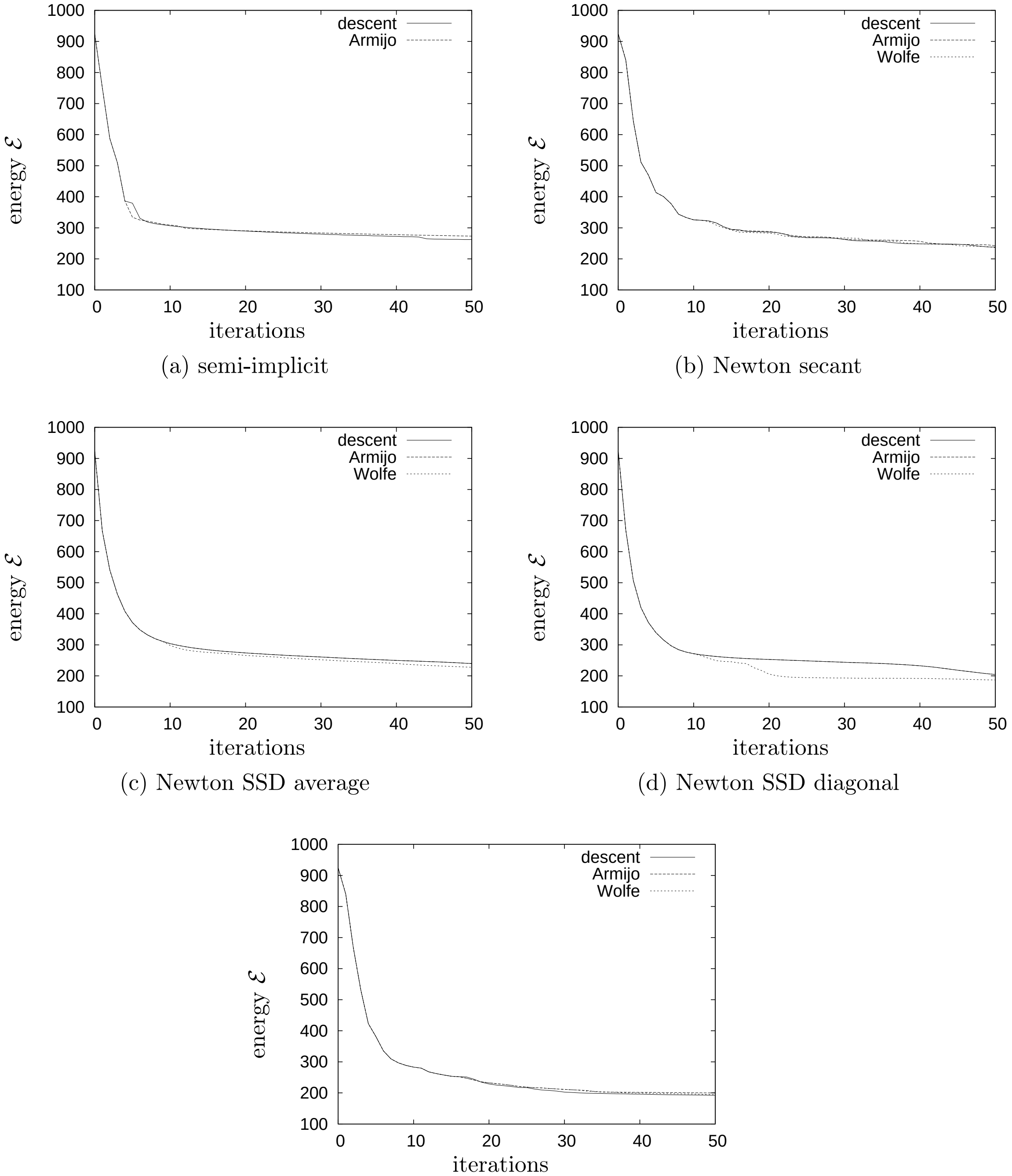


(a) Optimizer performances with Armijo type line search condition



(b) Optimizer performances with Wolfe type line search condition

Figure 4.9: Comparison of optimizer performances with Armijo and Wolfe type line search conditions. The “descent” line search condition is omitted as the results are almost identical to those of the Armijo conditions.



(a) semi-implicit

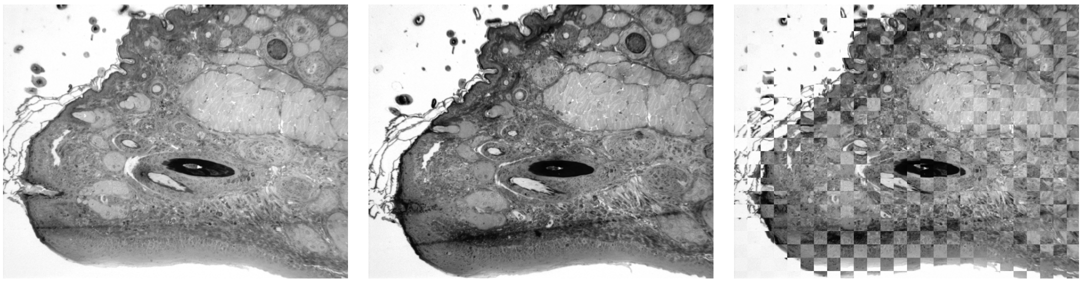
(b) Newton secant

(c) Newton SSD average

(d) Newton SSD diagonal

(e) L-BFGS secant

Figure 4.10: Comparison of the impact of the choice of line search condition on the different optimization algorithms, for the mono-modal kidney perfusion dataset. Except for the "Newton SSD diagonal" optimization algorithm (d) the different line search conditions show almost no effect. In all plots where the line of the descent condition is not visible it is identical to the Armijo.



(a) Fixed image F (b) Moving image M (c) Checkerboard overlay

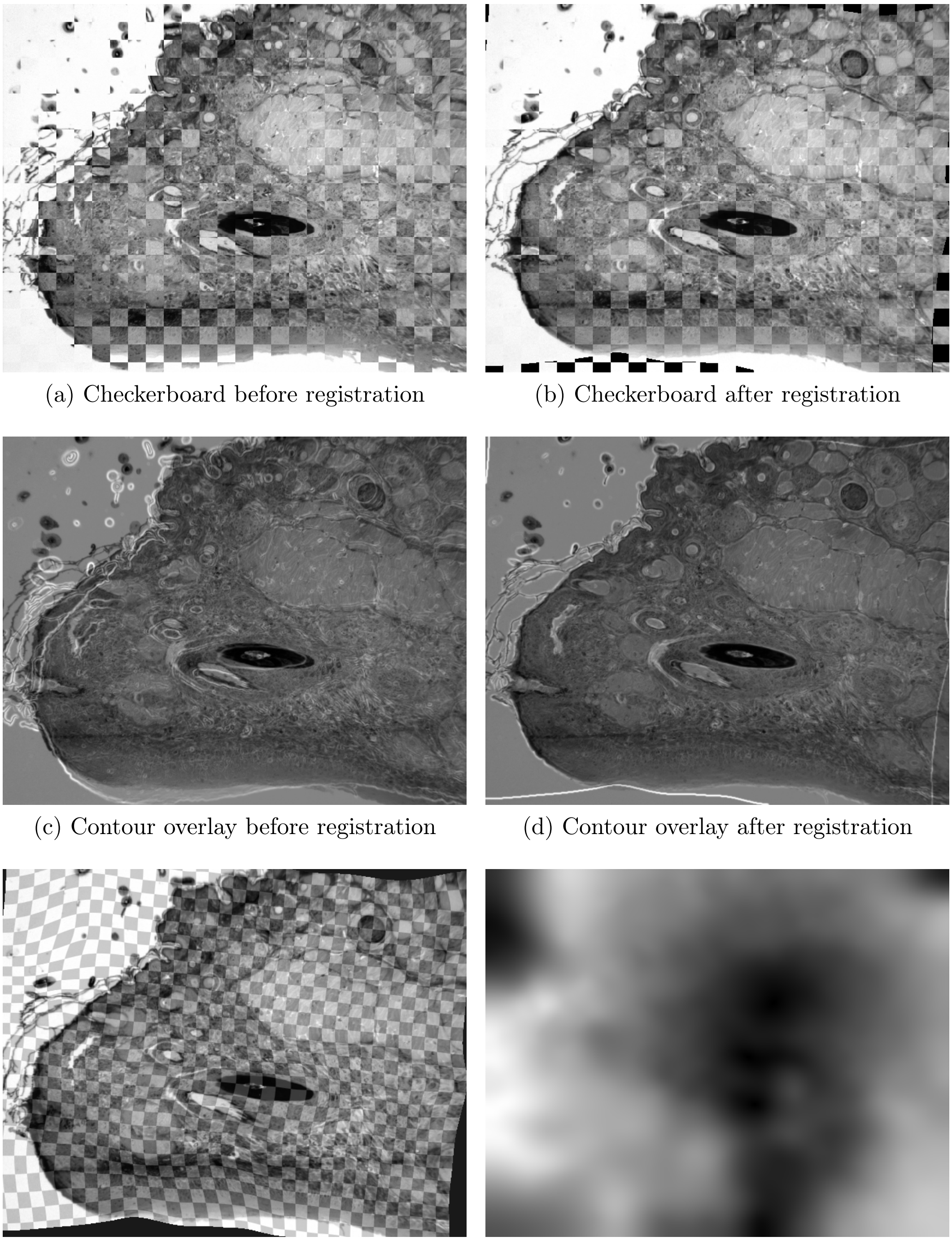
Figure 4.11: The hair follicle dataset.

4.7.2 Hair Follicle

The second application example we use to evaluate the optimization algorithms is the reconstruction of 3-D volume data from histological slices. Histological slices are produced by embedding a tissue sample, in this case a hair follicle, in a hard matrix, usually paraffin wax, and then cutting and photographing thin slices from it. However, the mechanical stress due to the cutting and the following staining of the tissue can introduce artifacts in the form of deformations and fissures. Some slices are also completely destroyed during this process. The registration algorithm is used in this application to interpolate lost slices by registering the two neighboring slices. The lost slice is then replaced by a "halfway" deformed neighbor i. e. only $0.5\boldsymbol{u}$ is applied. For more details please refer to Gaffling et al. [Gaff 09].

The staining process generally leads to considerable intensity differences between the images. Although an intensity standardization is employed in this application, there remain enough intensity differences between the images to make a multi-modal registration employing the mutual information as distance measure necessary. The two images used here (see Figure 4.11) have not been rigidly preregistered. They are reasonably well aligned, but, as the registration result (see Figure 4.12) shows, there is some rotational component in the solution. The images have a size of 648×514 and a gray value range of [0 255] i. e. 8-bit images. A spacing of $\mathbf{h} = (1, 1)$ is assumed as no specification for the image resolution is available. The algorithms used a stiffness for the registrations of $\kappa = 4$ and 5 levels for the multi-level optimization. The dataset and an example result is showcased in Figure 4.11 and 4.12 respectively.

The results depicted in Figure 4.13 and 4.14 show the L-BFGS algorithm with the Wolfe line search condition outperforming the semi-implicit time marching as well as the "Newton secant" method by a considerable amount. The L-BFGS algorithm also reacts in the expected way to the line search conditions (see Figure 4.15), with the Wolfe conditions yielding the fastest convergence, followed by the Armijo condition and the descent condition. The semi-implicit gradient descent on the other hand shows rather good results with the descent line search conditions but performs remarkably worse with the Armijo conditions. This bad performance in conjunction with the Armijo conditions might be related to the fact that reducing the step size in the semi-implicit method does not result in the search on a line, but rather on a non-linear curve instead, which is the same reason why the semi-implicit gradient descent is not run with the Wolfe conditions in these experiments. Finally, the "New-



(a) Checkerboard before registration

(b) Checkerboard after registration

(c) Contour overlay before registration

(d) Contour overlay after registration

(e) Deformed image with checkerboard

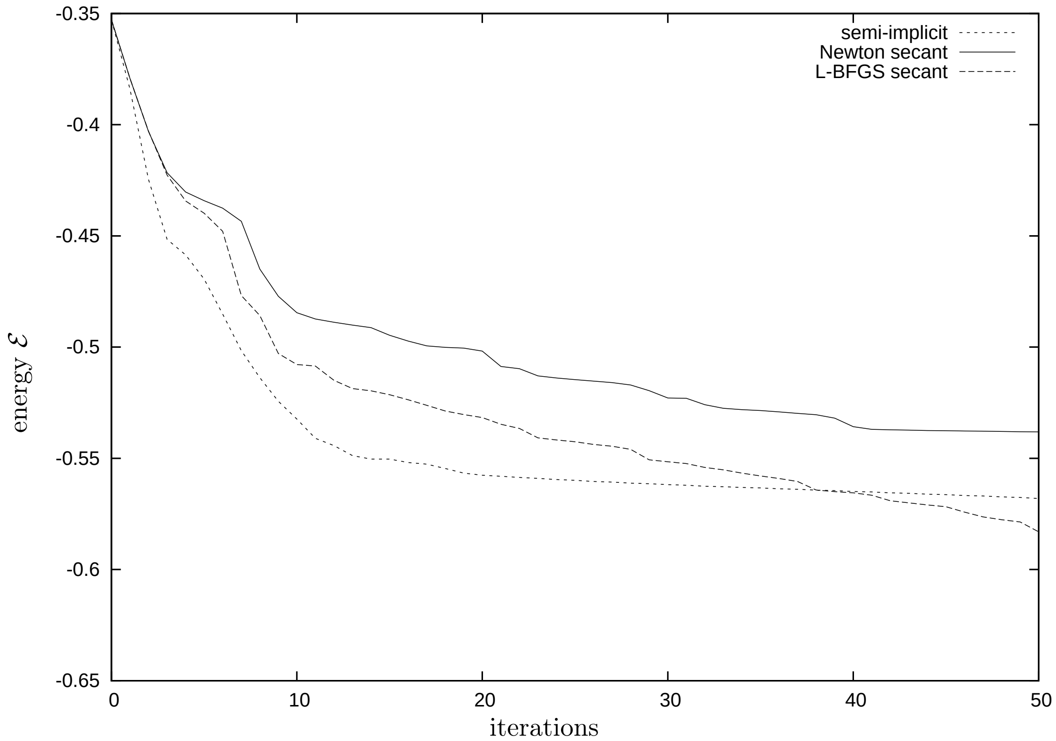
(f) Magnitude image of deformation

Figure 4.12: Sample results for the registration of the hair follicle dataset. The used optimization algorithm is the multi-level “L-BFGS secant” with the Wolfe line search conditions. First row: checkerboard overlays of fixed and moving image (a) before and (b) after registration. Second row: fixed image overlaid with contours of moving image (c) before and (d) after registration. Last row: (e) moving image overlaid with a checkerboard image and deformed; (f) magnitude image of the deformation.

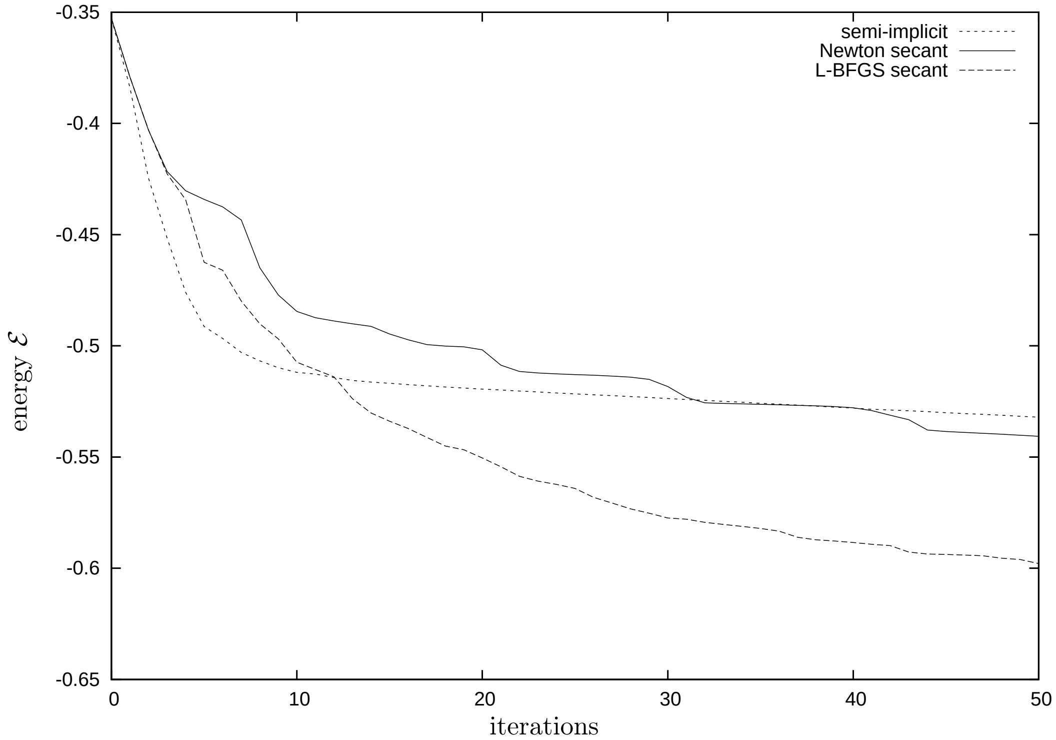
ton secant" method performs slightly worse than the semi-implicit gradient descent and does not seem to be significantly affected by the choice of the line search, even though the Wolfe line search conditions seem to have a slight edge on the others.

The numerical results in Table 4.3 again show large improvements in the multilevel application of the algorithms. The L-BFGS still performs best in the multilevel framework, but the "Newton secant" now outperforms the semi-implicit time marching. Furthermore the impact of the line search condition becomes relatively negligible, with the Wolfe conditions even performing worst of the three.

This dataset also illustrates nicely that the differences in the final registration energy are not only of numerical interest but actually result in significant differences in the final registration result as depicted in Figure 4.16.



(a) descent



(b) Armijo

Figure 4.13: Comparison of optimizer performances with descent and Armijo type line search conditions on the “hair follicle” dataset.

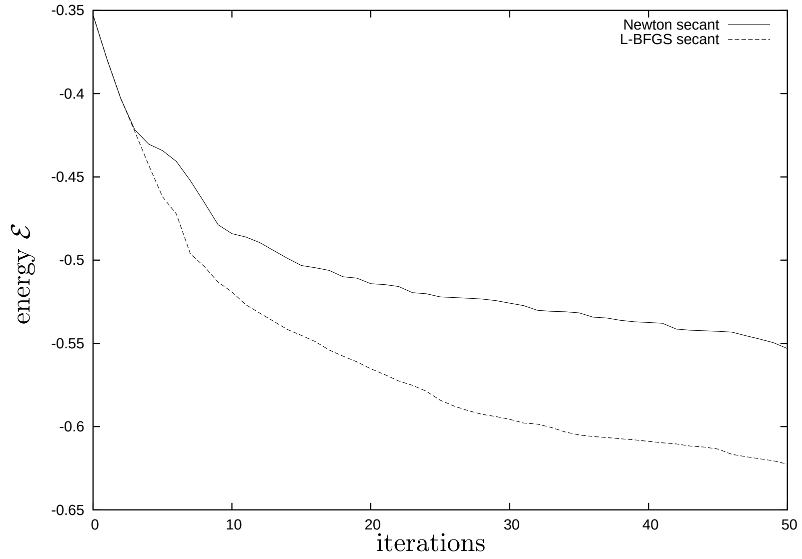
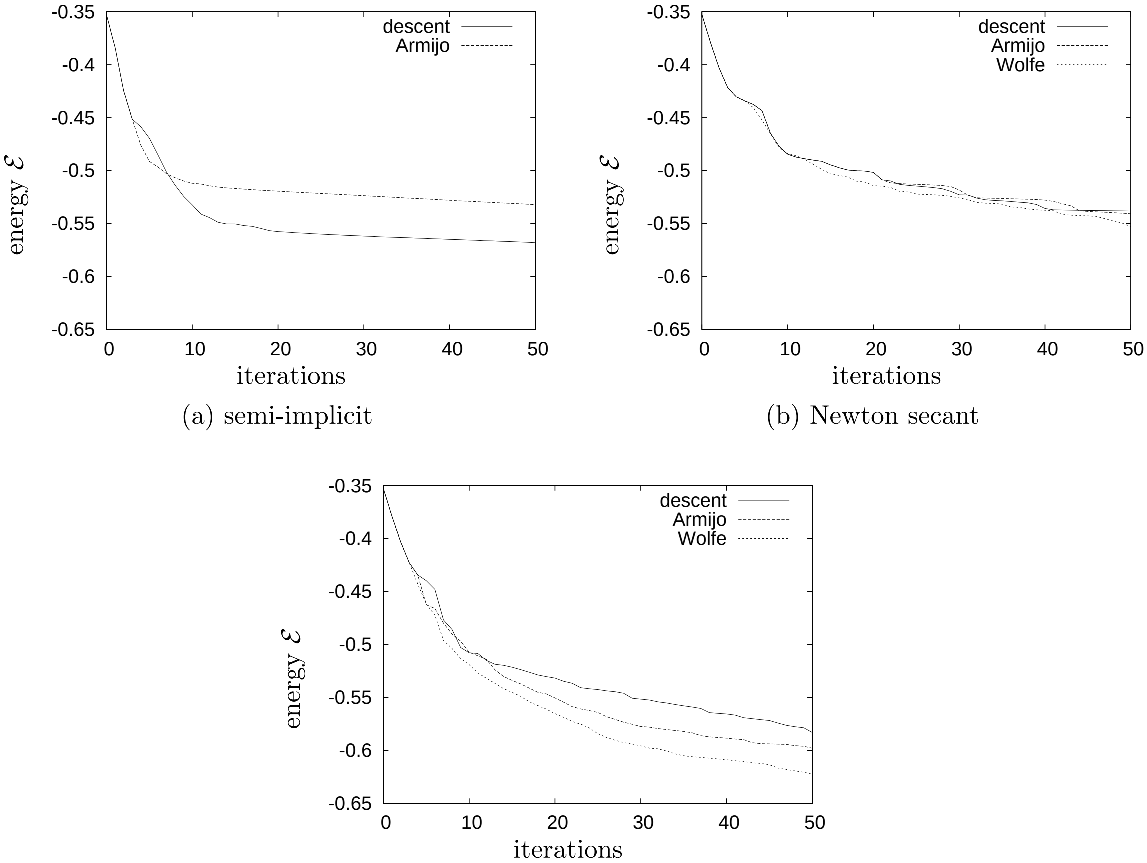


Figure 4.14: Comparison of optimizer performances with the Wolfe type line search conditions on the “hair follicle” dataset.



(a) semi-implicit

(b) Newton secant

(c) L-BFGS secant

Figure 4.15: Comparison of the impact of the choice of line search condition on the different optimization algorithms, for the “hair follicle” dataset.

	single level	multilevel
initial	$-3.527458 \cdot 10^{-1}$	$-3.527458 \cdot 10^{-1}$
semi-implicit, descent	$-5.680410 \cdot 10^{-1}$	$-6.725602 \cdot 10^{-1}$
semi-implicit, armijo	$-5.320870 \cdot 10^{-1}$	$-6.444729 \cdot 10^{-1}$
Newton secant, descent	$-5.381436 \cdot 10^{-1}$	$-6.903303 \cdot 10^{-1}$
Newton secant, armijo	$-5.406839 \cdot 10^{-1}$	$-6.865059 \cdot 10^{-1}$
Newton secant, wolfe	$-5.530918 \cdot 10^{-1}$	$-6.819689 \cdot 10^{-1}$
L-BFGS secant, descent	$-5.830720 \cdot 10^{-1}$	$-7.158433 \cdot 10^{-1}$
L-BFGS secant, armijo	$-5.979125 \cdot 10^{-1}$	$-7.160220 \cdot 10^{-1}$
L-BFGS secant, wolfe	$-6.225189 \cdot 10^{-1}$	$-7.119749 \cdot 10^{-1}$

(a) Energy

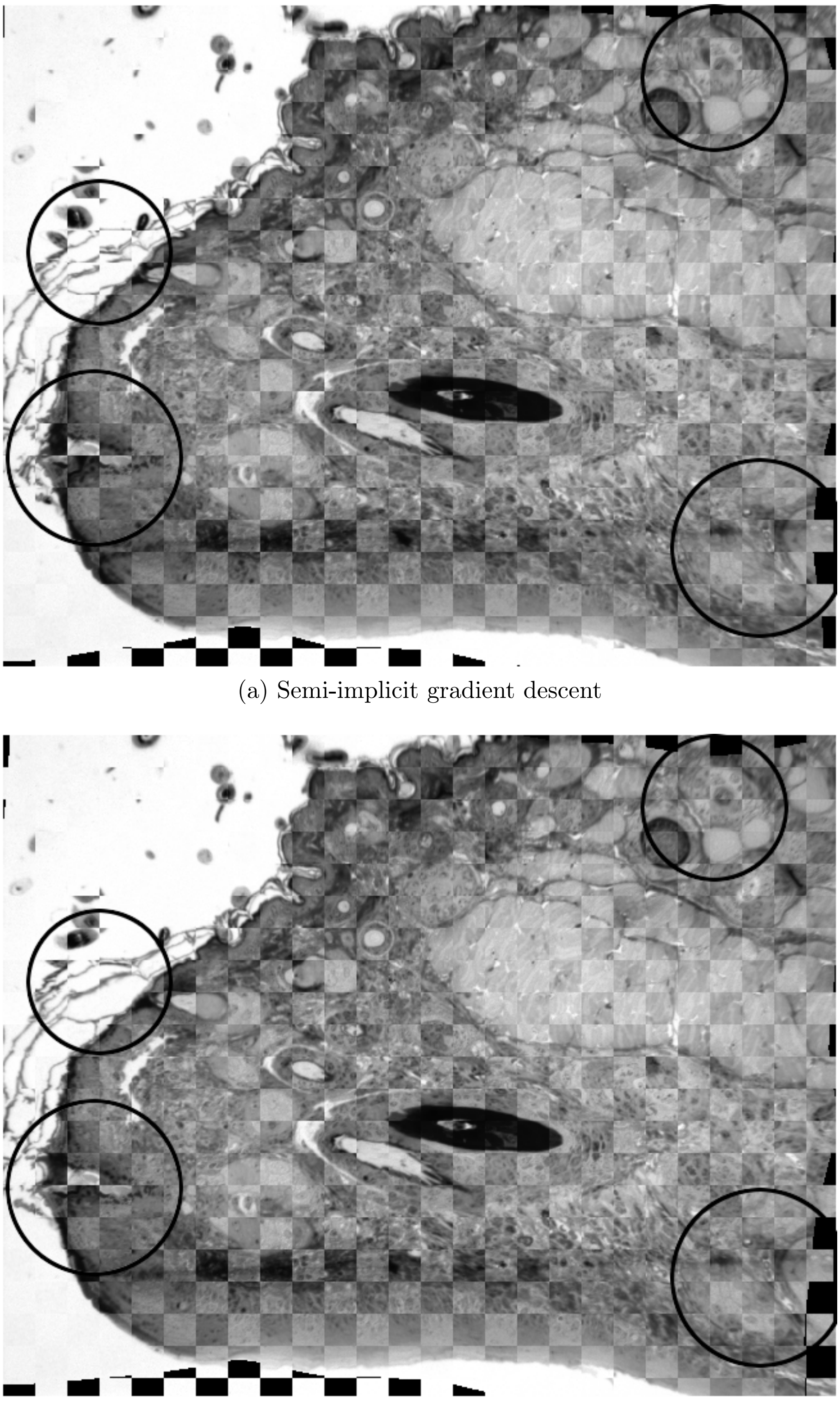
	single level	multilevel
initial	$-3.527458 \cdot 10^{-1}$	$-3.527458 \cdot 10^{-1}$
semi-implicit, descent	$-5.936443 \cdot 10^{-1}$	$-6.956034 \cdot 10^{-1}$
semi-implicit, armijo	$-5.556283 \cdot 10^{-1}$	$-6.688135 \cdot 10^{-1}$
Newton secant, descent	$-5.598435 \cdot 10^{-1}$	$-7.117882 \cdot 10^{-1}$
Newton secant, armijo	$-5.605795 \cdot 10^{-1}$	$-7.089288 \cdot 10^{-1}$
Newton secant, wolfe	$-5.771348 \cdot 10^{-1}$	$-7.064972 \cdot 10^{-1}$
L-BFGS secant, descent	$-6.165330 \cdot 10^{-1}$	$-7.381439 \cdot 10^{-1}$
L-BFGS secant, armijo	$-6.262602 \cdot 10^{-1}$	$-7.383496 \cdot 10^{-1}$
L-BFGS secant, wolfe	$-6.487738 \cdot 10^{-1}$	$-7.352672 \cdot 10^{-1}$

(b) Distance

	single level	multilevel
initial	$2.573186 \cdot 10^{-2}$	$2.573186 \cdot 10^{-2}$
semi-implicit, descent	$2.560326 \cdot 10^{-2}$	$2.304325 \cdot 10^{-2}$
semi-implicit, armijo	$2.354132 \cdot 10^{-2}$	$2.434056 \cdot 10^{-2}$
Newton secant, descent	$2.169989 \cdot 10^{-2}$	$2.145790 \cdot 10^{-2}$
Newton secant, armijo	$1.989564 \cdot 10^{-2}$	$2.242290 \cdot 10^{-2}$
Newton secant, wolfe	$2.404303 \cdot 10^{-2}$	$2.452834 \cdot 10^{-2}$
L-BFGS secant, descent	$3.346105 \cdot 10^{-2}$	$2.230057 \cdot 10^{-2}$
L-BFGS secant, armijo	$2.834769 \cdot 10^{-2}$	$2.232763 \cdot 10^{-2}$
L-BFGS secant, wolfe	$2.625494 \cdot 10^{-2}$	$2.329222 \cdot 10^{-2}$

(c) Regularizer

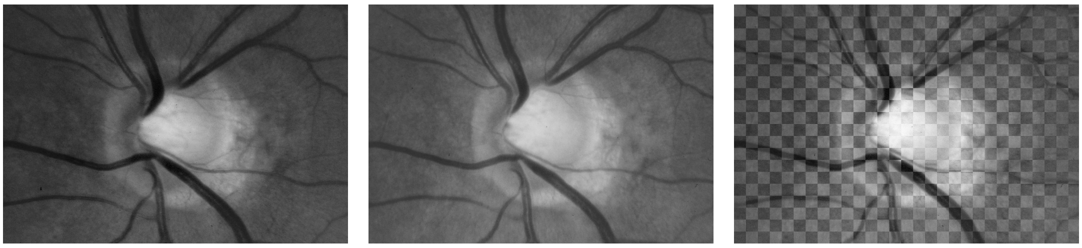
Table 4.3: Final energies after optimization with the different algorithm and line search combinations, applied to the “hair follicle” dataset.



(a) Semi-implicit gradient descent

(b) L-BFGS

Figure 4.16: Checkerboard images of the results of a multi-level registrations using the (a) semi-implicit gradient descent and the (b) L-BFGS method. Especially in the highlighted regions it becomes very visible that the numerical difference in the registration energies has a significant impact on the visual outcome of the registration.



(a) Fixed image F (b) Moving image M (c) Checkerboard overlay

Figure 4.17: The glaucoma dataset.

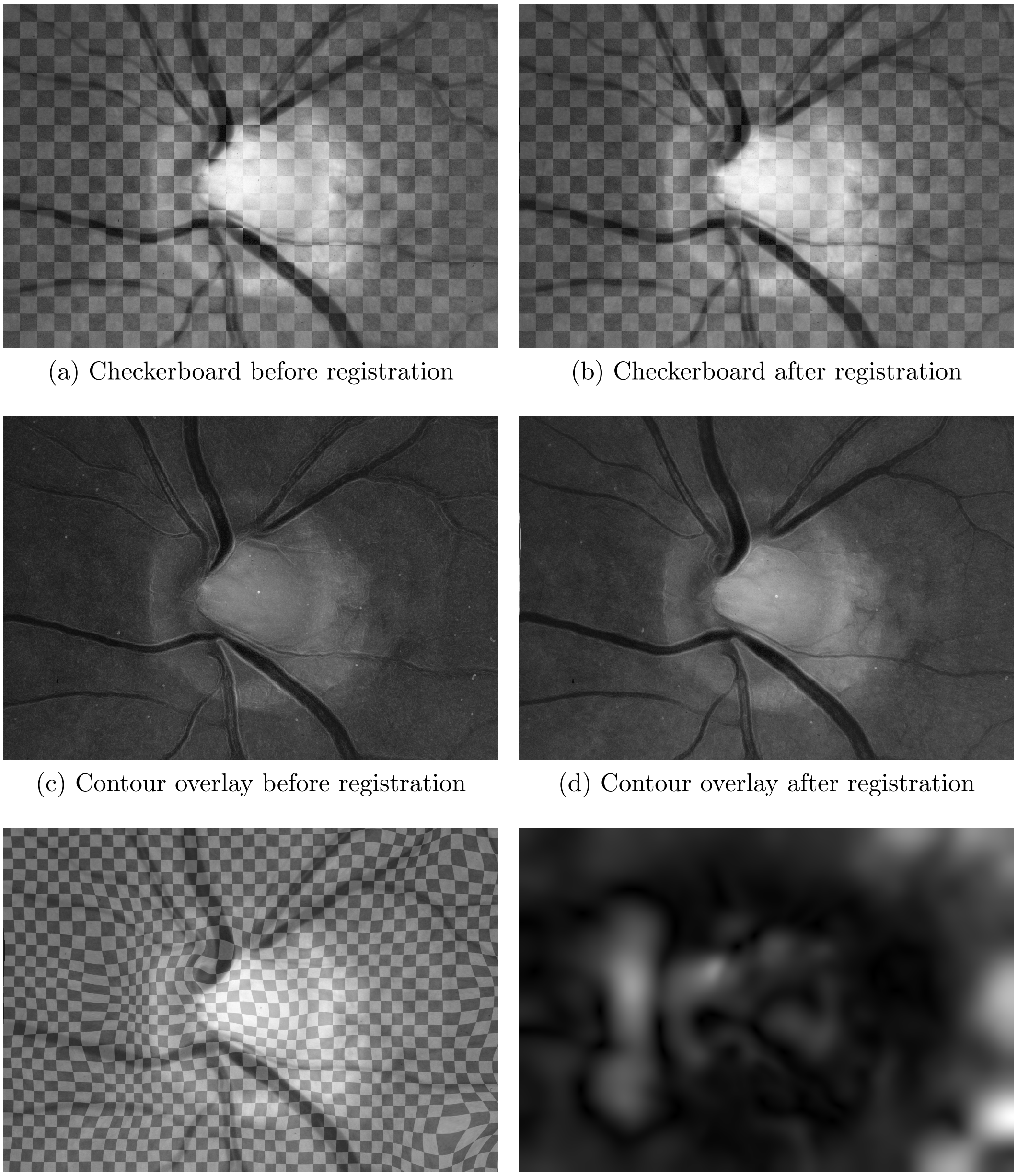
4.7.3 Glaucoma

The final dataset used for the evaluation is from a database of glaucoma, color fundus images. These are color images of the human retina, showing the optic nerve head in particular. There are currently several application scenarios based on these images that make use of a non-rigid registration. In [Paul 10] gradient magnitude images of these color fundus images are co-registered to get information about the optic nerve head variability that can provide information about the probability of glaucoma. The same method has also been applied directly to the images. A different approach registers two consecutive images of the same patient in order to extract movement from which 3-D information about the optic nerve head can be gained, which in turn can give further insights about the glaucoma risk of the patient.

The experiment performed for the optimization evaluation is concerned with the latter application. The images (see Figure 4.17) have a size of 1300 × 900 with gray values in the range of [0 255] (8-bit). The image spacing is taken as $\mathbf{h} = (1, 1)$ as no specific image resolution is available. As the illumination does not stay quite constant the registrations have to be considered multi-modal and the employed distance measure is the mutual information. The registrations use a stiffness of $\kappa = 3.5$ and 5 levels for the multi-level registrations. At first glance the images exhibit only slight differences, but results (see Figure 4.18) show some rather significant deformations especially at the left boundary of the optic nerve head.

The single level registration results (see Figure 4.19 and 4.20) show again the "L-BFGS secant" method in the lead. The "Newton secant" method and the semi-implicit gradient descent appear, depending on the used line search condition, pretty closely matched. The semi-implicit gradient descent again exhibits the peculiar behavior of performing worse in conjunction with Armijo condition, than with the simpler "descent" condition (see Figure 4.21). Aside from this the line search condition does not seem to have much impact in this application example.

The multi-level registration results show the same tendency as in the "hair follicle" dataset. The "L-BFGS secant" is in the lead followed by the "Newton secant" and the semi-implicit gradient descent The line search condition again has only a minor impact, but it improves the results somewhat, especially with the Wolfe conditions.



(a) Checkerboard before registration

(b) Checkerboard after registration

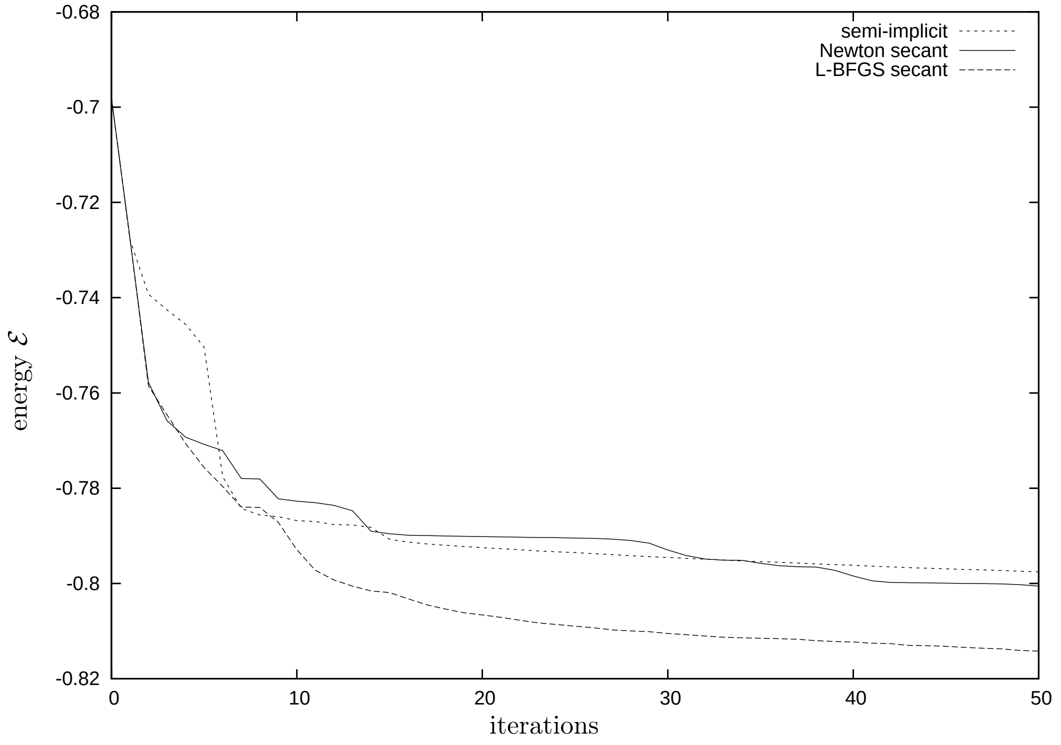
(c) Contour overlay before registration

(d) Contour overlay after registration

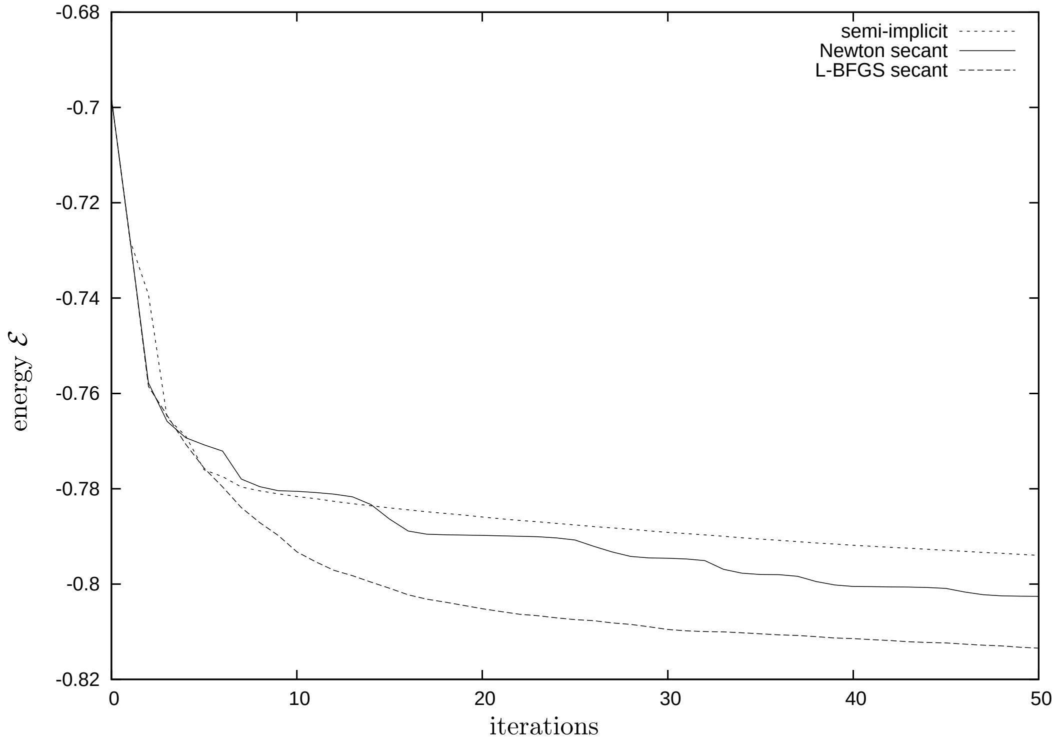
(e) Deformed image with checkerboard

(f) Magnitude image of deformation

Figure 4.18: Sample results for the registration of the glaucoma dataset. The used optimization algorithm is the multi-level “L-BFGS secant” with the Wolfe line search conditions. First row: checkerboard overlays of fixed and moving image (a) before and (b) after registration. Second row: fixed image overlaid with contours of moving image (c) before and (d) after registration. Last row: (e) moving image overlaid with a checkerboard image and deformed; (f) magnitude image of the deformation.



(a) descent



(b) Armijo

Figure 4.19: Comparison of optimizer performances with descent and Armijo type line search conditions on the "glaucoma" dataset.

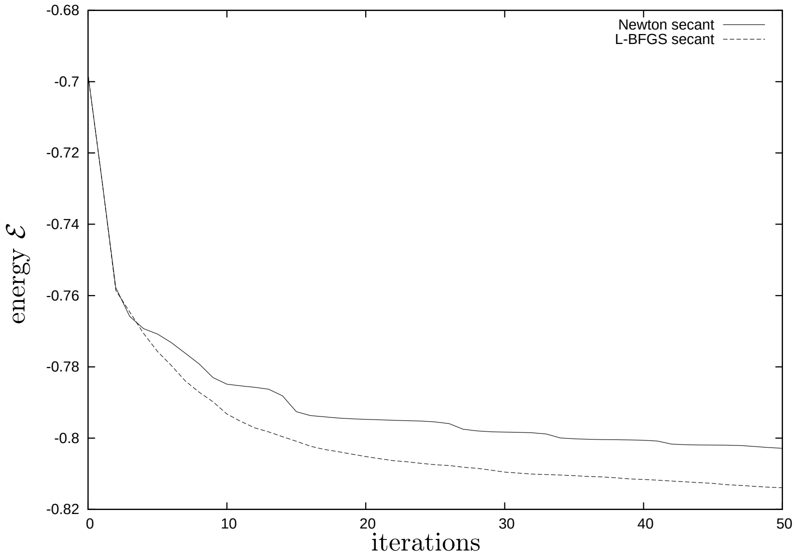
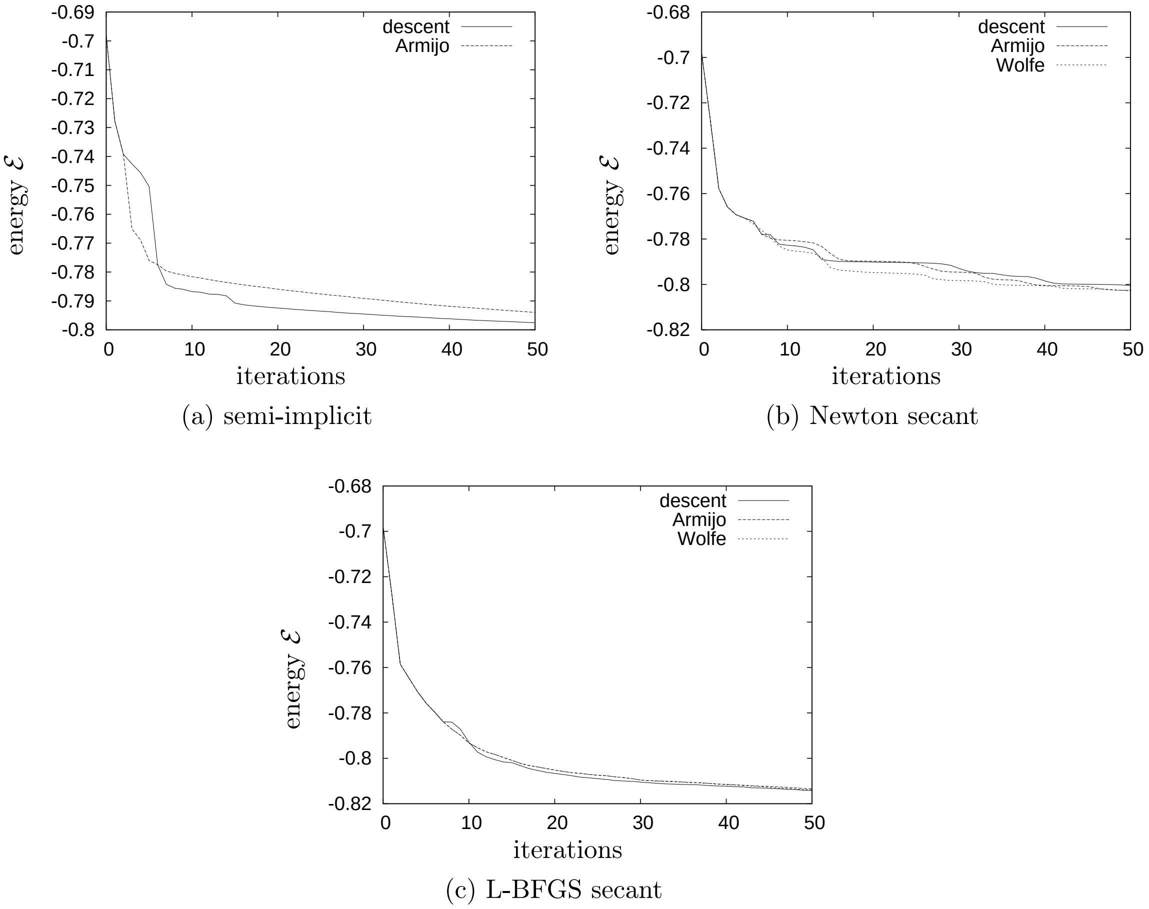


Figure 4.20: Comparison of optimizer performances with the Wolfe type line search conditions on the "glaucoma" dataset.



(a) semi-implicit

(b) Newton secant

(c) L-BFGS secant

Figure 4.21: Comparison of the impact of the choice of line search condition on the different optimization algorithms, for the "glaucoma" dataset.

	single level	multilevel
semi-implicit, descent	$-7.975746 \cdot 10^{-1}$	$-8.073360 \cdot 10^{-1}$
semi-implicit, armijo	$-7.939780 \cdot 10^{-1}$	$-8.067917 \cdot 10^{-1}$
Newton secant, descent	$-8.005637 \cdot 10^{-1}$	$-8.135049 \cdot 10^{-1}$
Newton secant, armijo	$-8.025690 \cdot 10^{-1}$	$-8.083490 \cdot 10^{-1}$
Newton secant, wolfe	$-8.028601 \cdot 10^{-1}$	$-8.106817 \cdot 10^{-1}$
L-BFGS secant, descent	$-8.142090 \cdot 10^{-1}$	$-8.194019 \cdot 10^{-1}$
L-BFGS secant, armijo	$-8.134388 \cdot 10^{-1}$	$-8.179640 \cdot 10^{-1}$
L-BFGS secant, wolfe	$-8.139087 \cdot 10^{-1}$	$-8.196936 \cdot 10^{-1}$

(a) Energy

	single level	multilevel
semi-implicit, descent	$-8.061491 \cdot 10^{-1}$	$-8.171590 \cdot 10^{-1}$
semi-implicit, armijo	$-8.021961 \cdot 10^{-1}$	$-8.161303 \cdot 10^{-1}$
Newton secant, descent	$-8.088138 \cdot 10^{-1}$	$-8.247281 \cdot 10^{-1}$
Newton secant, armijo	$-8.108823 \cdot 10^{-1}$	$-8.179362 \cdot 10^{-1}$
Newton secant, wolfe	$-8.116863 \cdot 10^{-1}$	$-8.206396 \cdot 10^{-1}$
L-BFGS secant, descent	$-8.246216 \cdot 10^{-1}$	$-8.339531 \cdot 10^{-1}$
L-BFGS secant, armijo	$-8.240784 \cdot 10^{-1}$	$-8.321323 \cdot 10^{-1}$
L-BFGS secant, wolfe	$-8.244515 \cdot 10^{-1}$	$-8.336899 \cdot 10^{-1}$

(b) Distance

	single level	multilevel
semi-implicit, descent	$8.574472 \cdot 10^{-3}$	$9.823000 \cdot 10^{-3}$
semi-implicit, armijo	$8.218024 \cdot 10^{-3}$	$9.338676 \cdot 10^{-3}$
Newton secant, descent	$8.250060 \cdot 10^{-3}$	$1.122314 \cdot 10^{-2}$
Newton secant, armijo	$8.313321 \cdot 10^{-3}$	$9.587228 \cdot 10^{-3}$
Newton secant, wolfe	$8.826252 \cdot 10^{-3}$	$9.957811 \cdot 10^{-3}$
L-BFGS secant, descent	$1.041264 \cdot 10^{-2}$	$1.455127 \cdot 10^{-2}$
L-BFGS secant, armijo	$1.063967 \cdot 10^{-2}$	$1.416825 \cdot 10^{-2}$
L-BFGS secant, wolfe	$1.054276 \cdot 10^{-2}$	$1.399628 \cdot 10^{-2}$

(c) Regularizer

Table 4.4: Final energies after optimization with the different algorithm and line search combinations, applied to the “glaucoma” dataset.

4.7.4 Discussion

Overall the results from the 3 example datasets lead to a few conclusions. In general the "L-BFGS secant" method performed best, or at least tied for best method in all application examples, single and multi-level. Even in the case of the mono-modal application where better, analytical estimates of the Hessian of the distance measure are available it performed almost on par with the Newton method employing that estimate. The "Newton secant" method usually performed either about as good or a little better than the semi-implicit gradient descent. In the multi-level framework it always outperformed the semi-implicit gradient descent. In general the multi-level results were in each instance a major improvement on the single level results. It is also noteworthy that the line search procedure is more complicated for the semi-implicit gradient descent and often required significantly more checks until a good first step size was found. In the Newton based methods, the line searching is computationally cheaper and easier to initialize. Those Newton based methods ("Newton secant", "L-BFGS secant") that employ a step of the semi-implicit gradient descent for initialization could run this single step, with a rather small step size, which made excessive backtracking unnecessary.

The choice of line search condition itself shows somewhat mixed results. In some cases of combinations of application example and optimization algorithm, a better and more computationally demanding line search condition like the Wolfe conditions also leads to an improvement in convergence speed. This is most notable in the "L-BFGS secant" and the "Newton SSD diagonal" algorithm. This is probably due to them yielding the best descent directions, for which a more demanding and thus more accurate line search will give the most benefits. On the other hand, in some applications the improvement was negligible even for those algorithms. With most of the other optimization algorithms the improvement is also quite disappointing. Finally, in the case of the semi-implicit gradient descent, the use of the Armijo condition instead of the simple "descent" condition actually made the algorithm perform much worse. As previously noted, this is probably due to the step size τ in this algorithm is not applied to a descent direction i. e. there is not actually a line to search on, which violates the basic assumptions made in the Armijo and Wolfe conditions.

Overall, it thus looks as if the "L-BFGS secant" was the clear winner of this comparison, combining good performance and multi-modal applicability. However, it also has some disadvantages. It has significant additional memory requirements to store the updates for the estimated Hessian matrix and some computational overhead applying them. For a 128^3 single precision floating point volume the additional memory requirement is for 5 updates to a 3-D vector field is $128^3 \cdot 3 \cdot 5 \cdot 4\text{Bytes} \approx 120\text{MB}$, which is manageable. For a 512^3 volume it is already 64 times as much i. e. 7.68GB, which should be quite detrimental on most consumer PCs. Additionally, the numerical estimates regarding the Hessian are always susceptible to ill behaved optimization functions. A sharp ridge in the target function can lead to a value that is far off from a good approximation to the Hessian. For the estimate of ϵ such a bad value only takes effect during the next iteration step, as ϵ is then replaced. In an L-BFGS scheme with 5 updates, it stays for the next 5 iterations. This admittedly happened very rarely and never for the datasets that were used to analyze the methods in this section. But it did happen and thus made the “Newton secant” an overall more robust algorithm. For these reasons, and for the reason that in the multi-level framework the convergence advantage of the L-BFGS algorithm is somewhat diminished the “Newton secant” method is used for all the applications presented in this work. As the “Newton secant” method showed very little dependency on the type of line search used, it is always used together with the “descent” line search condition, which requires the least computational amount to evaluate.

Chapter 5

Prior Information in Registration

The registration methods presented so far are based on some fundamental assumptions. The deformation field $\boldsymbol{u}$ should be smooth and the gray values should be nearly identical in a good match (for the sum of squared differences distance measure) or at least statistically dependent (for mutual information distance measure). All of these assumptions constitute some form of prior knowledge about the registration application. But so far the information incorporated into the registration approach is of a very broad and general nature. In this chapter we present some possibilities how more specific information, targeted at certain applications or even only valid for a specific pair of datasets can be used to add further stability and robustness to the registration algorithm.

Similarly to the components of the registration presented so far, the distance measure and the regularizer, the incorporation of prior information can have two targets. Either the type or shape of the deformation field can be further constrained or the way the gray values in the image are compared is augmented by some additional knowledge. We will concentrate on the first possibility, additional constraints on the deformation field. The advantage of constraints placed on the deformation field, is that they are applicable without change to any modality combination, assuming that the expected deformation is the same. This is an important property when applied for example to MR imaging, where even for the same imaging sequence the gray values are not standardized and thus can vary somewhat between successive scans. It also has the advantage that a training can be performed on simple to register modality combinations, with the result still being valid in more difficult application scenarios.

5.1 Landmark Correspondences

The most straightforward type of prior information about a registration is if some parts of the transform are already known. The classic case being known point-to-point correspondences or landmarks. Formally this means that there is an area or a set of areas $\Omega_{\boldsymbol{c}} \subset \Omega$ with known corresponding locations in the fixed image $\mathbf{x}_F$ and the moving image $\mathbf{x}_M$, and the function $\boldsymbol{c} : \Omega_{\boldsymbol{c}} \mapsto \mathbb{R}^d$ that associates them with each other as $\mathbf{x}_M = \boldsymbol{c}(\mathbf{x}_F)$. Therefore the transform in these areas is

$$\mathbf{x}_F = \boldsymbol{c}(\mathbf{x}_F) - \boldsymbol{u}(\mathbf{x}_F) \quad \forall \mathbf{x}_F \in \Omega_{\boldsymbol{c}}. \tag{5.1}$$

These landmarks could be used with a landmark based registration method such as thin-plate splines (compare Section 3.1.2). However, such a method only considers the landmarks and otherwise ignores the image content. Ideally, we would like to have a method that adheres to the known correspondences where available and considers the image content for the calculation of the match everywhere else.

This basic idea of combining landmark and intensity based registrations into one method has been treated in several works. Johnson and Christensen [John 02] propose an algorithm that alternatingly optimizes the landmark and intensity based registration. The intensity based registration itself is however unaware of the landmarks. They reason that if the landmarks are specified at image corners, edges or other significant points, the intensity based algorithm will not change them. Their algorithm, however, cannot guarantee that the landmarks are matched. This is especially the case if their assumption is violated and landmarks are specified so that they contradict the intensity information. Hartkens et al. [Hart 02] and Urschler et al. [Ursc 06] propose approaches that add an additional regularization term for the landmarks, to the energy term minimized in the intensity based registration. As the additional energy term only contributes a part of the matching energy both approaches also cannot guarantee an exact matching of the landmarks.

Fischer et al. propose in [Fisc 03a] to integrate (5.1) the landmarks with Lagrange multipliers as hard constraints into the registration energy. This allows an exact matching of the landmarks. The approach requires during each iteration step the calculation of the according Lagrange multipliers by solving an additional linear system, whose complexity depends on the number of landmark constraints used.

We propose a computationally simpler approach for introducing the landmark constraints in the registration formulation that likewise guarantees an exact match of the landmark points. The approach and its realization is outlined in the following section. In Section 5.1.2 a synthetic and a practically relevant application example of the method are presented. An application employing this approach in the context of the registration of histological slices has recently been published in [Gaff 11].

5.1.1 Optimization of Non-rigid Registration with Landmarks

The basic concept of our approach to integrate the landmarks is not to constrain the regions with known correspondences, but instead to entirely remove them from the computational domain. This can be mathematically expressed as

$$\tilde{\Omega} := \Omega \setminus \Omega_{\boldsymbol{c}}. \tag{5.2}$$

The known transform $\boldsymbol{u}(\mathbf{x}_F) \quad \forall \mathbf{x}_F \in \Omega_{\boldsymbol{c}}$ is used as a Dirichlet boundary condition (see (3.43)), which will influence the areas adjacent to these boundaries. This way the computational work is actually reduced for each constraint that is added, as the computational domain gets smaller and smaller.

We will use the semi-implicit gradient descent to outline a practical implementation of this approach, as it is much easier to integrate dirichlet boundary conditions into this scheme than it would be with the Newton based schemes. Let us recall the formulation for a problem without any point correspondences from (4.2)

$$(\mathbf{I}+\tau\alpha\mathbf{A})\mathbf{u}^{(t+1)} = \mathbf{u}^{(t)} - \tau\nabla_{\mathbf{u}}\mathcal{D}(F, M_{\mathbf{u}^{(t)}}), \tag{5.3}$$

where $\mathbf{A}$ is the system matrix of the regularizer (in this work either $\mathbf{A}_{\Delta}$ or $\mathbf{A}_{\Delta^2}$). The discretized correspondences from (5.1) are represented as known discrete transforms.

$$\mathbf{u}_i = \boldsymbol{u}(\mathbf{x}_i) = \mathbf{x}_i - \boldsymbol{c}(\mathbf{x}_i) \quad \forall i \;:\; \mathbf{x}_i \in \Omega_{\boldsymbol{c}} \tag{5.4}$$

With this information we could eliminate the rows i in the linear system of equation (5.3) right away, as these correspond to the, now known, variable $\mathbf{u}_i$. However, this would make it necessary to introduce a new mapping of values in $\mathbf{u}$ to the discrete positions in the image domain. For easier implementation we therefore instead replace the corresponding rows in the system matrix $\mathbf{I}+\tau\alpha\mathbf{A}$ by "identity stencils", which, in stencil notation (see Appendix B), are written as

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \tag{5.5}$$

Additionally, we set that $(\nabla\mathcal{D}(F, M_{\mathbf{u}}))_i = 0$, as the "derivative" with respect to a fixed value has to be 0. By making these two adjustments it is now ensured that $\mathbf{u}_i$ will not change if one iteration of the non-linear optimization is performed. Thus, if ${\mathbf{u}^{(0)}}_i$ is initialized with the known transform from the corresponding point-to-point correspondence, it will stay constant throughout the iteration. The neighborhood of these boundary points, however, will be influenced through the regularizer and result in a smooth deformation around the landmark.

It is debatable whether $\mathbf{u}_i$ should be included in the distance measure calculation. In the case of the sum of squared differences distance measure this would result in a constant contribution to the value of the distance measure and therefore be of no consequence. In the mutual information distance measure, in contrast, the points would influence the overall statistics and therefore have an influence on the distance measure value and also its non-linear behavior. In all our experiments we adhere to our initially proposed idea to completely remove $\Omega_{\boldsymbol{c}}$ from Ω and thus exclude the known points from the distance measure computation altogether.

A slight drawback of this approach is that the computational domain Ω loses its rectangular shape. This has the following consequences: Normally, the regularizer matrix $\mathbf{A}$ and therefore the system matrix $\mathbf{I}+\alpha\tau\mathbf{A}$ is symmetric (compare Section 3.4.1 and 3.4.2). However, as the changes indicated above are only applied to some of the rows of this matrix, the ones that correspond to pixels removed from the domain Ω, this symmetry is, in general, lost. As the symmetry of this matrix is a prerequisite of the application of the FFT based solver scheme or also a standard conjugate gradient solver these are now not applicable anymore. The linear system either has to be resymmetrized by performing Gaussian elimination on the columns containing i corresponding to the modified rows i or we have to rely on linear solvers that can deal with non-symmetric matrices like the stabilized bi-conjugate gradient method.

In our experiments the bi-conjugate gradient method often was slow to converge when started from an all zero initial deformation field. We therefore additionally try to provide a good initialization. This is done by solving the optimization for the regularizer only i. e.

$$\mathbf{A}\mathbf{u}^{(0)} = \mathbf{f} \tag{5.6}$$

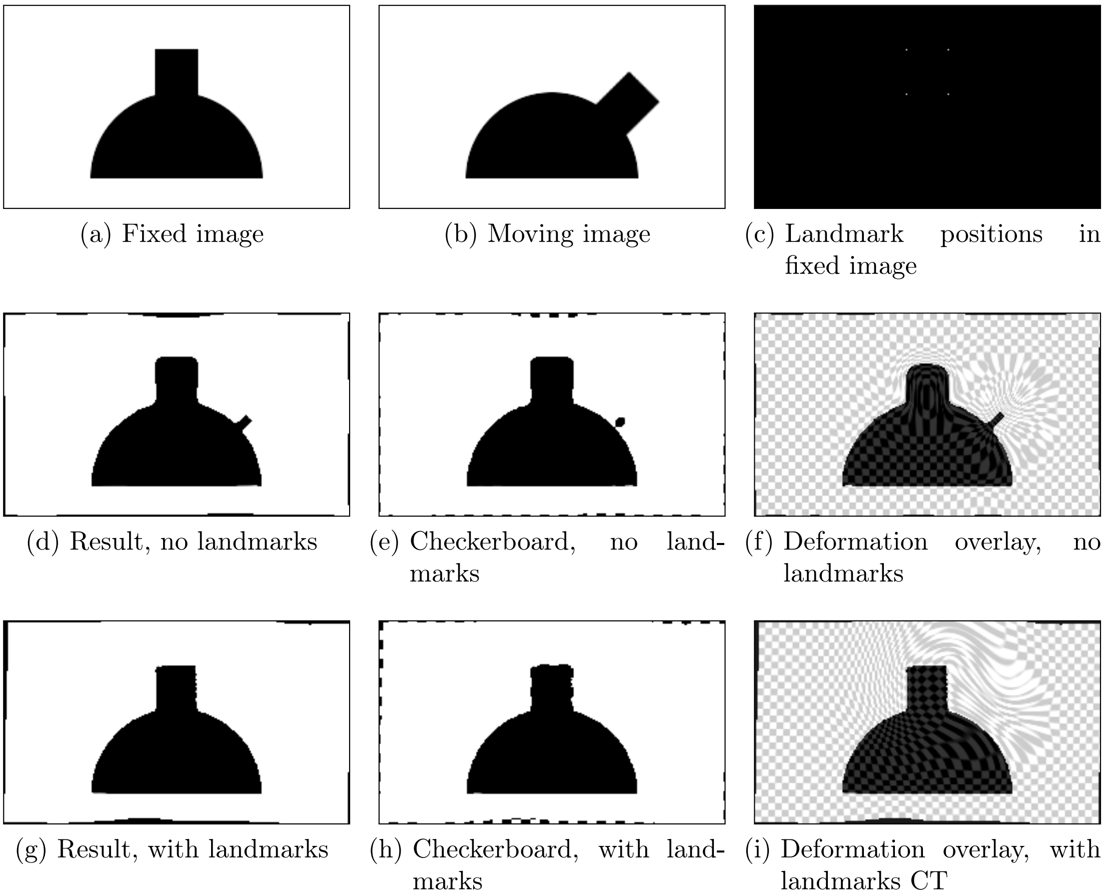
where $\mathbf{A}$ is again a placeholder for the regularizer matrix and $\mathbf{f}$ is an all 0 vector except for the entries corresponding to a landmark, which are set according to (5.1). The result is a smooth deformation field only based on the landmarks. For the curvature regularizer this deformation field is to some extent similar to the deformation that would have been obtained for a thin-plate spline registration. The difference is that the thin-plate spline also minimizes the mixed derivatives, which are not considered in the curvature regularizer. This deformation field, which is globally smooth and satisfies the landmark constraints, is used as an initial guess for the actual registration algorithm.

Another issue that has to be dealt with is how to treat the landmarks in the multilevel scheme that was outlined in Section 4.6. In order to specify landmarks on the coarse grid we consider each coarse grid pixel whose physical location coincides with one or more landmarks to be a landmark as well. Naturally, this will lead to a “growth” of the discrete landmark regions on the coarser grids, as an isolated fine grid landmark pixel will result in a single coarse grid landmark pixel, which is physically much larger. If there are larger landmark regions specified on the fine grid this growth effect can only occur on the boundary of these regions. Nevertheless, the coarse grid solutions always proved to be good enough in our experiments to benefit the optimization on the finer levels. The initialization of the next finer grid is then done by upsampling of the coarse grid solution as explained in Section 4.6), with one additional step: On the finer grid the landmark pixels are reset to the fixed values that they should have to compensate for smoothing or numerical errors from the up-sampling.

5.1.2 Application Examples

To illustrate the influence of the landmarks on the standard registration approach we first consider a synthetic example depicted in Figure 5.1. For a human observer it seems obvious that the square structure in the image has moved. For the registration algorithm, however, any solution that maps the black objects onto each other is fine. The standard algorithm therefore tries to shrink the square structure into the main object at the one location and pull it out again at the other. In order to get the rotating motion instead 4 landmarks are specified at the corners of the square. These landmarks are sufficient to nudge the registration algorithm into doing the desired deformation while still keeping the rest of the object in place.

As a real application problem we chose a full body PET with CT registration problem. This kind of registration application is difficult for several reasons. First, PET and CT show fundamentally different things (compare also Section 2.1 and 2.4),


(a) Fixed image

(b) Moving image

(c) Landmark positions in fixed image

(d) Result, no landmarks

(e) Checkerboard, no landmarks

(f) Deformation overlay, no landmarks

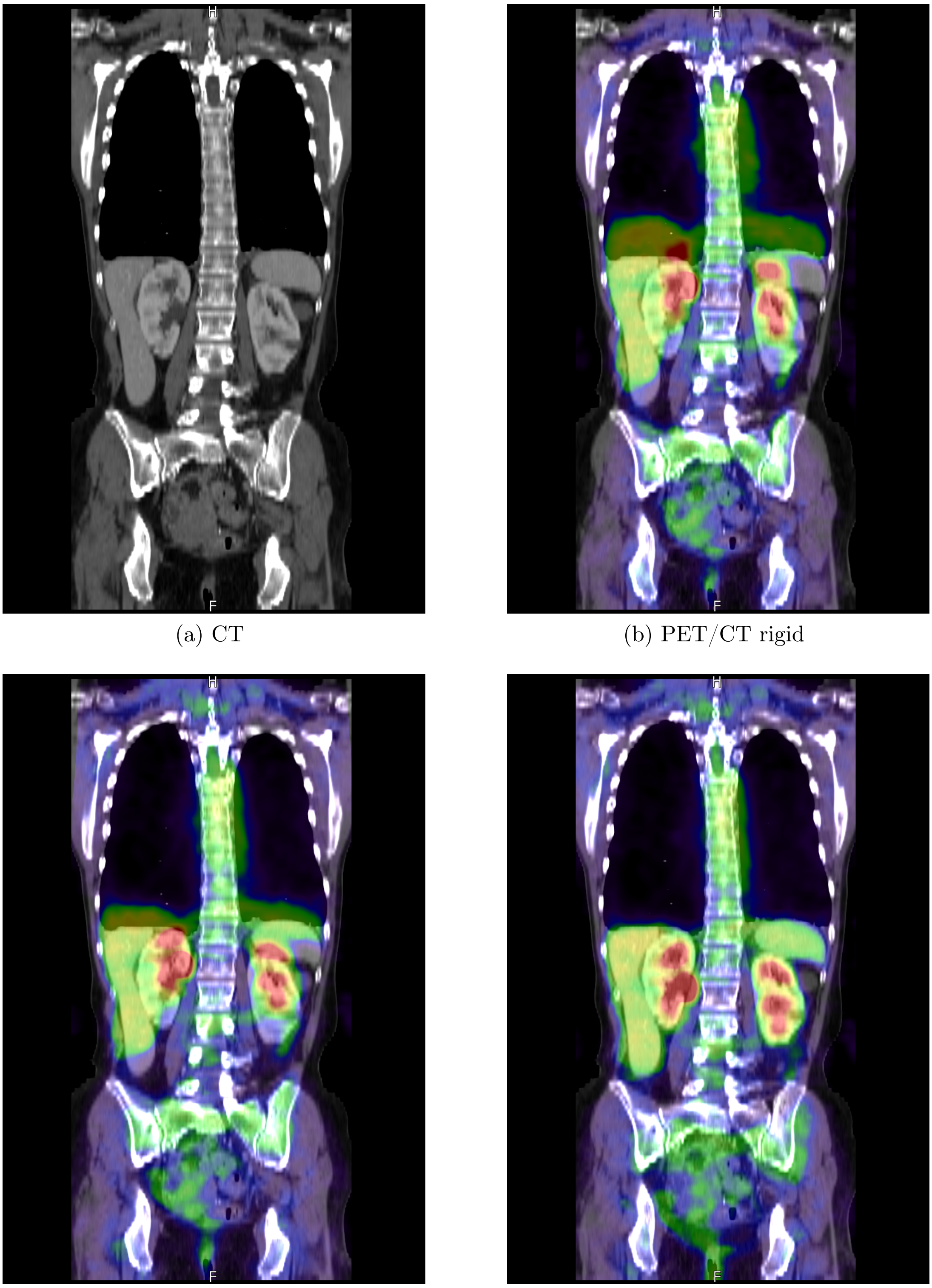
(g) Result, with landmarks

(h) Checkerboard, with landmarks

(i) Deformation overlay, with landmarks CT

Figure 5.1: Non-rigid registration of a synthetic dataset to illustrate the influence of the landmarks on the standard non-rigid registration algorithm. Without the additional landmarks (second row) the protruding square is shrunk into the structure and drawn out at the new location. That this process cannot be completed (remaining line and rounded corners of the new square) is due to the regularizer, prohibiting such a very non-smooth deformation to some extent. With the landmarks set (third row) it is actually rotated while the distance measure keeps the remaining structure in place.

which means that not every structure visible in one dataset has a counterpart in the other. Furthermore, the blurry appearance of the PET and the intensity gradients sometimes present in organs that appear homogeneous in the CT, leads to not very peaky, smeared looking joint histograms in the mutual information, which degrades the matching energy. Finally, the typical image acquisition protocol for CT requires the patient to lie with his arms up and fully inhaled, while in PET he has the arms down and is freely breathing. These very different patient positions lead to large deformations between the datasets. As a result one would need to select a rather low stiffness to have enough freedom to move the organs around. As a downside, this also makes it possible for the algorithm to locally deform the datasets in a way to compensate for their inherently different characteristics like the higher degree of blurriness in the PET, which is an intolerable behavior. Figure 5.2c shows the result of such a non-rigid registration with a stiffness parameter that guarantees that no such local deformations in the PET can occur. The coronal slices show a definite improvement over the only rigidly registered dataset (Figure 5.2b) but there is still a large degree of mismatch. With landmarks placed on the highest and lowest points of both kidneys along the axial dimension of the dataset, the lowest point of the liver and two at the diaphragm, the result improves immensely (Figure 5.2d). The improvement is also visible in areas like the lungs where the landmarks cannot have a direct influence. Instead, we presume the better match in other image parts such as the liver and the kidneys leads to a better overall matching energy. This in turn improves the match in image parts not directly affected by the landmarks.



(a) CT

(b) PET/CT rigid

(c) PET/CT non-rigid

(d) PET/CT non-rigid with landmarks

Figure 5.2: Coronal planes of fused 3-D CT and PET dataset, showing a (b) rigid, (c) non-rigid and (d) non-rigid with landmarks registration result, respectively. Landmarks have been specified in the highest and lowest points along the axial dimension of the kidneys (4 landmarks), at the lower tip of the liver (1 landmark) and at the diaphragm (2 landmarks).

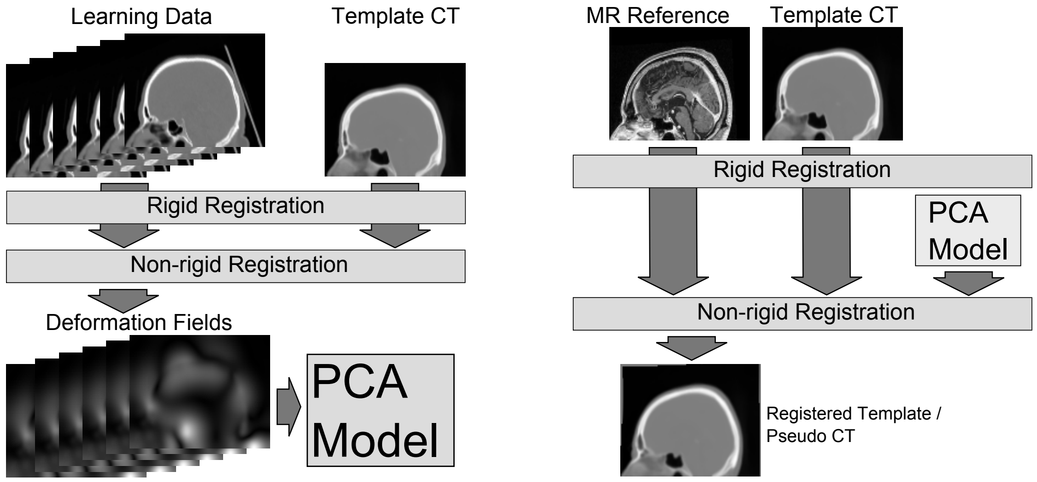
5.2 Deformation Models

A more involved, but also far more flexible way to impose constraints on a deformation field, is to constrain the deformation field to be similar to transformations that were observed previously for other datasets in the same application. In other words there has to be a learning phase where "good" transformations for the application in question are analyzed and the result is used to make subsequent registrations more robust. The deformations used for training are analyzed statistically, for example with principal component analysis (PCA) to determine major modes of variation that are characteristic for the kind of deformations occurring in the application. This kind of statistical model constraint in image registration is usually referred to as statistical deformation models (SDM). Wang and Staib [Wang 00] describe a method that generates a sparse PCA-based model on a set of boundary points that they use to constrain the dense non-rigid registration. Kim et al. [Kim 08] construct a dense PCA-based deformation model from registrations with a standard registration approach. The model is used to generate a large set of sample images which are then compared to the reference image in order to find a good starting position for a standard registration approach. Xue and Shen [Xue 09] propose a model base on the wavelet PCA that has the advantage to also capture very local and fine grained deformations. Nevertheless, the model is only used for an initial registration followed by an unconstrained non-rigid registration. Wouters et al. [Wout 06] use a PCA based regularization in conjunction with a viscous flow registration. They constrain the registration result completely within the space represented by the PCA. On a basis of 85 learning datasets of the brain they are still somewhat short of a complete coverage of the possible deformations. This indicates that for more complex anatomies it is rather unlikely that it is possible to capture the whole range of possible anatomical variability with such a model.

This work therefore focuses on a regularization with a PCA model that prefers the resulting deformation to be close to the model space, but does not rigidly enforce it. The hope is that, similarly to the addition of landmarks to the registration approach as in Section 5.1 it is sufficient to "nudge" the algorithm in the right direction, without the need to have a model that covers every little detail.

The general workflow of our approach is outlined in Figure 5.3. For training (see Figure 5.3a) a template dataset is registered with a number of training datasets. Preferably these registrations are more robust and reliable than the registration in the latter application. For instance, in our application case (see Section 6) we want to improve multi-modal registrations, by learning from the results of a set of mono-modal registrations, which are less error prone. These deformation fields are treated as the gold standard for the application and are used to train the PCA based model. Later on the model is then used to constrain the registration (see Figure 5.3b) to stay close to known deformation fields from the model, which makes the registration more robust. A first work describing this approach, was published in [Daum 09].

A related method has been published by Albrecht et al. [Albr 08]. They use a PCA based regularization to enhance a diffusion regularized registration of shape images, i. e. distance transforms of segmentations. One of the major advantages of the two approaches presented here, is the added robustness with respect to the initial rigid



(a) Model generation

(b) Model application

Figure 5.3: Workflow for (a) the generation and (b) application of the PCA model.

alignment of the datasets, either by incorporating translations explicitly in the model or by working on derivatives of the deformation field. In [Albr 08] the regularization scheme is also introduced only in the discrete, while we develop the regularization term consistently in the non-rigid registration framework outlined in Section 3.2. They also do not discuss how they optimize the final energy functional, although the optimization scheme can have a considerable influence on the final result.

In the subsequent sections the theoretical and practical tools to implement this regularizer are introduced. A practical application of this approach is presented in Section 6.

5.2.1 Functional Principal Component Analysis

In the field of non-parametric, non-rigid registration we are dealing with the problem of optimizing for an unknown function $\boldsymbol{u}$. To handle the probabilistic analysis of functions as training data we therefore have to turn to methods for functional data analysis (see e. g. [Rams 05]). As we want to identify common modes of variation we make use of the functional principle component analysis (PCA). The function space $\mathcal{U}$ introduced in Section 3.2 defines an inner product that induces the norm $\|\boldsymbol{u}\|_{\mathcal{U}} = \sqrt{\langle \boldsymbol{u}, \boldsymbol{u} \rangle_{\mathcal{U}}}$. The aim of a functional PCA decomposition is thus to find mutually orthogonal modes $\boldsymbol{v}_i$ that optimize

$$\hat{\boldsymbol{v}}_i(\mathbf{x}) = \underset{\boldsymbol{v}_i}{\operatorname{argmax}} \sum_{j=1}^{m} \langle \boldsymbol{w}_j - \bar{\boldsymbol{w}}, \boldsymbol{v}_i \rangle_{\mathcal{U}}^2 \qquad \text{(maximum variation)} \tag{5.7}$$

$$\text{with} \quad \bar{\boldsymbol{w}}(\mathbf{x}) = \frac{1}{m} \sum_{i=1}^{m} \boldsymbol{w}_i(\mathbf{x}) \qquad \text{(mean)} \tag{5.8}$$

$$\text{subject to} \tag{5.9}$$

$$\|\boldsymbol{v}_i\|_{\mathcal{U}}^2 = \langle \boldsymbol{v}_i, \boldsymbol{v}_i \rangle_{\mathcal{U}} = 1 \qquad \text{(normal length)} \tag{5.10}$$

$$\langle \boldsymbol{v}_i, \boldsymbol{v}_j \rangle_{\mathcal{U}} = 0 \quad \forall i \neq j \qquad \text{(orthogonality),} \tag{5.11}$$

where $\boldsymbol{w}_i \quad i = 1, \dots, m$ are the m training functions, and $\bar{\boldsymbol{w}}$ is the corresponding mean training deformation. The modes $\boldsymbol{v}_i$ then represent the principal axes along which the major variations in the sample data could be observed. They can be used to regularize the registration by favoring deformations that coincide with these axes. The principal axis $\boldsymbol{v}_i$ can be determined by incorporating the constraint ensuring the normal length by a Lagrangian multiplier

$$\hat{\boldsymbol{v}}_i = \underset{\boldsymbol{v}_i}{\operatorname{argmax}} \, \mathcal{E}_{\mathrm{PCA}}(\boldsymbol{v}_i) = \underset{\boldsymbol{v}_i}{\operatorname{argmax}} \left(\sum_{j=1}^{m} \langle \boldsymbol{w}_j - \bar{\boldsymbol{w}}, \boldsymbol{v}_i \rangle_{\mathcal{U}}^2 \right) - \lambda(\langle \boldsymbol{v}_i, \boldsymbol{v}_i \rangle_{\mathcal{U}} - 1). \tag{5.12}$$

From this, one can calculate the Gâteaux derivative to identify the extremal points.

$$\begin{aligned}
\mathsf{d}\mathcal{E}_{\mathrm{PCA}}(\boldsymbol{v}_i; \boldsymbol{\eta}) &= \frac{\mathsf{d}}{\mathsf{d}\epsilon} \left(\sum_{j=1}^{m} \left(\frac{1}{|\Omega|} \int_{\Omega} (\boldsymbol{w}_j(\mathrm{x}) - \bar{\boldsymbol{w}}(\mathrm{x}))^T (\boldsymbol{v}(\mathrm{x}) + \epsilon \boldsymbol{\eta}(\mathrm{x})) \, \mathsf{dx} \right)^2 \right) \Bigg|_{\epsilon=0} \\
&\quad - \frac{\mathsf{d}}{\mathsf{d}\epsilon} \lambda \frac{1}{|\Omega|} \int_{\Omega} (\boldsymbol{v}_i(\mathrm{x}) + \epsilon \boldsymbol{\eta}(\mathrm{x}))^2 \, \mathsf{dx} \Bigg|_{\epsilon=0} \\
&= \left(\sum_{j=1}^{m} \frac{2}{|\Omega|^2} \int_{\Omega} \boldsymbol{\eta}(\mathrm{y})^T (\boldsymbol{w}_j(\mathrm{y}) - \bar{\boldsymbol{w}}(\mathrm{y})) \, \mathsf{dy} \int_{\Omega} (\boldsymbol{w}_j(\mathrm{x}) - \bar{\boldsymbol{w}}(\mathrm{x}))^T \boldsymbol{v}(\mathrm{x}) \, \mathsf{dx} \right) \\
&\quad - \lambda \frac{2}{|\Omega|} \int_{\Omega} \boldsymbol{\eta}(\mathrm{x})^T \boldsymbol{v}_i(\mathrm{x}) \, \mathsf{dx} \\
&= 2 \left\langle \boldsymbol{\eta}, \left(\sum_{j=1}^{m} (\boldsymbol{w}_j - \bar{\boldsymbol{w}}) \langle (\boldsymbol{w}_j - \bar{\boldsymbol{w}}), \boldsymbol{v} \rangle_{\mathcal{U}} \right) - \lambda \boldsymbol{v}_i(\mathrm{x}) \right\rangle_{\mathcal{U}} \\
&= 0 \quad \forall \boldsymbol{\eta} \in \mathcal{U}
\end{aligned} \tag{5.13}$$

It therefore follows that

$$\left\langle \boldsymbol{\eta}, \sum_{j=1}^{m} (\boldsymbol{w}_j - \bar{\boldsymbol{w}}) \langle (\boldsymbol{w}_j - \bar{\boldsymbol{w}}), \boldsymbol{v}_i \rangle_{\mathcal{U}} \right\rangle_{\mathcal{U}} = \langle \boldsymbol{\eta}, \lambda_i \boldsymbol{v}_i(\mathrm{x}) \rangle_{\mathcal{U}} \tag{5.14}$$

which is the continuous version of an Eigenvalue problem. In practice this is solved in the discrete, by replacing the variables with their discrete equivalents and the inner product with its discretized version from equation (3.17). This essentially reduces the functional PCA to a PCA on the discrete values of $\mathbf{u}$ at the grid positions defined in $\mathbf{X}$. The discrete Eigensystem is thus

$$\begin{aligned}
\sum_{j=1}^{m} \frac{1}{s} (\mathbf{w}_j - \bar{\mathbf{w}})(\mathbf{w}_j - \bar{\mathbf{w}})^T \mathbf{v} &= \lambda \mathbf{v} \\
\frac{1}{s} \mathbf{W} \mathbf{W}^T \mathbf{v} &= \lambda \mathbf{v}.
\end{aligned} \tag{5.15}$$

with $\mathbf{W} = (\mathbf{w}_1 - \bar{\mathbf{w}}, \dots, \mathbf{w}_n - \bar{\mathbf{w}}) \in \mathbb{R}^{m \times sd}$, s the number of samples in the discrete domain Ω and d the number of dimension.

The Eigenvalue / Eigenvector analysis of $\frac{1}{s}\mathbf{W}\mathbf{W}^T$ to determine the principal axis $\mathbf{v}_i$ and the corresponding Eigenvalues λ_i, would be very computationally costly, if not outright impossible, if performed directly, as $\mathbf{W}\mathbf{W}^T \in \mathbb{R}^{sd \times sd}$ can become very large. As we have only a small number of sample deformation fields m, which is much smaller than the number of unknowns sd in the discrete deformations $\mathbf{w}_j$, we employ a trick for the Eigenvalue / Eigenvector analysis of $\mathbf{W}\mathbf{W}^T$ which was, to our knowledge introduced by Murase and Lindenbaum [Mura 95]. Multiplying (5.15) from the left with $\mathbf{W}^T$, yields

$$\frac{1}{s}(\mathbf{W}^T\mathbf{W})\mathbf{W}^T\mathbf{v} = \lambda \mathbf{W}^T\mathbf{v}. \tag{5.16}$$

It is thus evident that solving the smaller system

$$\frac{1}{s}(\mathbf{W}^T\mathbf{W})\tilde{\mathbf{v}} = \lambda \tilde{\mathbf{v}} \tag{5.17}$$

with $\mathbf{W}^T\mathbf{W} \in \mathbb{R}^{m \times m}$, also leads to a solution of the larger system (5.15) with $\mathbf{v} = \mathbf{W}^T\tilde{\mathbf{v}}$.

While the determined Eigenvectors $\boldsymbol{v}_i$ specify the directions of maximal variation, the associated Eigenvalues λ are related to the variation along these principal axes. The variance along the axis $\boldsymbol{v}_i$ is

$$\begin{aligned}
\sigma_i^2 &= \frac{1}{m}\sum_{j=1}^{m} \langle \boldsymbol{w}_j - \bar{\boldsymbol{w}}, \boldsymbol{v}_i\rangle_{\mathcal{U}}^2 \\
&= \frac{1}{m|\Omega|}\int_\Omega \boldsymbol{v}_i(\mathrm{x})^T \sum_{j=1}^{m} (\boldsymbol{w}_j(\mathrm{x}) - \bar{\boldsymbol{w}}(\mathrm{x}))\langle \boldsymbol{w}_j - \bar{\boldsymbol{w}}, \boldsymbol{v}_i\rangle_{\mathcal{U}} \,\mathsf{dx} \\
&= \frac{1}{m|\Omega|}\int_\Omega \boldsymbol{v}_i(\mathrm{x})^T \lambda \boldsymbol{v}_i(\mathrm{x}) \,\mathsf{dx} && \text{by (5.14)} \\
&= \frac{\lambda}{m} && \text{by (5.10)}\ .
\end{aligned} \tag{5.18}$$

Similarly, the total variance of the data can be expressed in terms of the variances along the Eigenvectors and therefore the Eigenvalues. In the first step of the following calculation we make use of the fact that all the Eigenvectors $\boldsymbol{v}_i$ together form an orthonormal basis for the samples $\boldsymbol{w}_j$. We can therefore apply a basis transform and get

$$\begin{aligned}
\sigma_{\text{total}}^2 &= \frac{1}{m}\sum_{j=1}^{m} \|\boldsymbol{w}_j - \bar{\boldsymbol{w}}\|_{\mathcal{U}}^2 \\
&= \frac{1}{m}\sum_{j=1}^{m} \left\| \sum_{i=1}^{m} \boldsymbol{v}_i \langle (\boldsymbol{w}_j - \bar{\boldsymbol{w}}), \boldsymbol{v}_i\rangle_{\mathcal{U}} \right\|_{\mathcal{U}}^2
\end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{m}\sum_{j=1}^{m}\frac{1}{|\Omega|}\int_{\Omega}\sum_{i=1}^{m}\sum_{k=1}^{m}\langle(\boldsymbol{w}_j-\bar{\boldsymbol{w}}),\boldsymbol{v}_i\rangle_{\mathcal{U}}\boldsymbol{v}_i(\mathrm{x})^T\boldsymbol{v}_k(\mathrm{x})\langle(\boldsymbol{w}_j-\bar{\boldsymbol{w}}),\boldsymbol{v}_k\rangle_{\mathcal{U}}\,\mathsf{dx}\\
&= \frac{1}{m}\sum_{j=1}^{m}\sum_{i=1}^{m}\sum_{k=1}^{m}\langle(\boldsymbol{w}_j-\bar{\boldsymbol{w}}),\boldsymbol{v}_i\rangle_{\mathcal{U}}\ \underbrace{\langle\boldsymbol{v}_i(\mathrm{x}),\boldsymbol{v}_k(\mathrm{x})\rangle_{\mathcal{U}}}_{=1\text{ if }i=k,\ =0\text{ otherwise}}\ \langle(\boldsymbol{w}_j-\bar{\boldsymbol{w}}),\boldsymbol{v}_k\rangle_{\mathcal{U}}\\
&= \sum_{j=1}^{m}\underbrace{\frac{1}{m}\sum_{i=1}^{m}\langle(\boldsymbol{w}_j-\bar{\boldsymbol{w}}),\boldsymbol{v}_i\rangle_{\mathcal{U}}^2}_{=\frac{\lambda_j}{m}\text{ by (5.18)}}\\
&= \sum_{j=1}^{m}\frac{\lambda_j}{m}=\sum_{j=1}^{m}\sigma_k^2. \qquad (5.19)
\end{aligned}
$$

These relations are used in Section 6.1.3 to calculate how much of the overall variation in the data is covered by a certain number of PCA modes in the model. This is necessary for choosing a suitable number of modes for the application.

5.2.2 PCA Regularization

A straightforward approach to generate the proposed morphological model is to perform the PCA directly on the training deformations. The resulting model is incorporated into the registration energy $\mathcal{E}(F,M,\boldsymbol{u})$ (from (3.10)) as an additional regularization, which forces the result to be close to the model by quadratically penalizing any deviation from the model space represented by the PCA.

$$\boldsymbol{u}^* = \operatorname*{argmin}_{\boldsymbol{u}}\mathcal{E}(F,M,\boldsymbol{u}) = \mathcal{D}(F,M_{\boldsymbol{u}})+\alpha\mathcal{R}(\boldsymbol{u})+\beta\mathcal{P}(\boldsymbol{u}) \qquad (5.20)$$

$$
\begin{aligned}
\mathcal{P}(\boldsymbol{u}) &= \left\|\boldsymbol{u}(\mathrm{x})-(\bar{\boldsymbol{w}}+\sum_{i=1}^{m}\boldsymbol{v}_i(\mathrm{x})\langle\boldsymbol{v}_i,\boldsymbol{u}-\bar{\boldsymbol{w}}\rangle_{\mathcal{U}})\right\|_{\mathcal{U}}^2\\
&= \frac{1}{|\Omega|}\int_{\Omega}\left(\boldsymbol{u}(\mathrm{x})-(\bar{\boldsymbol{w}}+\sum_{i=1}^{m}\boldsymbol{v}_i(\mathrm{x})\langle\boldsymbol{v}_i,\boldsymbol{u}-\bar{\boldsymbol{w}}\rangle_{\mathcal{U}})\right)^2\mathsf{dx}, \qquad (5.21)
\end{aligned}
$$

where β is a weighting factor that governs the strictness with which the morphological model is applied.

Even though all datasets are aligned rigidly before performing the non-rigid registration for the training as well as for the application of the model, the result of this rigid registration is not always consistent. As we are dealing with registrations between different patients the rigid registration cannot yield a perfect result. The rigid registration therefore tries to find a “best possible” match. Depending on the data this can lead to the algorithm aligning those parts of the patient data best that are most similar, i. e. in one registration the facial area might be matched best and in another the back of the skull. This is a problem for the PCA model as it is sensitive to differences in this initial rigid alignment. To alleviate this problem somewhat we make our model robust to variations in the translational part of the rigid alignment, by explicitly introducing global rigid translations into the model. We define the d constant global rigid translation functions as

$$\boldsymbol{t}_j(\mathbf{x}) = (0, \ldots, 0, \underbrace{1}_{j\text{-th entry}}, 0, \ldots, 0)^T \in \mathbb{R}^d \quad j = 1, \ldots, d. \tag{5.22}$$

Note that $\boldsymbol{t}_j$ are chosen such that they have unit length i.e. $\|\boldsymbol{t}_j\|_{\mathcal{U}}^2 = 1$, and are mutually orthogonal. These translations are then removed from all training data $\boldsymbol{w}_i$, by using

$$\tilde{\boldsymbol{w}}_i(\mathbf{x}) = \boldsymbol{w}_i(\mathbf{x}) - \sum_{j=1}^{d} \boldsymbol{t}_j(\mathbf{x}) \langle \boldsymbol{t}_j, \boldsymbol{w}_i \rangle_{\mathcal{U}} \qquad i = 1, \ldots, n, \tag{5.23}$$

as the new training data. The adjusted training data $\tilde{\boldsymbol{w}}_i$ is thus orthogonal to the global rigid translations $\boldsymbol{t}_j$. As the m Eigenmodes $\boldsymbol{v}_k$ are linear combinations of the training data, they have to be orthogonal to the functions $\boldsymbol{t}_j$ as well. This makes the modes $\boldsymbol{t}_j$ orthonormal to the PCA basis $\boldsymbol{v}_k$ and we are free to define $\boldsymbol{v}_{m+j} = \boldsymbol{t}_j$ as additional modes for the PCA basis. Adding these "artificial" modes to the PCA makes $\mathcal{P}$ invariant to global translations in the vector field $\boldsymbol{u}$, as they are now included in the model and will not be penalized by $\mathcal{P}(\boldsymbol{u})$.

Unfortunately, the same approach cannot be applied to compensate rotations. It is certainly possible to represent a rotation by a vector field. However, if that vector field is scaled by a scalar value, as this happens when used as a PCA mode, the rotation turns into a scaling and rotating operation, as the deformation vectors scale linearly and do not follow the circular movement given by the rotation.

In order to incorporate our new PCA energy term into the registration framework, the derivative $\nabla_{\boldsymbol{u}} \mathcal{P}$ has to be calculated. During this derivation we make, for now, the simplifying assumption that the samples have a zero mean, i.e. $\bar{\boldsymbol{w}}(\mathbf{x}) = 0$. The Gâteaux derivative of $\mathcal{P}$ is thus

$$\begin{aligned}
&\frac{\mathsf{d}}{\mathsf{d}\epsilon} \mathcal{P}(\boldsymbol{u} + \epsilon \boldsymbol{\eta}) \bigg|_{\epsilon=0} \\
&= \frac{2}{|\Omega|} \int_\Omega \left(\boldsymbol{\eta}(\mathbf{x}) - \sum_{i=1}^{m} \boldsymbol{v}_i(\mathbf{x}) \langle \boldsymbol{v}_i, \boldsymbol{\eta} \rangle_{\mathcal{U}} \right)^T \left(\boldsymbol{u}(\mathbf{x}) - \sum_{i=1}^{m} \boldsymbol{v}_i(\mathbf{x}) \langle \boldsymbol{v}_i, \boldsymbol{u} \rangle_{\mathcal{U}} \right) \mathsf{dx} \\
&= 2 \left\langle \boldsymbol{\eta}, \boldsymbol{u} - \sum_{i=1}^{m} \boldsymbol{v}_i \langle \boldsymbol{v}_i, \boldsymbol{u} \rangle_{\mathcal{U}} \right\rangle_{\mathcal{U}} - 2 \left(\sum_{i=1}^{m} \langle \boldsymbol{\eta}, \boldsymbol{v}_i \rangle_{\mathcal{U}} \langle \boldsymbol{v}_i, \boldsymbol{u} \rangle_{\mathcal{U}} \right) \\
&\quad + 2 \Big(\sum_{i=1}^{m} \sum_{j=1}^{m} \underbrace{\langle \boldsymbol{v}_i, \boldsymbol{v}_j \rangle_{\mathcal{U}}}_{\substack{=1 \text{ if } i=j, \\ =0 \text{ otherwise}}} \langle \boldsymbol{\eta}, \boldsymbol{v}_i \rangle_{\mathcal{U}} \langle \boldsymbol{u}, \boldsymbol{v}_j \rangle_{\mathcal{U}} \Big)
\end{aligned}$$

$$= \left\langle \boldsymbol{\eta}, 2\Big(\boldsymbol{u} - \sum_{i=1}^{m} \boldsymbol{v}_i \langle \boldsymbol{v}_i, \boldsymbol{u} \rangle_{\mathcal{U}}\Big) \right\rangle_{\mathcal{U}} . \tag{5.24}$$

Before we can wrap up the calculation of the derivative we have to return to a model with a non-zero mean. To this end we substitute $\boldsymbol{u}$ by $\boldsymbol{u} - \bar{\boldsymbol{w}}$, yielding

$$\begin{aligned}\frac{\mathsf{d}}{\mathsf{d}\epsilon}\mathcal{P}(\boldsymbol{u}+\epsilon\boldsymbol{\eta}) &= \left\langle \boldsymbol{\eta}, 2\left(\boldsymbol{u} - \bar{\boldsymbol{w}} - \sum_{i=1}^{m} \boldsymbol{v}_i \langle \boldsymbol{v}_i, \boldsymbol{u} - \bar{\boldsymbol{w}} \rangle_{\mathcal{U}}\right) \right\rangle_{\mathcal{U}} \\ &= \left\langle \boldsymbol{\eta}, 2\left(\boldsymbol{u} - \Big(\sum_{i=1}^{m} \boldsymbol{v}_i \langle \boldsymbol{v}_i, \boldsymbol{u} \rangle_{\mathcal{U}}\Big) - \Big(\bar{\boldsymbol{w}} - \sum_{i=1}^{m} \boldsymbol{v}_i \langle \boldsymbol{v}_i, \bar{\boldsymbol{w}} \rangle_{\mathcal{U}}\Big)\right) \right\rangle_{\mathcal{U}} \\ &= \left\langle \boldsymbol{\eta}, 2\left(\boldsymbol{u} - \Big(\sum_{i=1}^{m} \boldsymbol{v}_i \langle \boldsymbol{v}_i, \boldsymbol{u} \rangle_{\mathcal{U}}\Big) - \tilde{\boldsymbol{w}}\right) \right\rangle_{\mathcal{U}} \end{aligned} \tag{5.25}$$

$$\text{where} \quad \tilde{\boldsymbol{w}} = \bar{\boldsymbol{w}} - \sum_{i=1}^{m} \boldsymbol{v}_i \langle \boldsymbol{v}_i, \bar{\boldsymbol{w}} \rangle_{\mathcal{U}}, \tag{5.26}$$

This is possible as the model mean is a constant with respect to the function $\boldsymbol{u}$. The model mean $\bar{\boldsymbol{w}}$ is transformed into a constant offset $\tilde{\boldsymbol{w}}$ on the derivative, making it easier to handle. Using again (3.15), we can thus define

$$\nabla_{\boldsymbol{u}}\mathcal{P}(\boldsymbol{u})(\mathrm{x}) = 2\Big(\boldsymbol{u}(\mathrm{x}) - \sum_{i=1}^{m} \boldsymbol{v}_i(\mathrm{x}) \langle \boldsymbol{v}_i, \boldsymbol{u} \rangle_{\mathcal{U}}\Big) - 2\tilde{\boldsymbol{w}}. \tag{5.27}$$

Note that this term is identical to (5.21) except for the squared norm and the factor 2 in the derivative. This can be exploited in the numerical implementation of the regularizer to save computations.

Discretization and Optimization

The discretization can be directly obtained by replacing functions and the inner products with their discrete equivalents.

$$\begin{aligned}\mathcal{P}(\mathbf{u}) &= s^{-1}\left(\mathbf{u} - \bar{\mathbf{w}} - s^{-1}\sum_{i=1}^{m} \mathbf{v}_i \mathbf{v}_i^T (\mathbf{u} - \bar{\mathbf{w}})\right)^2 \\ &= s^{-1}\left((\mathbf{I} - s^{-1}\mathbf{V}\mathbf{V}^T)(\mathbf{u} - \bar{\mathbf{w}})\right)^2 \end{aligned} \tag{5.28}$$

$$\begin{aligned}\nabla_{\mathbf{u}}\mathcal{P}(\mathbf{u})(\mathbf{x}) &= 2\left(\Big(\mathbf{u} - s^{-1}\sum_{i=1}^{m} \mathbf{v}_i \mathbf{v}_i^T \mathbf{u}\Big) - \tilde{\mathbf{w}}\right) \\ &= 2\left((\mathbf{I} - s^{-1}\mathbf{V}\mathbf{V}^T)\mathbf{u} - \tilde{\mathbf{w}}\right) \end{aligned} \tag{5.29}$$

$$\tilde{\mathbf{w}} = s^{-1}(\mathbf{I} - \mathbf{V}\mathbf{V}^T)\bar{\mathbf{w}}, \tag{5.30}$$

where $\mathbf{V} = (\mathbf{v}_1, \ldots, \mathbf{v}_n)$. The s^{-1} preceding the sum here is due to the definition of the function scalar product we used which results in discrete Eigenvectors $\mathbf{v}_i$ with norm $s^{-1}\|\mathbf{v}_i\| = 1$.

As described above, the energy $\mathcal{P}$ is integrated straightforwardly as a penalty term into the registration energy (5.20), but adding it to the optimization algorithms is not quite as straightforward. The simplest solution is to treat the derivative of the PCA energy term $\nabla_\mathbf{u}\mathcal{P}$ in a way similarly to the distance measure. The semi-implicit gradient descent scheme from equation (4.2) then becomes

$$(\mathbf{I} + \tau\alpha\mathbf{A})\mathbf{u}^{(t+1)} = \mathbf{u}^{(t)} - \tau\left(\nabla_\mathbf{u}\mathcal{D}(F, M_{\mathbf{u}^{(t)}}) + 2\beta(\mathbf{I} - s^{-1}\mathbf{V}\mathbf{V}^T)\mathbf{u} - 2\beta\tilde{\mathbf{w}}\right), \quad (5.31)$$

where $\mathbf{A}$ is a placeholder for the linear operator resulting from the standard regularizer, i. e. $\mathbf{A}_\Delta$ (3.48) or $\mathbf{A}_{\Delta^2}$ (3.61). Similarly, for the Newton based methods, this results in

$$\begin{aligned}\mathbf{u}^{(t+1)} = \mathbf{u}^{(t)} - \tau &\left(\mathbf{H}_{\mathcal{D}(F,M_{\mathbf{u}^{(t)}})} + \beta\mathbf{H}_\mathcal{P} + \alpha\mathbf{A}\right)^{-1} \\ &\left(\nabla_\mathbf{u}\mathcal{D}(F, M_{\mathbf{u}^{(t)}}) + \alpha\mathbf{A}\mathbf{u}^{(t)} + 2\beta(\mathbf{I} - s^{-1}\mathbf{V}\mathbf{V}^T)\mathbf{u}^{(t)} - 2\beta\tilde{\mathbf{w}}\right). \quad (5.32)\end{aligned}$$

where

$$\mathbf{H}_\mathcal{P} = 2\left(\mathbf{I} - s^{-1}\mathbf{V}\mathbf{V}^T\right) \quad (5.33)$$

If we do not want to solve the large and densely populated $\mathbf{H}_\mathcal{P} = s^{-1}(\mathbf{I} - \mathbf{V}\mathbf{V}^T)$ this scheme can only be used with a numeric approximation for the Hessian of the overall energy. Based on the secant condition (4.7)(4.10) it is possible to calculate a rough estimate for $\mathbf{H}_{\mathcal{D}(F,M_{\mathbf{u}^{(t)}})} + \beta\mathbf{H}_\mathcal{P}$. While this method seems to work well enough in practice, synthetic examples showed that they are limited. For instance a synthetic test case proved impossible to optimize, due to the descent directions being so bad that the step size control essentially stopped any progress. In this test case a circle had to be registered to a box, with a high weighted PCA regularizer containing only translational components $\boldsymbol{t}_j$. Cases like this seem to require that the PCA regularizer is treated together with the standard regularizer. As the system matrix of the PCA $\mathbf{V}\mathbf{V}^T$ is a linear operator of rather low rank it is possible to solve for the linear system arising from the combination of the standard regularizer and the PCA term. Doing so for the semi-implicit gradient descent results in

$$\begin{aligned}(\mathbf{I} + \tau\alpha\mathbf{A} + 2\tau\beta(\mathbf{I} - s^{-1}\mathbf{V}\mathbf{V}^T))\mathbf{u}^{(t+1)} &= \mathbf{u}^{(t)} - \tau(\nabla_\mathbf{u}\mathcal{D}(F, M_{\mathbf{u}^{(t)}}) - 2\beta\tilde{\mathbf{w}}) \\ ((1 + 2\tau\beta)\mathbf{I} + \tau\alpha\mathbf{A} - 2\tau\beta s^{-1}\mathbf{V}\mathbf{V}^T)\mathbf{u}^{(t+1)} &= \mathbf{u}^{(t)} - \tau(\nabla_\mathbf{u}\mathcal{D}(F, M_{\mathbf{u}^{(t)}}) - 2\beta\tilde{\mathbf{w}}). \quad (5.34)\end{aligned}$$

This means one has to solve for the matrix $(1 + 2\tau\beta)\mathbf{I} + \tau\alpha\mathbf{A} - 2\tau\beta\mathbf{V}\mathbf{V}^T$. As the regularizer matrix $\mathbf{A}$ and the PCA system matrix $\mathbf{V}\mathbf{V}^T$ are symmetric and positive semi-definite, and the identity $\mathbf{I}$ is symmetric and positive definite, the whole system is symmetric and positive definite (compare Appendix A). It can thus be solved by the application of a Krylov subspace method like Conjugate Gradient (CG), which requires only an implementation of the multiplication with the system matrix. The multiplication with the matrix $\mathbf{V}\mathbf{V}^T$ can be efficiently implemented as a sum of inner products $\mathbf{V}\mathbf{V}^T\mathbf{u} = \sum_{i=1}^{m} \mathbf{v}_i(\mathbf{v}_i^T\mathbf{u})$. The same can be applied for the Newton based methods, with a system matrix of $\mathbf{H}_{\text{Dist}} + \alpha\mathbf{A} + 2\beta(\mathbf{I} - \mathbf{V}\mathbf{V}^T)$.

Another approach to solve for the linear system containing the added PCA matrix $\mathbf{V}\mathbf{V}^T$ is based on the Sherman-Morrison-Woodbury formula (see e. g. [Pete 08]). This formula allows to rewrite the inverse of a matrix that is composed of the sum of a full rank matrix and a low rank update.

$$(\mathbf{E} + a\mathbf{F}\mathbf{F}^T)^{-1} = \mathbf{E}^{-1}(a^{-1}\mathbf{I} + \mathbf{F}(\mathbf{I} - \mathbf{F}^T\mathbf{E}^{-1}\mathbf{F})^{-1}\mathbf{F}^T\mathbf{E}^{-1}), \tag{5.35}$$

with $\mathbf{E} \in \mathbb{R}^{n\times n}$ and $\mathbf{F} \in \mathbb{R}^{n\times m}$, $m \leq n$. The advantage of the reformulation is that the inverse of $(\mathbf{I} - \mathbf{F}^T\mathbf{E}^{-1}\mathbf{F})$ containing the low rank matrix $\mathbf{F}$ has only a size of $\mathbb{R}^{m\times m}$. If m is small compared to n this inverse is computationally cheap. The inversion of the original linear system can thus be reduced to solving several times for $\mathbf{E}$, which can be an advantage if $\mathbf{E}$ is sparse or otherwise easy to solve for. In the case of the semi-implicit gradient descent we can associate

$$\mathbf{E} = (1 + 2\tau\beta)\mathbf{I} + \tau\alpha\mathbf{A} \qquad \mathbf{F} = \mathbf{V} \qquad a = -2\tau\beta s^{-1}. \tag{5.36}$$

The linear system $\mathbf{E}$ is thus equivalent to the linear system arising from the application of the standard regularizer without the PCA. To evaluate (5.35) with these associations it is necessary to solve for $\mathbf{E}$ m times in order to evaluate $\mathbf{E}^{-1}\mathbf{F}$ and an additional 2 times to calculate the final result. If $\mathbf{E}$ does not change in between iterations i. e. if the step size is not changed then the result of $\mathbf{E}^{-1}\mathbf{F}$ can be cached. Otherwise it has to be recomputed in each iteration. This approach is, therefore, computationally quite demanding. The main advantage is that it allows to employ the FFT based direct solver for the solution of the standard regularizer matrix. Having a direct solver available for this kind of problem is advantageous, as it allows to assess whether this implicit approach has a real advantage. If the system matrix is instead solved by an iterative solver like conjugate gradient we are faced with the problem of having to trade off between accuracy and runtime, and can never be fully certain that the non-linear optimization could not have performed better if the result of the linear problem had been more accurate.

In the case of the Newton based non-linear solver we would have to associate

$$\mathbf{E} = \mathbf{H}_{\text{Dist}} + 2\beta\mathbf{I} + \alpha\mathbf{A} \qquad \mathbf{F} = \mathbf{V} \qquad a = -2\beta s^{-1}. \tag{5.37}$$

Here, the formulation is computationally even more disadvantageous as $\mathbf{H}_{\text{Dist}}$ and therefore $\mathbf{E}$ change in every step. This makes a reuse of $\mathbf{E}^{-1}\mathbf{F}^T$ in the next iteration impossible.

5.2.3 PCA Curvature Regularization

A different approach to a translation invariant deformation model is to generate the model not on the deformations themselves but instead on their derivatives. The first derivative of a deformation is already invariant to translations. As discussed in Section 3.4.2 the Laplacian of the deformation field $\Delta\boldsymbol{u}$ is invariant against all affine transforms. This makes the model completely robust to inconsistencies in the rigid alignment. As one of our standard regularizers is already based on Laplacian term we also use the Laplacians of the deformation fields as the basis for the PCA model. Please note that the invariance to rotation and translations in the model, does not altogether eliminate the need for a rigid pre-registration. The pre-registration is still required in order to give a good starting position to the non-rigid registration. However, if there is a need during the optimization of the non-rigid registration for a rigid component in the transform $\boldsymbol{u}$ it is not penalized by the model.

With the mean and the Eigenmodes $\boldsymbol{v}_i$ computed on the Laplacians of the learning data $\Delta\boldsymbol{w}$, the PCA becomes

$$\hat{\boldsymbol{v}}_i(\mathbf{x}) = \underset{\boldsymbol{v}_i}{\operatorname{argmax}} \sum_{j=1}^{m} \langle \boldsymbol{v}_i, \Delta\boldsymbol{w}_j - \Delta\bar{\boldsymbol{w}} \rangle_{\mathcal{U}}^2 \qquad \text{(maximum variation)} \tag{5.38}$$

$$\text{with} \qquad \bar{\boldsymbol{w}}(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{w}_i(\mathbf{x}) \qquad \text{(mean)} \tag{5.39}$$

$$\text{subject to} \tag{5.40}$$

$$\|\boldsymbol{v}_i\|_{\mathcal{U}}^2 = \langle \boldsymbol{v}_i, \boldsymbol{v}_i \rangle_{\mathcal{U}} = 1 \qquad \text{(normal length)} \tag{5.41}$$

$$\langle \boldsymbol{v}_i, \boldsymbol{v}_j \rangle_{\mathcal{U}} = 0 \quad \forall i \neq j \qquad \text{(orthogonality).} \tag{5.42}$$

The resulting regularization energy is defined similarly to (5.21) as

$$\boldsymbol{u}^* = \underset{\boldsymbol{u}}{\operatorname{argmin}}\, \mathcal{E}(F, M, \boldsymbol{u}) = \mathcal{D}(F, M_{\boldsymbol{u}}) + \alpha\mathcal{R}_{\text{Curv}}(\boldsymbol{u}) + \beta\mathcal{P}_{\Delta}(\boldsymbol{u}) \tag{5.43}$$

$$\begin{aligned}\mathcal{P}_{\Delta}(\boldsymbol{u}) &= \left\| \Delta\boldsymbol{u}(\mathbf{x}) - \Big(\Delta\bar{\boldsymbol{w}} + \sum_{i=1}^{m} \boldsymbol{v}_i(\mathbf{x}) \langle \boldsymbol{v}_i, \Delta\boldsymbol{u} - \Delta\bar{\boldsymbol{w}} \rangle_{\mathcal{U}}\Big) \right\|_{\mathcal{U}}^2 \\ &= \frac{1}{|\Omega|} \int_{\Omega} \left(\Delta\boldsymbol{u}(\mathbf{x}) - \Big(\Delta\bar{\boldsymbol{w}} + \sum_{i=1}^{m} \boldsymbol{v}_i(\mathbf{x}) \langle \boldsymbol{v}_i, \Delta\boldsymbol{u} - \Delta\bar{\boldsymbol{w}} \rangle_{\mathcal{U}}\Big) \right)^2 \mathsf{dx}. \end{aligned} \tag{5.44}$$

For this approach we only consider the curvature regularizer as standard regularizer, as this will allow some simplifications later on. For the calculation of the derivative, we again make the simplifying assumption that the model has a zero mean, i. e. $\bar{\boldsymbol{w}} = \Delta\bar{\boldsymbol{w}} = \mathbf{0}$. For the application of the PCA to the Laplacian of the deformation function we will see during the calculation of the Gâteaux derivative that it is necessary to impose von Neumann type boundary condition (3.44)(3.56) on the functions of $\mathcal{U}$, similarly to what is needed for the curvature regularizer. Note that as the $\boldsymbol{v}_i$ are composed of Laplacians of the training data $\boldsymbol{w}_j$, generated with a curvature regularizer based registration method, these deformations also have the von Neumann boundary condition (3.44) imposed on them. It therefore holds that

$$\boldsymbol{n}(\mathbf{x})^T\, \nabla\boldsymbol{v}(\mathbf{x}) = \big(\boldsymbol{n}(\mathbf{x})^T\, \nabla v_1(\mathbf{x}), \ldots, \boldsymbol{n}(\mathbf{x})^T\, \nabla v_d(\mathbf{x})\big)^T = \mathbf{0} \quad \forall \mathbf{x} \in \partial\Omega. \tag{5.45}$$

To derive the Euler-Lagrange equations it is again necessary to consider the variation of $\mathcal{P}_\vartriangle$ with respect to $\epsilon\boldsymbol{\eta}$.

$$\begin{aligned}
&\frac{\mathsf{d}}{\mathsf{d}\epsilon}\mathcal{P}_\vartriangle(\boldsymbol{u}+\epsilon\boldsymbol{\eta})\bigg|_{\epsilon=0} = \\
&= \frac{2}{|\Omega|}\int_\Omega \left(\Delta\boldsymbol{\eta}(\mathbf{x}) - \sum_{i=1}^m \boldsymbol{v}_i(\mathbf{x})\langle\boldsymbol{v}_i,\Delta\boldsymbol{\eta}\rangle_\mathcal{U}\right)^T \\
&\qquad\qquad \left(\Delta\boldsymbol{u}(\mathbf{x}) - \sum_{i=1}^m \boldsymbol{v}_i(\mathbf{x})\langle\boldsymbol{v}_i,\Delta\boldsymbol{u}\rangle_\mathcal{U}\right)\,\mathsf{dx} \\
&= \frac{2}{|\Omega|}\int_\Omega (\Delta\boldsymbol{\eta}(\mathbf{x}))^T \left(\Delta\boldsymbol{u}(\mathbf{x}) - \sum_{i=1}^m \boldsymbol{v}_i(\mathbf{x})\langle\boldsymbol{v}_i,\Delta\boldsymbol{u}\rangle_\mathcal{U}\right)\,\mathsf{dx} \\
&\quad -\frac{2}{\Omega}\int_\Omega \left(\sum_{i=1}^m \boldsymbol{v}_i(\mathbf{x})\langle\boldsymbol{v}_i,\Delta\boldsymbol{\eta}\rangle_\mathcal{U}\right)^T \left(\Delta\boldsymbol{u}(\mathbf{x}) - \sum_{i=1}^m \boldsymbol{v}_i(\mathbf{x})\langle\boldsymbol{v}_i,\Delta\boldsymbol{u}\rangle_\mathcal{U}\right)\,\mathsf{dx} \qquad (5.46)
\end{aligned}$$

Before the test function $\boldsymbol{\eta}$ can be isolated its derivatives have to be eliminated. This is done similarly to the derivation of the curvature regularizer (see Section 3.4.2), by applying Green's theorem and making use of the boundary conditions imposed on $\boldsymbol{\eta}$, $\boldsymbol{u}$ and $\boldsymbol{v}_i$. First the inner product of the second summand is treated this way. With $\boldsymbol{v}_i = (v_{i,1},\dots,v_{i,d})$, we get

$$\begin{aligned}
\langle\boldsymbol{v}_i,\Delta\boldsymbol{\eta}\rangle_\mathcal{U} &= \frac{1}{|\Omega|}\int_\Omega \sum_{j=1}^d v_{i,j}(\mathbf{y})\Delta\eta_j(\mathbf{y})\,\mathsf{dy} \\
&= \frac{1}{|\Omega|}\sum_{j=1}^d \left(\int_{\partial\Omega} v_{i,j}(\mathbf{y})\underbrace{(\nabla\eta_j(\mathbf{y}))^T\boldsymbol{n}(\mathbf{y})}_{=0 \text{ by } (3.44)}\,\mathsf{dy} - \int_\Omega (\nabla v_{i,j}(\mathbf{y}))^T\nabla\eta_j(\mathbf{y})\,\mathsf{dy}\right) \\
&= \frac{1}{|\Omega|}\sum_{j=1}^d \left(-\int_{\partial\Omega} \eta_j(\mathbf{y})\underbrace{(\nabla v_{i,j}(\mathbf{y}))^T\boldsymbol{n}(\mathbf{y})}_{=0 \text{ by } (5.45)}\,\mathsf{dy} + \int_\Omega (\Delta v_{i,j}(\mathbf{y}))\eta_j(\mathbf{y})\,\mathsf{dy}\right) \\
&= \langle\Delta\boldsymbol{v}_i,\boldsymbol{\eta}\rangle_\mathcal{U}. \qquad (5.47)
\end{aligned}$$

In a second step, the same is applied to the first part of the summand.

$$\begin{aligned}
&\int_\Omega (\Delta\boldsymbol{\eta}(\mathbf{x}))^T \left(\Delta\boldsymbol{u}(\mathbf{x}) - \sum_{i=1}^m \boldsymbol{v}_i(\mathbf{x})\langle\boldsymbol{v}_i,\Delta\boldsymbol{u}\rangle_\mathcal{U}\right)\,\mathsf{dx} \\
&= \sum_{j=1}^d \int_\Omega (\Delta\eta_j(\mathbf{x}))\left(\Delta u_j(\mathbf{x}) - \sum_{i=1}^m v_{i,j}(\mathbf{x})\langle\boldsymbol{v}_i,\Delta\boldsymbol{u}\rangle_\mathcal{U}\right)
\end{aligned}$$

$$= \sum_{j=1}^{d} \Bigg(\int_{\partial\Omega} \underbrace{(\nabla \eta_j(\mathbf{x}))^T \boldsymbol{n}(\mathbf{x})}_{=0 \text{ by } (3.44)} \left(\Delta u_j(\mathbf{x}) - \sum_{i=1}^{m} v_{i,j}(\mathbf{x}) \langle \boldsymbol{v}_i, \Delta \boldsymbol{u} \rangle_{\mathcal{U}} \right) \, \mathsf{dx}$$
$$- \int_{\Omega} (\nabla \eta_j(\mathbf{x}))^T \left(\nabla \Delta u_j(\mathbf{x}) - \sum_{i=1}^{m} \nabla v_{i,j}(\mathbf{x}) \langle \boldsymbol{v}_i, \Delta \boldsymbol{u} \rangle_{\mathcal{U}} \right) \, \mathsf{dx} \Bigg)$$

$$= \sum_{j=1}^{d} \Bigg(- \int_{\partial\Omega} \eta_j(\mathbf{x}) \Big(\underbrace{(\nabla \Delta u_j(\mathbf{x}))^T \boldsymbol{n}(\mathbf{x})}_{=0 \text{ by } (3.56)} - \sum_{i=1}^{m} \underbrace{(\nabla v_{i,j}(\mathbf{x}))^T \boldsymbol{n}(\mathbf{x})}_{=0 \text{ by } (5.45)} \langle \boldsymbol{v}_i, \Delta \boldsymbol{u} \rangle_{\mathcal{U}} \Big) \, \mathsf{dx}$$
$$+ \int_{\Omega} \eta_j(\mathbf{x}) \left(\Delta^2 u_j(\mathbf{x}) - \sum_{i=1}^{m} \Delta v_{i,j}(\mathbf{x}) \langle \boldsymbol{v}_i, \Delta \boldsymbol{u} \rangle_{\mathcal{U}} \right) \, \mathsf{dx} \Bigg)$$

$$= \int_{\Omega} \boldsymbol{\eta}(\mathbf{x})^T \left(\Delta^2 \boldsymbol{u}(\mathbf{x}) - \sum_{i=1}^{m} \Delta \boldsymbol{v}_i(\mathbf{x}) \langle \boldsymbol{v}_i, \Delta \boldsymbol{u} \rangle_{\mathcal{U}} \right) \, \mathsf{dx} \tag{5.48}$$

Substituting the results of (5.47) and (5.48) into (5.46) we can now continue to isolate the test function $\boldsymbol{\eta}$.

$$\frac{\mathsf{d}}{\mathsf{d}\epsilon} \mathcal{P}_{\Delta}(\boldsymbol{u} + \epsilon \boldsymbol{\eta}) \bigg|_{\epsilon=0}$$

$$= \frac{2}{|\Omega|} \int_{\Omega} (\boldsymbol{\eta}(\mathbf{x}))^T \left(\Delta^2 \boldsymbol{u}(\mathbf{x}) - \sum_{i=1}^{m} \Delta \boldsymbol{v}_i(\mathbf{x}) \langle \boldsymbol{v}_i, \Delta \boldsymbol{u} \rangle_{\mathcal{U}} \right) \, \mathsf{dx}$$
$$- \frac{2}{|\Omega|} \int_{\Omega} \left(\sum_{i=1}^{m} \boldsymbol{v}_i(\mathbf{x}) \langle \Delta \boldsymbol{v}_i, \boldsymbol{\eta} \rangle_{\mathcal{U}} \right)^T \left(\Delta \boldsymbol{u}(\mathbf{x}) - \sum_{i=1}^{m} \boldsymbol{v}_i(\mathbf{x}) \langle \boldsymbol{v}_i, \Delta \boldsymbol{u} \rangle_{\mathcal{U}} \right) \, \mathsf{dx}$$

$$= 2 \langle \boldsymbol{\eta}, \Delta^2 \boldsymbol{u} \rangle_{\mathcal{U}} - \left(\sum_{i=1}^{m} \langle \boldsymbol{\eta}, \Delta \boldsymbol{v}_i \rangle_{\mathcal{U}} \langle \boldsymbol{v}_i, \Delta \boldsymbol{u} \rangle_{\mathcal{U}} \right) - 2 \left(\sum_{i=1}^{m} \langle \Delta \boldsymbol{v}_i, \boldsymbol{\eta} \rangle_{\mathcal{U}} \langle \Delta \boldsymbol{u}, \boldsymbol{v}_i \rangle_{\mathcal{U}} \right)$$
$$+ 2 \left(\sum_{i=1}^{m} \sum_{j=1}^{m} \underbrace{\langle \boldsymbol{v}_i, \boldsymbol{v}_j \rangle_{\mathcal{U}}}_{\substack{=1 \text{ if } i=j, \\ =0 \text{ otherwise}}} \langle \boldsymbol{v}_i, \Delta \boldsymbol{u} \rangle_{\mathcal{U}} \langle \boldsymbol{\eta}, \Delta \boldsymbol{v}_j \rangle_{\mathcal{U}} \right)$$

$$= 2 \left\langle \boldsymbol{\eta}, \Delta^2 \boldsymbol{u} - \sum_{i=1}^{m} \Delta \boldsymbol{v}_i \langle \boldsymbol{v}_i, \Delta \boldsymbol{u} \rangle_{\mathcal{U}} \right\rangle_{\mathcal{U}} \tag{5.49}$$

As a last step, it is necessary to substitute $\boldsymbol{u}$ by $\boldsymbol{u} - \bar{\boldsymbol{w}}$ to return to a non-zero mean

PCA model.

$$\begin{aligned}\frac{\mathsf{d}}{\mathsf{d}\epsilon}\mathcal{P}_{\triangle}(\boldsymbol{u}+\epsilon\boldsymbol{\eta})\bigg|_{\epsilon=0} &= \left\langle \boldsymbol{\eta}, 2\left(\Delta^2(\boldsymbol{u}-\bar{\boldsymbol{w}}) - \sum_{i=1}^{m}\Delta\boldsymbol{v}_i\langle\boldsymbol{v}_i,\Delta(\boldsymbol{u}-\bar{\boldsymbol{w}})\rangle_{\mathcal{U}}\right)\right\rangle_{\mathcal{U}} \\ &= \left\langle \boldsymbol{\eta}, 2\left(\Delta^2\boldsymbol{u} - \sum_{i=1}^{m}\Delta\boldsymbol{v}_i\langle\boldsymbol{v}_i,\Delta\boldsymbol{u}\rangle_{\mathcal{U}}\right) - 2\left(\Delta^2\bar{\boldsymbol{w}} - \sum_{i=1}^{m}\Delta\boldsymbol{v}_i\langle\boldsymbol{v}_i,\Delta\bar{\boldsymbol{w}}\rangle_{\mathcal{U}}\right)\right\rangle_{\mathcal{U}} \\ &= \left\langle \boldsymbol{\eta}, 2\left(\Delta^2\boldsymbol{u} - \Big(\sum_{i=1}^{m}\Delta\boldsymbol{v}_i\langle\boldsymbol{v}_i,\Delta\boldsymbol{u}\rangle_{\mathcal{U}}\Big) - \tilde{\boldsymbol{w}}\right)\right\rangle_{\mathcal{U}} \qquad (5.50)\end{aligned}$$

$$\tilde{\boldsymbol{w}} = 2\Big(\Delta^2\bar{\boldsymbol{w}} - \sum_{i=1}^{m}\Delta\boldsymbol{v}_i\langle\boldsymbol{v}_i,\Delta\bar{\boldsymbol{w}}\rangle_{\mathcal{U}}\Big)$$

Finally, with this simplified version of the Gâteaux derivative, we can identify $\nabla_{\boldsymbol{u}}\mathcal{P}_{\triangle}$ by making use of the inner product.

$$\nabla_{\boldsymbol{u}}\mathcal{P}_{\triangle}(\boldsymbol{u})(\mathbf{x}) = 2\Big(\Delta^2\boldsymbol{u}(\mathbf{x}) - \sum_{i=1}^{m}\Delta\boldsymbol{v}_i(\mathbf{x})\langle\boldsymbol{v}_i,\Delta\boldsymbol{u}\rangle_{\mathcal{U}} - \tilde{\boldsymbol{w}}\Big). \qquad (5.51)$$

Discretization and Optimization

The measure and its derivative is discretized by replacing the differential operators Δ and Δ^2 with their respective discrete versions $\mathbf{A}_{\triangle}$ (3.48) and $\mathbf{A}_{\triangle^2}$ (3.61) that were introduced in Section 3.4. Combined with the discretized inner product this yields

$$\begin{aligned}\mathcal{P}_{\triangle}(\mathbf{u}) &= \left(\mathbf{A}_{\triangle}(\mathbf{u}-\bar{\mathbf{w}}) - s^{-1}\sum_{i}\mathbf{v}\mathbf{v}^T\mathbf{A}_{\triangle}(\mathbf{u}-\bar{\mathbf{w}})\right)^2 \\ &= \big((\mathbf{I}-s^{-1}\mathbf{V}\mathbf{V}^T)\mathbf{A}_{\triangle}(\mathbf{u}-\bar{\mathbf{w}})\big)^2 \qquad (5.52)\end{aligned}$$

$$\begin{aligned}\nabla_{\mathbf{u}}\mathcal{P}_{\triangle}(\mathbf{u})(\mathbf{x}) &= 2\left(\mathbf{A}_{\triangle^2}\mathbf{u} - s^{-1}\Big(\sum_{i}\mathbf{A}_{\triangle}\mathbf{v}\mathbf{v}^T\mathbf{A}_{\triangle}\mathbf{u}\Big) - \tilde{\mathbf{w}}\right) \\ &= 2\left(\mathbf{A}_{\triangle^2}\mathbf{u} - s^{-1}\mathbf{A}_{\triangle}\mathbf{V}\mathbf{V}^T\mathbf{A}_{\triangle}\mathbf{u} - \tilde{\mathbf{w}}\right) \\ &= 2\left(\mathbf{A}_{\triangle^2}\mathbf{u} - s^{-1}\tilde{\mathbf{V}}\tilde{\mathbf{V}}^T\mathbf{u} - \tilde{\mathbf{w}}\right) \qquad (5.53)\end{aligned}$$

$$\tilde{\mathbf{w}} = (\mathbf{I}-s^{-1}\mathbf{V}\mathbf{V}^T)\mathbf{A}_{\triangle}(\mathbf{u}-\bar{\mathbf{w}}).$$

where $\mathbf{V} = (\mathbf{v}_1, \ldots, \mathbf{v}_n)$ and accordingly $\tilde{\mathbf{V}} = \mathbf{A}_\triangle \mathbf{V} = (\mathbf{A}_\triangle \mathbf{v}_1, \ldots, \mathbf{A}_\triangle \mathbf{v}_n)$. If $\mathbf{A}_{\triangle^2}$ is realized as $\mathbf{A}_{\triangle^2} = \mathbf{A}_\triangle \mathbf{A}_\triangle$, then (5.53) can also be rewritten as

$$\nabla_{\mathbf{u}} \mathcal{P}_\triangle(\mathbf{u})(\mathbf{x}) = 2\mathbf{A}_\triangle \left(\mathbf{I} - s^{-1}\mathbf{V}\mathbf{V}^T\right) \mathbf{A}_\triangle \mathbf{u} - 2\tilde{\mathbf{w}} \tag{5.54}$$

The integration into the optimization algorithms is performed, to some extent, analogously to the integration of the PCA regularizer. For the semi-implicit gradient descent, with the PCA curvature regularizer added explicitly, this leads to the following formulation.

$$(\mathbf{I} + \tau\alpha\mathbf{A}_{\triangle^2})\mathbf{u}^{(t+1)} = \mathbf{u}^{(t)} - \tau\left(\nabla_{\mathbf{u}}\mathcal{D}(F, M_{\mathbf{u}^{(t)}}) + 2\beta(\mathbf{A}_{\triangle^2} - s^{-1}\tilde{\mathbf{V}}\tilde{\mathbf{V}}^T)\mathbf{u}^{(t)} - \tilde{\mathbf{w}}\right). \tag{5.55}$$

The same can be applied for the Newton type methods

$$\begin{aligned}\mathbf{u}^{(t+1)} = \mathbf{u}^{(t)} - \tau & \left(\mathbf{H}_{\mathcal{D}(F, M_{\mathbf{u}^{(t)}})} + \beta\mathbf{H}_{\mathcal{P}_\triangle} + \alpha\mathbf{A}\right)^{-1} \\ & \left(\nabla_{\mathbf{u}}\mathcal{D}(F, M_{\mathbf{u}^{(t)}}) + \alpha\mathbf{A}\mathbf{u}^{(t)} + (\mathbf{A}_{\triangle^2} - s^{-1}\tilde{\mathbf{V}}\tilde{\mathbf{V}}^T)\mathbf{u}^{(t)} - \tilde{\mathbf{w}}\right),\end{aligned} \tag{5.56}$$

where

$$\mathbf{H}_{\mathcal{P}_\triangle} = 2\left(\mathbf{A}_{\triangle^2} - s^{-1}\tilde{\mathbf{V}}\tilde{\mathbf{V}}^T\right). \tag{5.57}$$

For the formulation of the semi-implicit gradient descent that handles the curvature PCA term implicitly, we get

$$\begin{aligned}\left(\mathbf{I} + \tau(\alpha\mathbf{A}_{\triangle^2} + 2\beta(\mathbf{A}_{\triangle^2} - s^{-1}\tilde{\mathbf{V}}\tilde{\mathbf{V}}^T))\right)\mathbf{u}^{(t+1)} &= \mathbf{u}^{(t)} - \tau(\nabla_{\mathbf{u}}\mathcal{D}(F, M_{\mathbf{u}^{(t)}}) - \tilde{\mathbf{w}}) \\ \left(\mathbf{I} + \tau(\alpha + 2\beta)\mathbf{A}_{\triangle^2} - 2\tau\beta s^{-1}\tilde{\mathbf{V}}\tilde{\mathbf{V}}^T\right)\mathbf{u}^{(t+1)} &= \mathbf{u}^{(t)} - \tau(\nabla_{\mathbf{u}}\mathcal{D}(F, M_{\mathbf{u}^{(t)}}) - \tilde{\mathbf{w}}).\end{aligned} \tag{5.58}$$

As far as the optimization algorithms are concerned this is qualitatively equivalent to the formulations presented for the PCA regularization. The same is true for the approach making use of the Sherman-Morrison-Woodbury formula (5.35) to solve for the matrices incorporating the PCA core matrix $\tilde{\mathbf{V}}\tilde{\mathbf{V}}^T$. For the semi-implicit algorithm we would associate

$$\mathbf{E} = \mathbf{I} + \tau\alpha(\alpha + 2\beta)\mathbf{A}_{\triangle^2} \qquad \mathbf{F} = \tilde{\mathbf{V}}^T \qquad a = -2\tau\beta s^{-1}, \tag{5.59}$$

for the Newton based methods

$$\mathbf{E} = \mathbf{H}_{\text{Dist}} + (\alpha + 2\beta)\mathbf{A} \qquad \mathbf{F} = \tilde{\mathbf{V}} \qquad a = -\beta s^{-1}. \tag{5.60}$$

Especially in the case of a regularization term that, analogously to the curvature regularizer, is based on the Laplacians of the vector field, it is a reasonable expectation that an implicit treatment of this term is advantageous.

Chapter 6

Application: MR/PET Attenuation Correction

The introduction of hybrid scanners, for example the combination of PET and CT imaging within one machine, has brought many new possibilities to the field of medical imaging, such as the invention of highly specific tumor markers. However, the superior tissue contrast and the large variety of different sequences offered by MR imaging make it desirable to have such hybrid systems with an MR scanner instead of the CT. Although there are technical difficulties to overcome, combined MR/PET scanners for the human head have been studied for a while now and currently the first full body scanners are undergoing clinical trials.

One of the problems posed by the combination of PET and MR is the attenuation correction of the PET image (see Section 2.3 and 2.4). In PET the aim is to measure the concentration of a radioactive tracer within the patient body. However, the quantity that the machine can actually measure is the radiation emitted by the tracer and attenuated by the patient body. In order to correct for the attenuation of the measured radiation it is necessary to provide an attenuation map for each acquisition. The map can be created, for instance, from a CT, where the relation between the energy dependent Hounsfield units and the tissue densities is known.

The values measured by MR, however, are not related to the attenuation or the tissue density, therefore, no straightforward solution is currently available for the attenuation correction in case of a hybrid MR/PET scanner. To generate attenuation maps from MR images there are two main categories of approaches. Segmentation based approaches try to segment the MR image into different tissue classes (usually air, soft tissue and bone) with known attenuation values. Registration based approaches non-rigidly register an atlas CT to the patient MR image, thus creating a pseudo CT for the patient. A survey of MR/PET attenuation techniques can be found in [Hofm 09]. In addition, combined methods can be applied that first perform an atlas registration and use the knowledge from the registered atlas as an additional input to the classification step to get overall improved results [Hofm 08].

Some recent methods instead focus on ultra short echo (UTE) sequences for the MR/PET attenuation correction, as these are to some extent able to image bone in MR. For instance, Keereman et al. [Keer 10] use two UTE sequences with different echo times to generate air, bone and soft tissue masks by an approach composed of thresholding and combinations of the resulting masks. They compare their results to also thresholded ground truth CT data and report a sensitivity of 89% for bone, 91% for soft tissue and 82% for air on phantom data. On real patient data they achieve a (quote) "overall accuracy of between 85% and 95% for all patients", which we assume refers to the sensitivity to all tissue classes.

In this work we will concern ourselves only with an improvement of the atlas registration. The multi-modal, non-rigid registration, which is used to perform such an atlas registration, offers many degrees of freedom in the spatial domain and also in the mapping of corresponding intensities, which is not known a priori. We therefore employ the statistical regularization of the deformation field presented in Section 5.2. The model we use in this application is generated from gold standard deformation fields computed on mono-modal CT registrations. The model trained on these mono-modal registration results is then used to constrain the more difficult multi-modal registration problem. The training CT data is also used to generate good atlas image for use during the atlas registrations.

In the following two sections we first present the generation of the atlas, the model and the choice of parameters during their generation, followed by an evaluation of the atlas registration approach using a standard, a PCA constrained and a curvature PCA constrained registration method.

6.1 Model Generation

6.1.1 Data

For this evaluation data of 34 different patients is used. Data from 9 patients is taken from the RIRE database [West 97]. The remaining patient data is of epilepsy patients who had a PET/CT scan and a varying number and type of MR sequences taken. All in all, 34 CT scans, 25 T1 weighted and 17 T2 weighted MR scans are used during the evaluation.

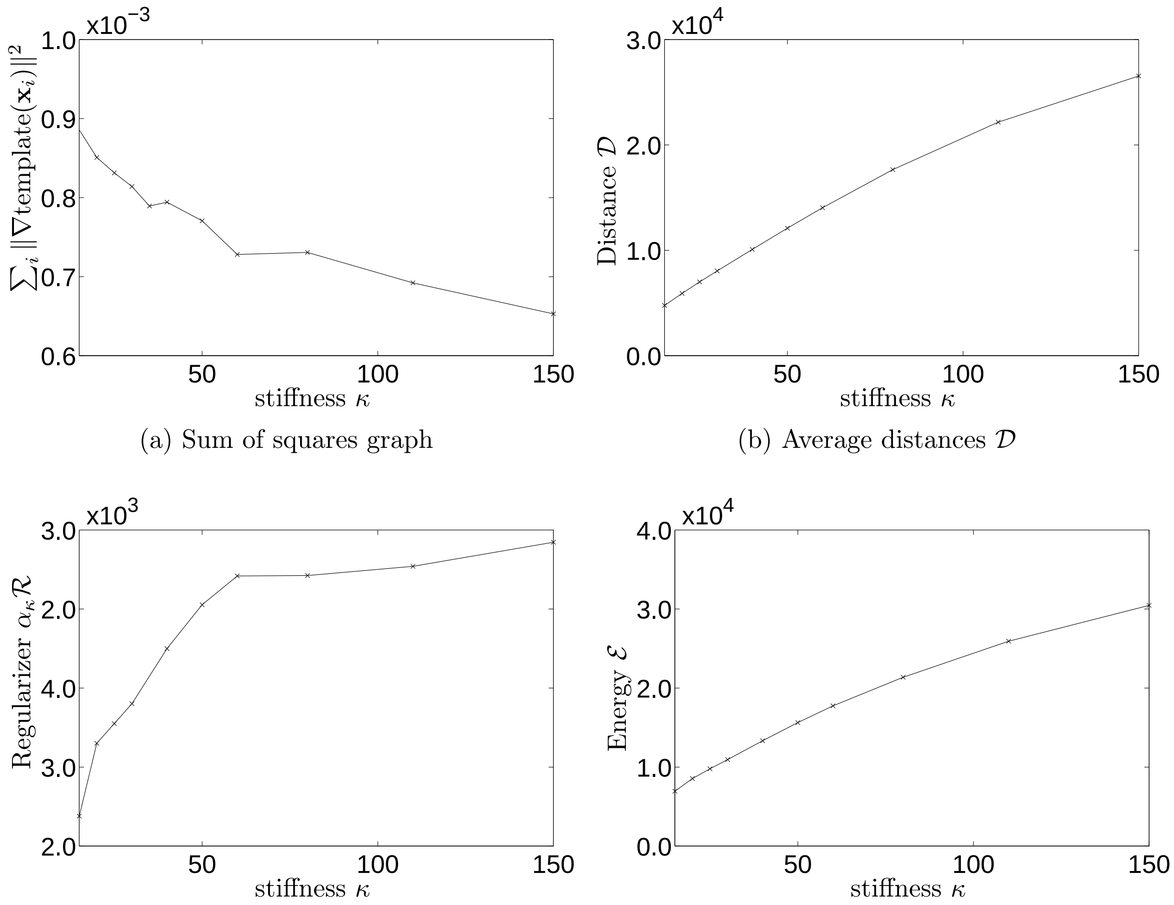
All the scans (CT and MR) are downsampled to an isotropic resolution of $1.95mm$ and volume size of $128 \times 128 \times 71$. The downsampling is performed in order to keep computation times down, safe memory during the later application of the PCA model for the multi-modal registration and to make additional resampling during the application of the PCA model unnecessary. The used resolution is sufficient for the application in MR/PET attenuation correction, as PET images that were acquired together with the CT image used in this evaluation had an isotropic spatial resolution of $2mm$. Additional structures in the images, like the table in CT images, are masked and ignored during processing (template image generation, model generation, evaluation).

6.1.2 Template Image Generation

The first step in the generation of a model is the computation of the image that will be registered to the patient MR images, which we will refer to as template image. In [Daum 09] we simply used another patients CT image. However, this is not ideal as, even though CT offers the normed Hounsfield unit (HU) for denoting image intensities, there are significant differences of the values in similar tissue types between different patients. For example the Hounsfield value of bone varies between 800 and 1200 HU in our data, depending on the general bone density of the patients. In order to generate a good template from the available data we therefore register (mono-modal) all datasets to one manually picked reference that is used as the fixed image in the non-rigid registration. All non-rigid registrations are preceded by a rigid pre-registration. The fixed image and the resulting deformed moving images from all other datasets are averaged in this common frame of reference. This mean image is then used as the template for the MR/PET attenuation correction.

As side note, it has to be mentioned that in contrast to the generation of the PCA model, the template image is not generated in a leave-one-out manner. The ground-truth CT image of the patient for most of our datasets is thus included in the averaging of the template image. This choice has been made to reduce some of the workload in the leave-one-out evaluation as it means that the gold-standard mono-modal registrations have to be performed only once as long as the template image does not change. As it is only one of 34 images that are averaged and the main aim of this evaluation is to compare the registration with and without the PCA regularizer, the effects should be minimal.

The result of the averaging naturally depends on the stiffness used during the non-rigid registrations. If the stiffness is chosen high, bad matches and therefore blurry averaged template images are generated. If the stiffness is chosen low, the registration has too much freedom and the reference image is more or less just replicated. In Figure 6.1 the registration energies for different choices of the stiffness parameter κ are shown, together with a plot of the mean squared gradient magnitude in the resulting template image, which is supposed to give an idea of the sharpness of the edges in the image. Figure 6.2 depicts slices of the generated template along with a sample deformation field for one of the registrations performed during its generation. The sample images show the progression from very local and non-smooth deformation fields that result in very good matches and therefore a very sharp average image to very smooth deformation fields and a blurry average image. The main intent behind the averaging is to get average values for structures that can vary in the Hounsfield unit they are represented in, like bones and to get rid of anatomical details that are not present in a majority of the scans. However, if the non-rigid registration is given too much freedom, details present in the dataset used as fixed image can be generated from unrelated structures in the other images. As is usually the case in non-rigid registration a compromise has to be found between the aim of a good match and a match that identifies unrelated structures with each other. For the very low stiffness value $\kappa = 15$ the deformation field is exceedingly local and in some slices even exhibits edges. The resulting template image accordingly looks almost the same as the reference used during the registrations. As a good compromise for this work we identified a stiffness of $\kappa = 25$ by manual inspection of the data.



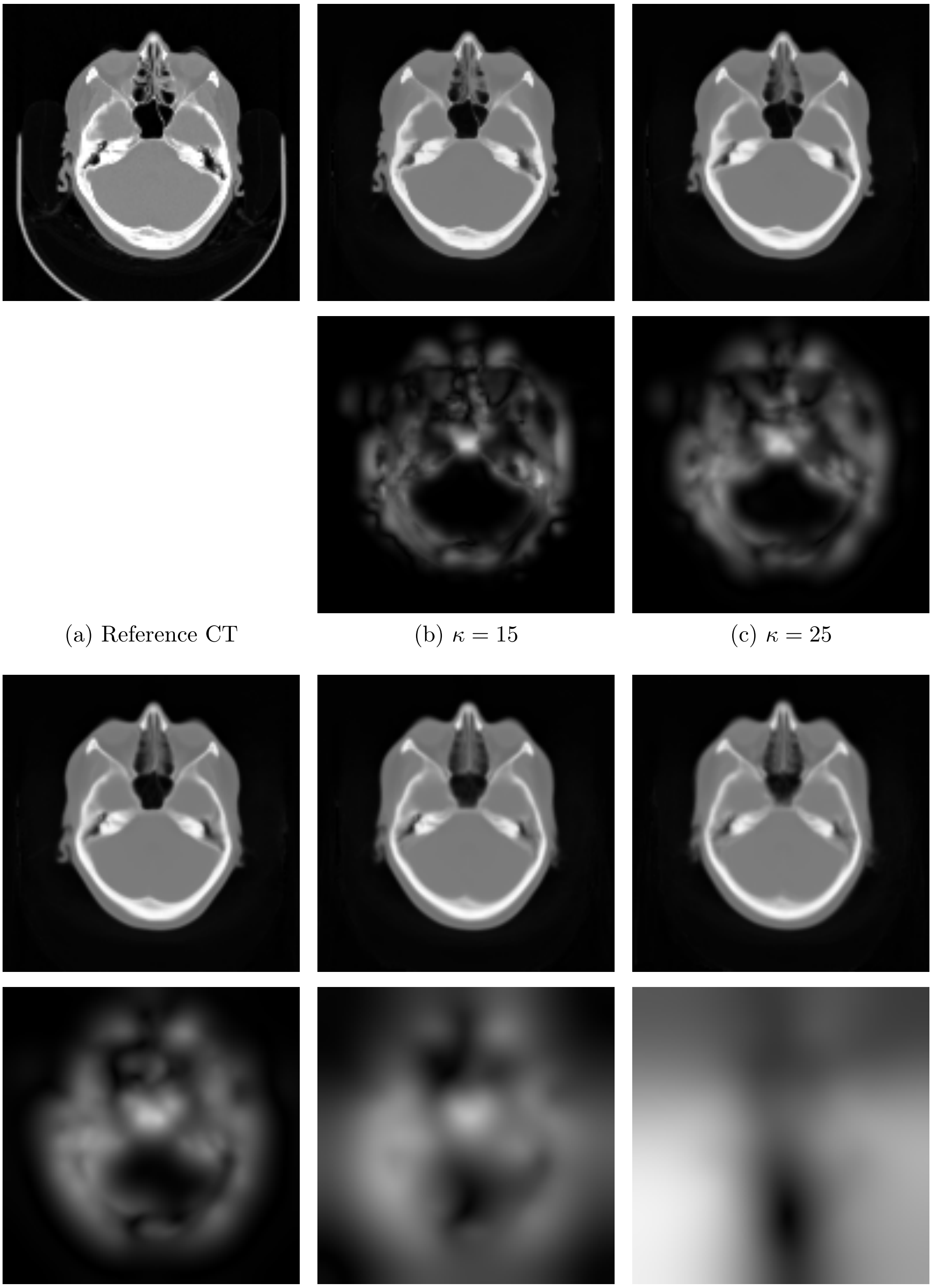
(a) Sum of squares graph

(b) Average distances $\mathcal{D}$

(c) Average regularizers $\alpha_\kappa \mathcal{R}$

(d) Average energies $\mathcal{E}$

Figure 6.1: Average registration energies for different choices of the stiffness κ in the generation of the template image. (a) shows the average, squared, gradient magnitude in the atlas image as in indicator of image contrast and (b) the corresponding average value of the distance measure, (c) regularizer and (d) overall registration energy.



(a) Reference CT

(b) $\kappa = 15$

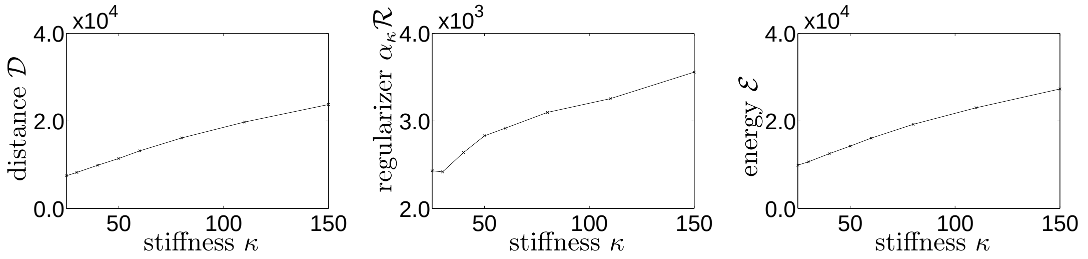
(c) $\kappa = 25$

(d) $\kappa = 40$

(e) $\kappa = 80$

(f) $\kappa = 150$

Figure 6.2: Different results for the averaged image used as the template for different choices of the stiffness κ. (a) shows the CT image used as fixed image F for the registrations. The remaining columns show the resulting template image (top row) and a sample deformation field of one of the registration performed during the template image generation. The higher the stiffness the smoother the deformations and the blurrier and less detailed the template image.



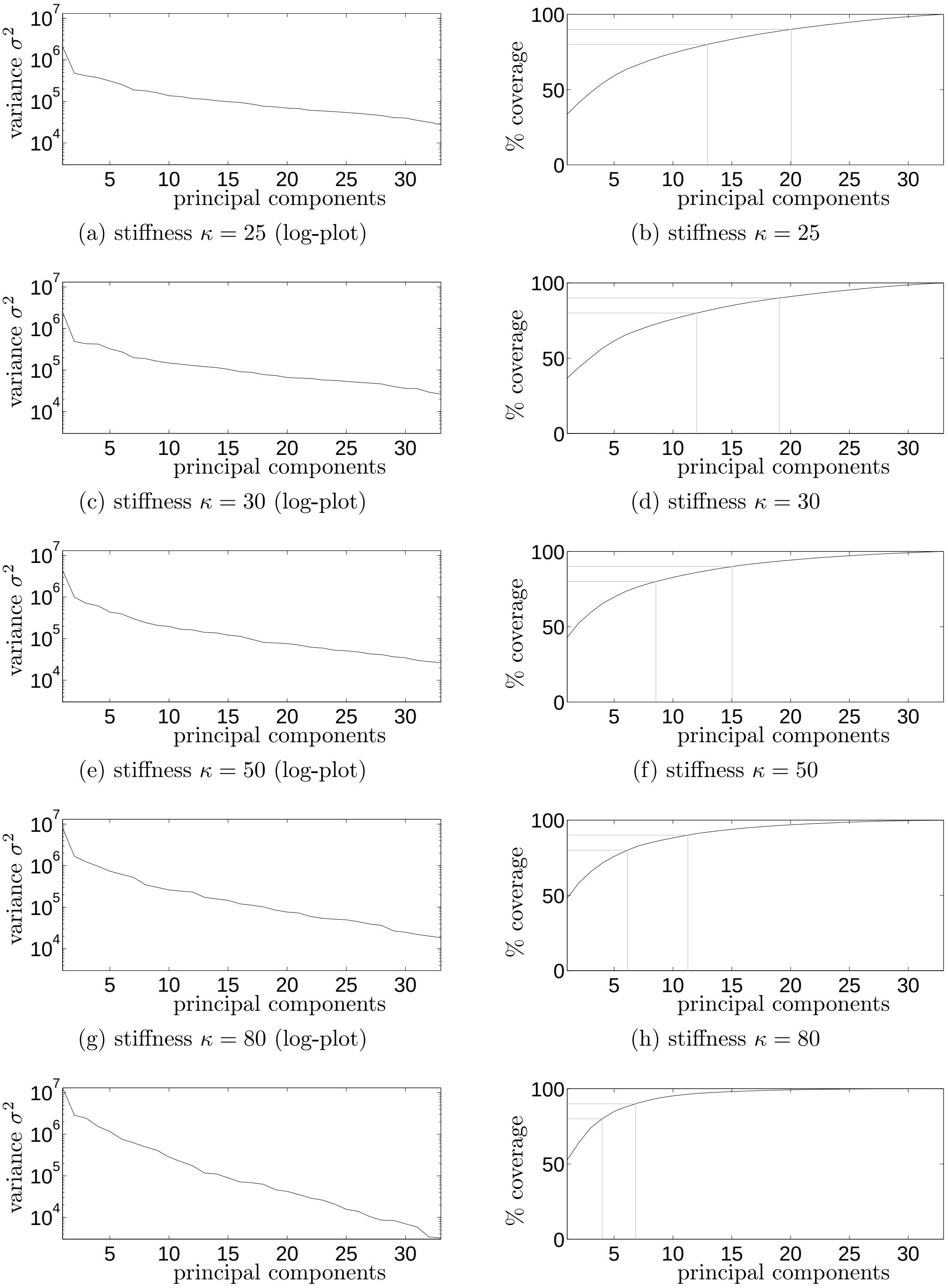
(a) Average distance $\mathcal{D}$ (b) Average regularizer $\alpha_\kappa \mathcal{R}$ (c) Average energie $\mathcal{E}$

Figure 6.3: The plots show the average registration energies over all training data used for the gold standard generation, with respect to the chosen stiffness parameter κ. This illustrates nicely that, the lower κ is chosen, the better the distance measure can be minimized.

6.1.3 PCA Generation

The next step is the generation of the PCA model. As we later want to register the template image onto the patient data, the training data also has to be generated by registering the template onto the patient CT images i. e. the template is used as moving image M in these mono-modal registrations. Again the unknown in this process is the stiffness parameter κ that should be used during the non-rigid registrations. If the stiffness is chosen low the deformation field will be very fine grained and local and a PCA decomposition will not find too many meaningful major axes. If it is chosen high the final match will not be good and the determined major axes will describe deformation fields yielding bad matches. To get an intuition about the effects of the stiffness κ several experiments are run with a stiffness parameter of $\kappa \in \{25, 30, 4050, 60, 80, 110, 150\}$. These can be analyzed with respect to the quality of the numerical match i. e. the registration energies (see Figure 6.3) and the variances along the axes of the PCA model (see Figure 6.4) and the curvature PCA model (see Figure 6.5).

The average resulting registration energies after the non-rigid registration (see Figure 6.3) show a steady decrease with decreasing κ, i. e. the lower the stiffness, the better the match, which does not indicate a specific value for κ as a good value. The second criterion we have to keep in mind is how well the results of these gold standard registrations can be captured by the PCA models. This is characterized by the number of Eigenmodes necessary to capture a certain percentage of the variation in the data. This naturally improves the higher the stiffness during the training registrations was set as this will lead to generally smoother results, which are more likely to coincide between different datasets. This behavior is illustrated in Figure 6.4 and 6.5. The plots show the percentage of the overall variation in the training data covered by a number of principal components. It is interesting to see that in general the PCA model indicates a larger coverage of the variance in the learning data, than the curvature PCA model. It would be rash, however, to judge from this that the PCA model is superior. For example the curvature training data does not contain any rotational information as this is lost in calculating the derivatives and similar things will be true for relatively stiff local motions. This kind of information is therefore not



(a) stiffness $\kappa = 25$ (log-plot)

(b) stiffness $\kappa = 25$

(c) stiffness $\kappa = 30$ (log-plot)

(d) stiffness $\kappa = 30$

(e) stiffness $\kappa = 50$ (log-plot)

(f) stiffness $\kappa = 50$

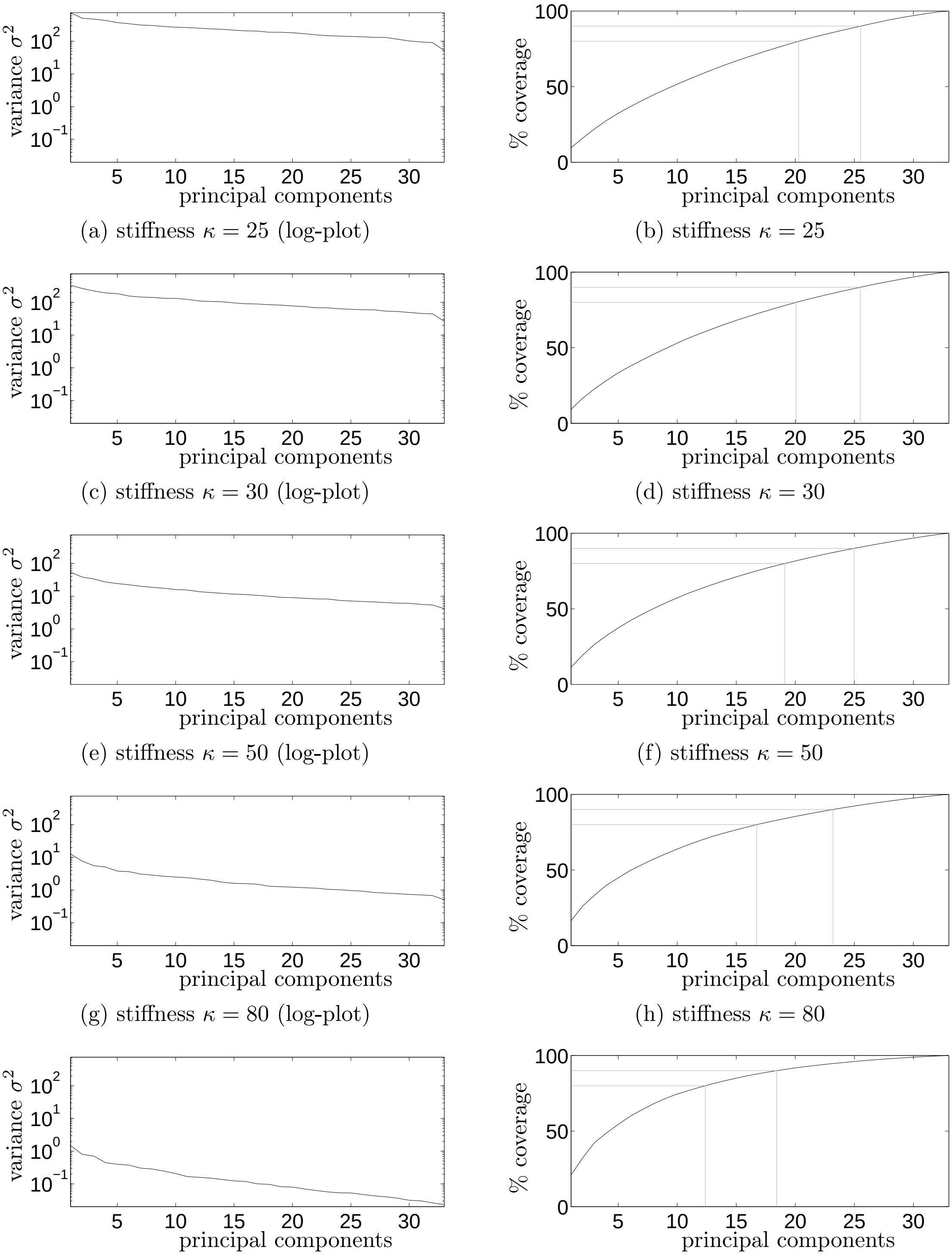
(g) stiffness $\kappa = 80$ (log-plot)

(h) stiffness $\kappa = 80$

(i) stiffness $\kappa = 150$ (log-plot)

(j) stiffness $\kappa = 150$

Figure 6.4: Plots showing the model variation for different stiffness values used during the gold standard generation for the PCA model. Left: Log-plots of the variance σ^2 along the major axes of the PCA model. Right: Ratio of the cumulative variance to the overall variance contained in the model i. e. $\frac{\sum_{i=1}^{k} \sigma_i^2}{\sigma^2}$. An 80% and a 90% coverage respectively are indicated by the light gray lines.



(a) stiffness $\kappa = 25$ (log-plot)

(b) stiffness $\kappa = 25$

(c) stiffness $\kappa = 30$ (log-plot)

(d) stiffness $\kappa = 30$

(e) stiffness $\kappa = 50$ (log-plot)

(f) stiffness $\kappa = 50$

(g) stiffness $\kappa = 80$ (log-plot)

(h) stiffness $\kappa = 80$

(i) stiffness $\kappa = 150$ (log-plot)

(j) stiffness $\kappa = 150$

Figure 6.5: Plots showing the model variation for different stiffness values used during the gold standard generation for the curvature PCA model. Left: Log-plots of the variance σ^2 along the major axes of the PCA model. Right: Ratio of the cumulative variance to the overall variance contained in the model i. e. $\frac{\sum_{i=1}^{k} \sigma_i^2}{\sigma^2}$. An 80% and a 90% coverage respectively are indicated by the light gray lines.

or only to a small extent present in the learning data of the curvature PCA, while it might contribute a significant variance to the plain PCA model. As this kind of information is easy to represent in the PCA model, it will bias the indicated coverage.

Another noteworthy observation concerning the curvature PCA is that not only the coverage increases, but also the model variation itself decreases steadily with an increasing value for the stiffness κ. This behavior is a consequence of the standard regularizer used in the registrations penalizing exactly the thing that we are trying to learn in the curvature PCA model.

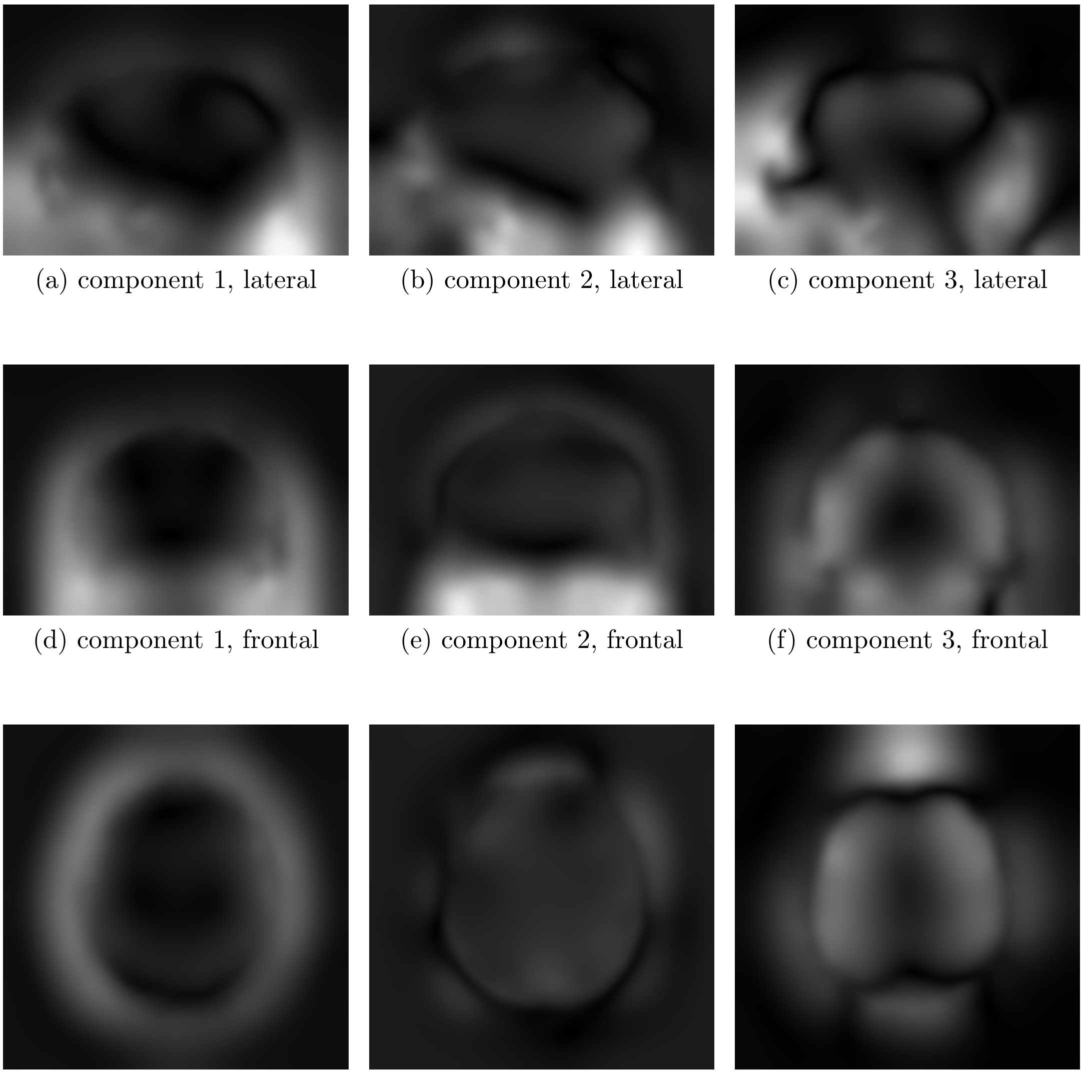
For the final evaluation a stiffness of $\kappa = 50$ was used as a compromise between match quality and model coverage, which gives us a coverage of 90% with 15 components for the plain PCA model, and 80% with 19 components in the curvature PCA model. To give an impression of the resulting model the first three components of the standard PCA model are depicted in Figure 6.6. These first components show very smooth and global deformations mostly related to changes in size (mostly component 1) and shape of the skull.

As a final step it is worth to consider the integration of the PCA model mean, into the template image. This could either be done by directly applying the mean deformation $\bar{\boldsymbol{w}}$ to the template image or by combining it with the current estimated transform $\boldsymbol{u}$ during the registration. The second approach generates more overhead during the registration, but has the advantage that we circumvent one additional resampling step on the template. If the mean is treated in this way this has two consequences: First, the mean transform does not generate a penalty energy in the standard regularizer, as it has been incorporated into the template image and second, the PCA model can be treated as having a zero-mean, thus simplifying the PCA regularization terms.

6.2 Evaluation

The basic registration algorithm used for the evaluation uses the mutual information as distance measure in conjunction with the curvature regularizer. This algorithm is run with the additional PCA regularizer, the curvature PCA regularizer or with no additional regularization. The used optimization algorithm is the Newton formulation (4.3). For the treatment of the Hessian both the choices presented in Section 5.2.2 and Section 5.2.3 are employed: Either the Hessian of the PCA is treated together with the Hessian of the distance measure and estimated numerically via the secant condition according to (4.7) and (4.10) or it is handled individually and therefore exactly, by making use of the Sherman-Morrison-Woodbury formula (5.35) for the PCA regularizer (5.37) or for the curvature PCA regularizer (5.60). There are thus five different algorithms to compare:

- standard – standard registration (no model regularization)
- PCA approximate – registration with PCA regularization and a numerical estimate for the application of the inverse Hessian of the PCA term
- PCA exact – registration with PCA regularization, with the Hessian solved exactly for the PCA



(a) component 1, lateral (b) component 2, lateral (c) component 3, lateral

(d) component 1, frontal (e) component 2, frontal (f) component 3, frontal

(g) component 1, axial (h) component 2, axial (i) component 3, axial

Figure 6.6: Frontal, axial and lateral views of the gradient magnitude of the first three principal components of the standard PCA regularizer model.

- curvature PCA approximate – registration with curvature PCA regularization and a numerical estimate for the application of the inverse Hessian of the PCA term

- curvature PCA exact – registration with curvature PCA regularization, with the Hessian solved exactly for the PCA

The quality of the atlas registrations is evaluated by comparison with the CT scans available for the patients. The CT images are aligned with the MR images used during the atlas registration by a rigid registration. As the skull is a rigid object and the rigid registration we employ has an accuracy of about 0.7mm - 1.2mm target registration error for CT to MR registrations (see [Hahn 10] for a presentation of the used rigid registration algorithm and an evaluation of its accuracy; the used method is KCR), the aligned CT images can be treated as a reasonable ground truth for the evaluation. The scanner table visible in the ground truth CT is masked and ignored. For the comparison of the ground truth CT with the registered template CT image several measures are employed.

- RMSE – root mean square error (in Hounsfield units (HU))

- MAE – mean absolute error (in HU)

- BDICE – Dice's coefficient of thresholded bone areas (thresholded at 600 HU)

- STDICE – Dice's coefficient of thresholded bone and soft tissue areas (thresholded at -200 HU)

Mathematically these are defined as follows: Let R be the ground truth reference image, T the deformed template image and the points in the discretized image domain Ω given by $\mathbf{x}_i \in \Omega$ with $i = 1, ..., n$, then they can be written as

$$\text{RMSE}(R, T) = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (R(\mathbf{x}_i) - T(\mathbf{x}_i))^2} \tag{6.1}$$

$$\text{MAE}(R, T) = \frac{1}{n} \sum_{i=1}^{n} |R(\mathbf{x}_i) - T(\mathbf{x}_i)|. \tag{6.2}$$

For the DICE measures we additionally, introduce $N(I, t) = |\{\mathbf{x}_i \mid I(\mathbf{x}_i) > t\}|$ and $N(I, J, t) = |\{\mathbf{x}_i \mid I(\mathbf{x}_i) > t \wedge J(\mathbf{x}_i) > t\}|$ i. e. N counts the number of pixels above a threshold in one or two images. The two DICE scores we use in our evaluation are therefore

$$\text{BDICE}(R, T) = \frac{2N(R, T, 600)}{N(R, 600) + N(T, 600)} \tag{6.3}$$

$$\text{STDICE}(R, T) = \frac{2N(R, T, -200)}{N(R, -200) + N(T, -200)}. \tag{6.4}$$

None of these measures alone is perfect for an evaluation of this sort, but together they give a good impression of the quality of the registrations. The main problem of RMSE and MAE as a quality measure is their dependence on the size of the background area. The larger the background area, the lower the mean values will be as the registration of air to air will obviously always yield good results. They are, therefore, not well suited for a comparison between different works that employ different datasets, as these may contain more or less large background areas, which would bias the results. However, a comparison of our different methods, which are all evaluated on the same datasets will work fine. RMSE and MAE allow a fine grained comparison on the actual HU, which are after all, what we try to recover by the atlas registration. The Dice measures BDICE and STDICE do not have a problem with a varying background area. In exchange they can only be applied on masks, thus ignoring the actual HU units. This is most prominent in the results for the bone masks BDICE. The low values achieved there by all registrations is, in part, due to the template image we use. Especially in the facial area i. e. around the nasal cavity, sinuses etc. there are a lot of rather thin bone structures. These already have a somewhat lower HU in the reference CT images we use. In the template image this is compounded, as it is a mean of these CT images, and the registrations in the template image generation do not yield identical images. The averaging thus leads to a smoothing of the bone structures in this area and therefore to lower HU values. Many of these structures are therefore not captured by the thresholding used to generate the bone masks used for BDICE. In the original CT images however some of these structures are included in the bone mask and consequently lead to mismatches in the masks that no registration algorithm can compensate.

The evaluation is split for the 25 T1 and 17 T2 weighted MR scans, as the distance measure might give slightly different energies for the different MR sequences (the normalization introduced in Section 3.5.2 is not used here). Otherwise results for the different patient data is averaged and presented with the resulting mean and variance for the individual measures.

A problem in the application of the registration algorithm is the parameter choice for the stiffness κ, the weighting β of the PCA regularizer and the number of PCA components used. Even though one can reason about the amount of morphological variation covered by a particular model and the number of components used (compare Section 6.1.3) this does not directly suggest for which number of components used in a regularization term we will get the best results. It is possible to limit the range of reasonable values, but eventually experimentation or simply a brute force parameter search is necessary to determine a good parameter set. In our case a brute force search on the complete evaluation data is rather prohibitive due to runtime constraints. Especially in the case of the "exact" application of the PCA regularizers the runtime of a single registration is already rather high, as one has to solve several times for the standard regularizer in each iteration step. Therefore, a brute force search of many parameter combinations is performed on a small subset of the evaluation data. The most promising candidates are then applied to the full data. All the evaluations are performed in a leave-one-out manner i. e. the dataset that the atlas registration is applied to is excluded from the model generations process.

The results for the standard approach not using a PCA regularizer are given in table 6.2. As this method only requires the stiffness parameter κ it is possible to give results for the whole range of reasonable values. The best results (highlighted in each column) show that for the T2 weighted MR scans a somewhat lower stiffness of $\kappa = 4$ gave the best results compared with the best parameter setting of $\kappa = 6$ for the T1 weighted scans. Overall we have a mean absolute error of about 76 HU with a rather high standard deviation of 19 HU and a root mean square error of about 189 HU for the T1 weighted MR data as reference. As these values include the background the actual deviation in the foreground pixels is higher. Nonetheless, we get a very good agreement in the STDICE i. e. soft-tissue and bones to air match. The BDICE value is substantially lower which is in part due to the used template image, as discussed above. The results for the T2 weighted reference data is overall a bit worse than the results for the T1 weighted reference MR images in the measures comparing HU directly (RMSE, MAE) and comparable in the Dice measures.

Table 6.2 shows the result for the PCA regularizer applied with the "approximate" optimization. Both the MAE and RMSE decrease with the decrease being much more noticeable for the T2 weighted reference MR images. For these we also see a large decrease in the variance of the measures, indicating a higher robustness of the model constrained registration. The Dice measures also improve. The largest improvement can been seen in the BDICE. As the bones are relatively fine structures by comparison this indicates a better match in the details. The "exact" version shows similar results, although with slightly different parameters.

The results of the curvature PCA regularizer presented in table 6.4 for the "approximate" and in table 6.5 for the "exact" optimization, give a very similar impression.

Whether any of these improvements are significant is difficult to say, as the variance for the results given not only depends on the quality of the registration, but also on the quality of the atlas. As our atlas is not perfect, even a perfect registration would likely not be able to generate a result with no distance in any of our measures. This would incur a "base" variance even for a perfect registration algorithm. The variances for the measures given are therefore not suited to judge the significance of the results. We can however see that we get general improvements in all measures and usually also their variances, for the model constrained approaches. This means the average result improves and the likelihood of a really bad (outlier) result decreases.

Overall the results give the impression that the type of model (plain or curvature) or its "approximate" or "exact" solution only make a minor difference in the real application. The main difference seems to be that plain wrong deformations are inhibited enough to nudge the registration towards the correct match. Usually, a bad result that matches unrelated parts with each other will constitute a local minimum in the landscape of the registration energy. The deformation models help to eliminate these undesirable local minima, while leaving the local minima of the desired solutions intact. Hence, the relatively minor influence of the actual formulation of the constraining model term.

We can therefore conclude that the "approximate" implementation, which requires far less computational overhead, is sufficient to improve the results of our atlas registration. This implementation only requires the computation of a few addition dot-products and vector additions. In exchange we get a higher combined robustness and accuracy of the registration application if we have an application case that has a limited variability in the kind of deformations that can occur.

κ	RMSE		MAE		BDICE		STDICE	
	E	σ	E	σ	E	σ	E	σ
1.5	$1.943\cdot10^{2}$	$3.282\cdot10^{1}$	$8.414\cdot10^{1}$	$1.894\cdot10^{1}$	$6.234\cdot10^{-1}$	$8.962\cdot10^{-2}$	$9.544\cdot10^{-1}$	$1.600\cdot10^{-2}$
2	$1.894\cdot10^{2}$	$3.455\cdot10^{1}$	$8.196\cdot10^{1}$	$2.036\cdot10^{1}$	$6.172\cdot10^{-1}$	$9.882\cdot10^{-2}$	$9.611\cdot10^{-1}$	$1.125\cdot10^{-2}$
3	$1.815\cdot10^{2}$	$3.487\cdot10^{1}$	$7.875\cdot10^{1}$	$2.084\cdot10^{1}$	$6.418\cdot10^{-1}$	$1.069\cdot10^{-1}$	$9.685\cdot10^{-1}$	$6.134\cdot10^{-3}$
4	$1.775\cdot10^{2}$	$3.476\cdot10^{1}$	$7.643\cdot10^{1}$	$2.075\cdot10^{1}$	$6.544\cdot10^{-1}$	$1.166\cdot10^{-1}$	$9.696\cdot10^{-1}$	$5.467\cdot10^{-3}$
6	$1.771\cdot10^{2}$	$3.169\cdot10^{1}$	$7.575\cdot10^{1}$	$1.904\cdot10^{1}$	$6.660\cdot10^{-1}$	$1.060\cdot10^{-1}$	$9.697\cdot10^{-1}$	$4.861\cdot10^{-3}$
10	$1.888\cdot10^{2}$	$3.211\cdot10^{1}$	$8.027\cdot10^{1}$	$1.908\cdot10^{1}$	$6.524\cdot10^{-1}$	$9.572\cdot10^{-2}$	$9.648\cdot10^{-1}$	$5.632\cdot10^{-3}$
15	$2.146\cdot10^{2}$	$3.687\cdot10^{1}$	$9.083\cdot10^{1}$	$2.086\cdot10^{1}$	$5.978\cdot10^{-1}$	$9.329\cdot10^{-2}$	$9.531\cdot10^{-1}$	$1.094\cdot10^{-2}$

(a) MR T1 weighted reference images (mean and variance over 25 datasets)

κ	RMSE		MAE		BDICE		STDICE	
	E	σ	E	σ	E	σ	E	σ
1.5	$2.159\cdot10^{2}$	$7.800\cdot10^{1}$	$8.873\cdot10^{1}$	$3.502\cdot10^{1}$	$6.178\cdot10^{-1}$	$1.722\cdot10^{-1}$	$9.513\cdot10^{-1}$	$2.992\cdot10^{-2}$
2	$2.000\cdot10^{2}$	$7.322\cdot10^{1}$	$8.345\cdot10^{1}$	$3.287\cdot10^{1}$	$6.511\cdot10^{-1}$	$1.504\cdot10^{-1}$	$9.584\cdot10^{-1}$	$2.438\cdot10^{-2}$
3	$1.846\cdot10^{2}$	$6.121\cdot10^{1}$	$7.709\cdot10^{1}$	$2.480\cdot10^{1}$	$6.826\cdot10^{-1}$	$1.123\cdot10^{-1}$	$9.662\cdot10^{-1}$	$1.387\cdot10^{-2}$
4	$1.827\cdot10^{2}$	$5.273\cdot10^{1}$	$7.629\cdot10^{1}$	$2.214\cdot10^{1}$	$6.805\cdot10^{-1}$	$1.029\cdot10^{-1}$	$9.675\cdot10^{-1}$	$1.082\cdot10^{-2}$
6	$1.886\cdot10^{2}$	$5.966\cdot10^{1}$	$7.945\cdot10^{1}$	$2.643\cdot10^{1}$	$6.661\cdot10^{-1}$	$1.078\cdot10^{-1}$	$9.670\cdot10^{-1}$	$1.308\cdot10^{-2}$
10	$2.078\cdot10^{2}$	$8.205\cdot10^{1}$	$8.812\cdot10^{1}$	$3.787\cdot10^{1}$	$6.271\cdot10^{-1}$	$1.377\cdot10^{-1}$	$9.601\cdot10^{-1}$	$2.291\cdot10^{-2}$
15	$2.340\cdot10^{2}$	$8.537\cdot10^{1}$	$9.890\cdot10^{1}$	$4.073\cdot10^{1}$	$5.801\cdot10^{-1}$	$1.410\cdot10^{-1}$	$9.471\cdot10^{-1}$	$2.665\cdot10^{-2}$

(b) MR T2 weighted reference images (mean and variance over 17 datasets)

Table 6.1: Results of the atlas registration using the standard registration approach (no PCA regularization).

κ	β	comp	RMSE		MAE		BDICE		STDICE	
			E	σ	E	σ	E	σ	E	σ
2.5	10^{-8}	23	$1.687\cdot10^{2}$	$3.331\cdot10^{1}$	$7.452\cdot10^{1}$	$1.870\cdot10^{1}$	$6.824\cdot10^{-1}$	$8.339\cdot10^{-2}$	$9.681\cdot10^{-1}$	$7.769\cdot10^{-3}$
3.5	10^{-8}	23	$1.714\cdot10^{2}$	$3.702\cdot10^{1}$	$7.602\cdot10^{1}$	$2.082\cdot10^{1}$	$6.757\cdot10^{-1}$	$1.191\cdot10^{-1}$	$9.686\cdot10^{-1}$	$6.869\cdot10^{-3}$
5.0	10^{-10}	10	$1.702\cdot10^{2}$	$3.693\cdot10^{1}$	$7.530\cdot10^{1}$	$2.087\cdot10^{1}$	$6.789\cdot10^{-1}$	$1.305\cdot10^{-1}$	$9.701\cdot10^{-1}$	$6.111\cdot10^{-3}$
5.0	10^{-9}	26	$1.691\cdot10^{2}$	$3.565\cdot10^{1}$	$7.471\cdot10^{1}$	$2.012\cdot10^{1}$	$6.844\cdot10^{-1}$	$1.196\cdot10^{-1}$	$9.703\cdot10^{-1}$	$5.920\cdot10^{-3}$
6.0	10^{-10}	10	$1.687\cdot10^{2}$	$3.429\cdot10^{1}$	$7.422\cdot10^{1}$	$1.928\cdot10^{1}$	$6.873\cdot10^{-1}$	$1.126\cdot10^{-1}$	$9.709\cdot10^{-1}$	$5.575\cdot10^{-3}$
6.0	10^{-10}	20	$1.685\cdot10^{2}$	$3.364\cdot10^{1}$	$7.409\cdot10^{1}$	$1.902\cdot10^{1}$	$6.879\cdot10^{-1}$	$1.096\cdot10^{-1}$	$9.710\cdot10^{-1}$	$5.481\cdot10^{-3}$
6.0	10^{-11}	10	$1.690\cdot10^{2}$	$3.423\cdot10^{1}$	$7.436\cdot10^{1}$	$1.933\cdot10^{1}$	$6.857\cdot10^{-1}$	$1.164\cdot10^{-1}$	$9.708\cdot10^{-1}$	$5.511\cdot10^{-3}$
6.0	10^{-11}	20	$1.693\cdot10^{2}$	$3.461\cdot10^{1}$	$7.449\cdot10^{1}$	$1.950\cdot10^{1}$	$6.850\cdot10^{-1}$	$1.170\cdot10^{-1}$	$9.707\cdot10^{-1}$	$5.557\cdot10^{-3}$
6.0	10^{-11}	23	$1.692\cdot10^{2}$	$3.465\cdot10^{1}$	$7.445\cdot10^{1}$	$1.952\cdot10^{1}$	$6.852\cdot10^{-1}$	$1.175\cdot10^{-1}$	$9.708\cdot10^{-1}$	$5.478\cdot10^{-3}$

(a) MR T1 weighted reference images (mean and variance over 25 datasets)

κ	β	comp	RMSE		MAE		BDICE		STDICE	
			E	σ	E	σ	E	σ	E	σ
2.5	10^{-8}	23	$1.810\cdot10^{2}$	$6.020\cdot10^{1}$	$7.757\cdot10^{1}$	$2.476\cdot10^{1}$	$6.845\cdot10^{-1}$	$1.105\cdot10^{-1}$	$9.669\cdot10^{-1}$	$1.425\cdot10^{-2}$
3.5	10^{-8}	23	$1.768\cdot10^{2}$	$4.365\cdot10^{1}$	$7.583\cdot10^{1}$	$1.825\cdot10^{1}$	$6.902\cdot10^{-1}$	$9.300\cdot10^{-2}$	$9.695\cdot10^{-1}$	$9.909\cdot10^{-3}$
5.0	10^{-10}	10	$1.693\cdot10^{2}$	$2.619\cdot10^{1}$	$7.341\cdot10^{1}$	$1.302\cdot10^{1}$	$7.086\cdot10^{-1}$	$8.142\cdot10^{-2}$	$9.715\cdot10^{-1}$	$6.679\cdot10^{-3}$
5.0	10^{-9}	26	$1.723\cdot10^{2}$	$3.175\cdot10^{1}$	$7.439\cdot10^{1}$	$1.452\cdot10^{1}$	$7.015\cdot10^{-1}$	$8.383\cdot10^{-2}$	$9.710\cdot10^{-1}$	$7.463\cdot10^{-3}$
6.0	10^{-10}	10	$1.676\cdot10^{2}$	$2.100\cdot10^{1}$	$7.284\cdot10^{1}$	$1.197\cdot10^{1}$	$7.105\cdot10^{-1}$	$7.878\cdot10^{-2}$	$9.723\cdot10^{-1}$	$7.387\cdot10^{-3}$
6.0	10^{-10}	20	$1.679\cdot10^{2}$	$2.136\cdot10^{1}$	$7.295\cdot10^{1}$	$1.209\cdot10^{1}$	$7.098\cdot10^{-1}$	$7.801\cdot10^{-2}$	$9.722\cdot10^{-1}$	$7.613\cdot10^{-3}$
6.0	10^{-11}	10	$1.679\cdot10^{2}$	$2.118\cdot10^{1}$	$7.298\cdot10^{1}$	$1.208\cdot10^{1}$	$7.096\cdot10^{-1}$	$7.833\cdot10^{-2}$	$9.721\cdot10^{-1}$	$7.434\cdot10^{-3}$
6.0	10^{-11}	20	$1.679\cdot10^{2}$	$2.129\cdot10^{1}$	$7.296\cdot10^{1}$	$1.209\cdot10^{1}$	$7.099\cdot10^{-1}$	$7.836\cdot10^{-2}$	$9.721\cdot10^{-1}$	$7.627\cdot10^{-3}$
6.0	10^{-11}	23	$1.678\cdot10^{2}$	$2.100\cdot10^{1}$	$7.291\cdot10^{1}$	$1.200\cdot10^{1}$	$7.100\cdot10^{-1}$	$7.823\cdot10^{-2}$	$9.722\cdot10^{-1}$	$7.556\cdot10^{-3}$

(b) MR T2 weighted reference images (mean and variance over 17 datasets)

Table 6.2: Results of the atlas registration using the PCA regularizer and the “approximate” optimization scheme.

κ	β	comp	RMSE		MAE		BDICE		STDICE	
			E	σ	E	σ	E	σ	E	σ
5.0	10^{-10}	10	$1.701{\cdot}10^{2}$	$3.566{\cdot}10^{1}$	$7.528{\cdot}10^{1}$	$2.013{\cdot}10^{1}$	$6.812{\cdot}10^{-1}$	$1.215{\cdot}10^{-1}$	$9.700{\cdot}10^{-1}$	$6.115{\cdot}10^{-3}$
5.0	10^{-10}	16	$1.703{\cdot}10^{2}$	$3.624{\cdot}10^{1}$	$7.539{\cdot}10^{1}$	$2.046{\cdot}10^{1}$	$6.794{\cdot}10^{-1}$	$1.279{\cdot}10^{-1}$	$9.701{\cdot}10^{-1}$	$5.981{\cdot}10^{-3}$
5.0	10^{-12}	16	$1.708{\cdot}10^{2}$	$3.728{\cdot}10^{1}$	$7.567{\cdot}10^{1}$	$2.097{\cdot}10^{1}$	$6.777{\cdot}10^{-1}$	$1.320{\cdot}10^{-1}$	$9.699{\cdot}10^{-1}$	$6.268{\cdot}10^{-3}$
6.0	10^{-10}	10	$1.684{\cdot}10^{2}$	$3.313{\cdot}10^{1}$	$7.418{\cdot}10^{1}$	$1.866{\cdot}10^{1}$	$6.908{\cdot}10^{-1}$	$1.062{\cdot}10^{-1}$	$9.712{\cdot}10^{-1}$	$5.561{\cdot}10^{-3}$
6.0	10^{-10}	16	$1.683{\cdot}10^{2}$	$3.296{\cdot}10^{1}$	$7.413{\cdot}10^{1}$	$1.859{\cdot}10^{1}$	$6.916{\cdot}10^{-1}$	$1.054{\cdot}10^{-1}$	$9.712{\cdot}10^{-1}$	$5.654{\cdot}10^{-3}$
6.0	10^{-10}	20	$1.682{\cdot}10^{2}$	$3.294{\cdot}10^{1}$	$7.405{\cdot}10^{1}$	$1.858{\cdot}10^{1}$	$6.917{\cdot}10^{-1}$	$1.060{\cdot}10^{-1}$	$9.714{\cdot}10^{-1}$	$5.467{\cdot}10^{-3}$
6.0	10^{-12}	16	$1.698{\cdot}10^{2}$	$3.456{\cdot}10^{1}$	$7.485{\cdot}10^{1}$	$1.942{\cdot}10^{1}$	$6.845{\cdot}10^{-1}$	$1.203{\cdot}10^{-1}$	$9.705{\cdot}10^{-1}$	$5.574{\cdot}10^{-3}$
6.0	10^{-12}	20	$1.697{\cdot}10^{2}$	$3.438{\cdot}10^{1}$	$7.476{\cdot}10^{1}$	$1.932{\cdot}10^{1}$	$6.851{\cdot}10^{-1}$	$1.186{\cdot}10^{-1}$	$9.706{\cdot}10^{-1}$	$5.612{\cdot}10^{-3}$

(a) MR T1 weighted reference images (mean and variance over 25 datasets)

κ	β	comp	RMSE		MAE		BDICE		STDICE	
			E	σ	E	σ	E	σ	E	σ
5.0	10^{-10}	10	$1.683{\cdot}10^{2}$	$2.324{\cdot}10^{1}$	$7.328{\cdot}10^{1}$	$1.218{\cdot}10^{1}$	$7.097{\cdot}10^{-1}$	$7.974{\cdot}10^{-2}$	$9.718{\cdot}10^{-1}$	$6.717{\cdot}10^{-3}$
5.0	10^{-10}	16	$1.689{\cdot}10^{2}$	$2.480{\cdot}10^{1}$	$7.348{\cdot}10^{1}$	$1.260{\cdot}10^{1}$	$7.082{\cdot}10^{-1}$	$8.008{\cdot}10^{-2}$	$9.719{\cdot}10^{-1}$	$6.487{\cdot}10^{-3}$
5.0	10^{-12}	16	$1.690{\cdot}10^{2}$	$2.467{\cdot}10^{1}$	$7.351{\cdot}10^{1}$	$1.251{\cdot}10^{1}$	$7.092{\cdot}10^{-1}$	$8.019{\cdot}10^{-2}$	$9.717{\cdot}10^{-1}$	$6.558{\cdot}10^{-3}$
6.0	10^{-10}	10	$1.682{\cdot}10^{2}$	$2.149{\cdot}10^{1}$	$7.322{\cdot}10^{1}$	$1.202{\cdot}10^{1}$	$7.091{\cdot}10^{-1}$	$7.802{\cdot}10^{-2}$	$9.724{\cdot}10^{-1}$	$7.425{\cdot}10^{-3}$
6.0	10^{-10}	16	$1.684{\cdot}10^{2}$	$2.163{\cdot}10^{1}$	$7.334{\cdot}10^{1}$	$1.205{\cdot}10^{1}$	$7.087{\cdot}10^{-1}$	$7.863{\cdot}10^{-2}$	$9.723{\cdot}10^{-1}$	$7.509{\cdot}10^{-3}$
6.0	10^{-10}	20	$1.683{\cdot}10^{2}$	$2.154{\cdot}10^{1}$	$7.327{\cdot}10^{1}$	$1.207{\cdot}10^{1}$	$7.089{\cdot}10^{-1}$	$7.792{\cdot}10^{-2}$	$9.724{\cdot}10^{-1}$	$7.350{\cdot}10^{-3}$
6.0	10^{-12}	16	$1.678{\cdot}10^{2}$	$2.101{\cdot}10^{1}$	$7.307{\cdot}10^{1}$	$1.193{\cdot}10^{1}$	$7.108{\cdot}10^{-1}$	$7.700{\cdot}10^{-2}$	$9.723{\cdot}10^{-1}$	$7.451{\cdot}10^{-3}$
6.0	10^{-12}	20	$1.676{\cdot}10^{2}$	$2.080{\cdot}10^{1}$	$7.298{\cdot}10^{1}$	$1.191{\cdot}10^{1}$	$7.111{\cdot}10^{-1}$	$7.667{\cdot}10^{-2}$	$9.724{\cdot}10^{-1}$	$7.264{\cdot}10^{-3}$

(b) MR T2 weighted reference images (mean and variance over 17 datasets)

Table 6.3: Results of the atlas registration using the PCA regularizer and the "exact" optimization scheme.

κ	β	comp	RMSE		MAE		BDICE		STDICE	
			E	σ	E	σ	E	σ	E	σ
4.0	10^{-07}	26	$1.742{\cdot}10^{2}$	$3.913{\cdot}10^{1}$	$7.763{\cdot}10^{1}$	$2.207{\cdot}10^{1}$	$6.614{\cdot}10^{-1}$	$1.433{\cdot}10^{-1}$	$9.687{\cdot}10^{-1}$	$6.883{\cdot}10^{-3}$
5.0	10^{-09}	20	$1.709{\cdot}10^{2}$	$3.701{\cdot}10^{1}$	$7.570{\cdot}10^{1}$	$2.087{\cdot}10^{1}$	$6.772{\cdot}10^{-1}$	$1.326{\cdot}10^{-1}$	$9.699{\cdot}10^{-1}$	$6.169{\cdot}10^{-3}$
5.0	10^{-09}	26	$1.710{\cdot}10^{2}$	$3.726{\cdot}10^{1}$	$7.576{\cdot}10^{1}$	$2.099{\cdot}10^{1}$	$6.767{\cdot}10^{-1}$	$1.328{\cdot}10^{-1}$	$9.698{\cdot}10^{-1}$	$6.214{\cdot}10^{-3}$
5.0	10^{-11}	26	$1.710{\cdot}10^{2}$	$3.704{\cdot}10^{1}$	$7.574{\cdot}10^{1}$	$2.085{\cdot}10^{1}$	$6.772{\cdot}10^{-1}$	$1.316{\cdot}10^{-1}$	$9.698{\cdot}10^{-1}$	$6.210{\cdot}10^{-3}$
6.0	10^{-09}	20	$1.694{\cdot}10^{2}$	$3.421{\cdot}10^{1}$	$7.468{\cdot}10^{1}$	$1.922{\cdot}10^{1}$	$6.859{\cdot}10^{-1}$	$1.190{\cdot}10^{-1}$	$9.707{\cdot}10^{-1}$	$5.474{\cdot}10^{-3}$
6.0	10^{-09}	26	$1.694{\cdot}10^{2}$	$3.422{\cdot}10^{1}$	$7.468{\cdot}10^{1}$	$1.922{\cdot}10^{1}$	$6.859{\cdot}10^{-1}$	$1.191{\cdot}10^{-1}$	$9.707{\cdot}10^{-1}$	$5.483{\cdot}10^{-3}$
6.0	10^{-11}	20	$1.694{\cdot}10^{2}$	$3.421{\cdot}10^{1}$	$7.468{\cdot}10^{1}$	$1.923{\cdot}10^{1}$	$6.859{\cdot}10^{-1}$	$1.190{\cdot}10^{-1}$	$9.707{\cdot}10^{-1}$	$5.475{\cdot}10^{-3}$
6.0	10^{-11}	26	$1.694{\cdot}10^{2}$	$3.416{\cdot}10^{1}$	$7.464{\cdot}10^{1}$	$1.920{\cdot}10^{1}$	$6.862{\cdot}10^{-1}$	$1.193{\cdot}10^{-1}$	$9.708{\cdot}10^{-1}$	$5.502{\cdot}10^{-3}$
6.0	10^{-13}	26	$1.695{\cdot}10^{2}$	$3.427{\cdot}10^{1}$	$7.472{\cdot}10^{1}$	$1.927{\cdot}10^{1}$	$6.856{\cdot}10^{-1}$	$1.188{\cdot}10^{-1}$	$9.707{\cdot}10^{-1}$	$5.538{\cdot}10^{-3}$

(a) MR T1 weighted reference images (mean and variance over 25 datasets)

κ	β	comp	RMSE		MAE		BDICE		STDICE	
			E	σ	E	σ	E	σ	E	σ
4.0	10^{-07}	26	$1.693{\cdot}10^{2}$	$2.699{\cdot}10^{1}$	$7.343{\cdot}10^{1}$	$1.256{\cdot}10^{1}$	$7.104{\cdot}10^{-1}$	$8.474{\cdot}10^{-2}$	$9.710{\cdot}10^{-1}$	$6.012{\cdot}10^{-3}$
5.0	10^{-09}	20	$1.679{\cdot}10^{2}$	$2.247{\cdot}10^{1}$	$7.316{\cdot}10^{1}$	$1.204{\cdot}10^{1}$	$7.097{\cdot}10^{-1}$	$8.030{\cdot}10^{-2}$	$9.721{\cdot}10^{-1}$	$6.004{\cdot}10^{-3}$
5.0	10^{-09}	26	$1.681{\cdot}10^{2}$	$2.315{\cdot}10^{1}$	$7.317{\cdot}10^{1}$	$1.218{\cdot}10^{1}$	$7.105{\cdot}10^{-1}$	$7.965{\cdot}10^{-2}$	$9.720{\cdot}10^{-1}$	$6.262{\cdot}10^{-3}$
5.0	10^{-11}	26	$1.678{\cdot}10^{2}$	$2.242{\cdot}10^{1}$	$7.308{\cdot}10^{1}$	$1.203{\cdot}10^{1}$	$7.110{\cdot}10^{-1}$	$7.947{\cdot}10^{-2}$	$9.720{\cdot}10^{-1}$	$6.163{\cdot}10^{-3}$
6.0	10^{-09}	20	$1.676{\cdot}10^{2}$	$2.092{\cdot}10^{1}$	$7.296{\cdot}10^{1}$	$1.188{\cdot}10^{1}$	$7.118{\cdot}10^{-1}$	$7.801{\cdot}10^{-2}$	$9.724{\cdot}10^{-1}$	$7.322{\cdot}10^{-3}$
6.0	10^{-09}	26	$1.676{\cdot}10^{2}$	$2.089{\cdot}10^{1}$	$7.297{\cdot}10^{1}$	$1.188{\cdot}10^{1}$	$7.114{\cdot}10^{-1}$	$7.770{\cdot}10^{-2}$	$9.724{\cdot}10^{-1}$	$7.404{\cdot}10^{-3}$
6.0	10^{-11}	20	$1.676{\cdot}10^{2}$	$2.084{\cdot}10^{1}$	$7.299{\cdot}10^{1}$	$1.188{\cdot}10^{1}$	$7.113{\cdot}10^{-1}$	$7.747{\cdot}10^{-2}$	$9.724{\cdot}10^{-1}$	$7.383{\cdot}10^{-3}$
6.0	10^{-11}	26	$1.676{\cdot}10^{2}$	$2.095{\cdot}10^{1}$	$7.299{\cdot}10^{1}$	$1.190{\cdot}10^{1}$	$7.111{\cdot}10^{-1}$	$7.782{\cdot}10^{-2}$	$9.723{\cdot}10^{-1}$	$7.537{\cdot}10^{-3}$
6.0	10^{-13}	26	$1.679{\cdot}10^{2}$	$2.120{\cdot}10^{1}$	$7.308{\cdot}10^{1}$	$1.196{\cdot}10^{1}$	$7.107{\cdot}10^{-1}$	$7.764{\cdot}10^{-2}$	$9.722{\cdot}10^{-1}$	$7.632{\cdot}10^{-3}$

(b) MR T2 weighted reference images (mean and variance over 17 datasets)

Table 6.4: Results of the atlas registration using the curvature PCA regularizer and the “approximate” optimization scheme.

κ	β	comp	RMSE		MAE		BDICE		STDICE	
			E	σ	E	σ	E	σ	E	σ
5.0	10^{-09}	16	$1.712{\cdot}10^{2}$	$3.730{\cdot}10^{1}$	$7.586{\cdot}10^{1}$	$2.104{\cdot}10^{1}$	$6.765{\cdot}10^{-1}$	$1.326{\cdot}10^{-1}$	$9.696{\cdot}10^{-1}$	$6.170{\cdot}10^{-3}$
5.0	10^{-13}	16	$1.709{\cdot}10^{2}$	$3.726{\cdot}10^{1}$	$7.571{\cdot}10^{1}$	$2.099{\cdot}10^{1}$	$6.773{\cdot}10^{-1}$	$1.326{\cdot}10^{-1}$	$9.698{\cdot}10^{-1}$	$6.255{\cdot}10^{-3}$
6.0	10^{-09}	16	$1.701{\cdot}10^{2}$	$3.485{\cdot}10^{1}$	$7.497{\cdot}10^{1}$	$1.960{\cdot}10^{1}$	$6.839{\cdot}10^{-1}$	$1.201{\cdot}10^{-1}$	$9.704{\cdot}10^{-1}$	$5.467{\cdot}10^{-3}$
6.0	10^{-09}	26	$1.701{\cdot}10^{2}$	$3.486{\cdot}10^{1}$	$7.498{\cdot}10^{1}$	$1.960{\cdot}10^{1}$	$6.838{\cdot}10^{-1}$	$1.202{\cdot}10^{-1}$	$9.704{\cdot}10^{-1}$	$5.476{\cdot}10^{-3}$
6.0	10^{-11}	16	$1.696{\cdot}10^{2}$	$3.433{\cdot}10^{1}$	$7.475{\cdot}10^{1}$	$1.930{\cdot}10^{1}$	$6.856{\cdot}10^{-1}$	$1.190{\cdot}10^{-1}$	$9.707{\cdot}10^{-1}$	$5.555{\cdot}10^{-3}$
6.0	10^{-11}	26	$1.690{\cdot}10^{2}$	$3.420{\cdot}10^{1}$	$7.430{\cdot}10^{1}$	$1.928{\cdot}10^{1}$	$6.868{\cdot}10^{-1}$	$1.184{\cdot}10^{-1}$	$9.709{\cdot}10^{-1}$	$5.429{\cdot}10^{-3}$
6.0	10^{-13}	16	$1.698{\cdot}10^{2}$	$3.452{\cdot}10^{1}$	$7.484{\cdot}10^{1}$	$1.938{\cdot}10^{1}$	$6.850{\cdot}10^{-1}$	$1.186{\cdot}10^{-1}$	$9.706{\cdot}10^{-1}$	$5.705{\cdot}10^{-3}$
6.0	10^{-13}	26	$1.697{\cdot}10^{2}$	$3.457{\cdot}10^{1}$	$7.480{\cdot}10^{1}$	$1.941{\cdot}10^{1}$	$6.851{\cdot}10^{-1}$	$1.187{\cdot}10^{-1}$	$9.706{\cdot}10^{-1}$	$5.672{\cdot}10^{-3}$

(a) MR T1 weighted reference images (mean and variance over 25 datasets)

κ	β	comp	RMSE		MAE		BDICE		STDICE	
			E	σ	E	σ	E	σ	E	σ
5.0	10^{-09}	16	$1.692{\cdot}10^{2}$	$2.540{\cdot}10^{1}$	$7.354{\cdot}10^{1}$	$1.268{\cdot}10^{1}$	$7.090{\cdot}10^{-1}$	$8.003{\cdot}10^{-2}$	$9.718{\cdot}10^{-1}$	$6.802{\cdot}10^{-3}$
5.0	10^{-13}	16	$1.672{\cdot}10^{2}$	$2.097{\cdot}10^{1}$	$7.287{\cdot}10^{1}$	$1.165{\cdot}10^{1}$	$7.127{\cdot}10^{-1}$	$7.882{\cdot}10^{-2}$	$9.720{\cdot}10^{-1}$	$6.539{\cdot}10^{-3}$
6.0	10^{-09}	16	$1.685{\cdot}10^{2}$	$2.169{\cdot}10^{1}$	$7.335{\cdot}10^{1}$	$1.207{\cdot}10^{1}$	$7.087{\cdot}10^{-1}$	$7.741{\cdot}10^{-2}$	$9.722{\cdot}10^{-1}$	$7.829{\cdot}10^{-3}$
6.0	10^{-09}	26	$1.684{\cdot}10^{2}$	$2.161{\cdot}10^{1}$	$7.328{\cdot}10^{1}$	$1.205{\cdot}10^{1}$	$7.093{\cdot}10^{-1}$	$7.751{\cdot}10^{-2}$	$9.722{\cdot}10^{-1}$	$7.853{\cdot}10^{-3}$
6.0	10^{-11}	16	$1.678{\cdot}10^{2}$	$2.103{\cdot}10^{1}$	$7.308{\cdot}10^{1}$	$1.194{\cdot}10^{1}$	$7.107{\cdot}10^{-1}$	$7.754{\cdot}10^{-2}$	$9.723{\cdot}10^{-1}$	$7.513{\cdot}10^{-3}$
6.0	10^{-11}	26	$1.671{\cdot}10^{2}$	$2.126{\cdot}10^{1}$	$7.249{\cdot}10^{1}$	$1.222{\cdot}10^{1}$	$7.124{\cdot}10^{-1}$	$7.792{\cdot}10^{-2}$	$9.722{\cdot}10^{-1}$	$7.625{\cdot}10^{-3}$
6.0	10^{-13}	16	$1.677{\cdot}10^{2}$	$2.074{\cdot}10^{1}$	$7.299{\cdot}10^{1}$	$1.185{\cdot}10^{1}$	$7.108{\cdot}10^{-1}$	$7.756{\cdot}10^{-2}$	$9.725{\cdot}10^{-1}$	$7.215{\cdot}10^{-3}$
6.0	10^{-13}	26	$1.681{\cdot}10^{2}$	$2.127{\cdot}10^{1}$	$7.314{\cdot}10^{1}$	$1.197{\cdot}10^{1}$	$7.102{\cdot}10^{-1}$	$7.780{\cdot}10^{-2}$	$9.722{\cdot}10^{-1}$	$7.694{\cdot}10^{-3}$

(b) MR T2 weighted reference images (mean and variance over 17 datasets)

Table 6.5: Results of the atlas registration using the curvature PCA regularizer and the “exact” optimization scheme.

Chapter 7

Summary

Image registration is a standard task in medical image processing. The retrospective combination of images from different modalities, time points or even different patients is used directly for visualization, as well as indirectly as the basis for numerous applications, such as difference imaging, atlas registration, the training of shape models and many more. Accordingly, this field has sparked a lot of research interest and lead to numerous different registration approaches that differ from each other in the type of transform they employ, such as rigid, affine and non-rigid non-rigid, whether they model it parametric or non-parametric and how they calculate it. An overview over the most commonly employed is given in Section 3.1.

This work focuses on a non-rigid, non-parametric registration scheme. Especially in non-rigid registration approaches the general usability and the evaluation of the registration results is a problem. Many registration approaches require the user to specify numerical parameters necessary for the optimization algorithm. Additionally, in non-rigid registration, it is always necessary for the user to specify how non-rigid the desired result may be, as this is purely problem dependent. Finally, if a visually appealing registration result is achieved it is still not completely clear if it only looks good or actually makes medical and physiological sense. The aim of this work is therefore to ease the use of non-rigid registration by providing a solid standard registration approach that makes parameter selection as easy as possible, and integrating additional prior information into the registration that can make the registration more robust and the output more predictable.

The registration scheme used in this work assigns a vector to each individual position in the fixed image that describes its corresponding location in the moving image's frame of reference. If the images are represented as continuous functions this is therefore a vector valued function over the image domain. The registration is now concerned with finding such a transform that leads to a good match and is regular in some sense. The quality of the match is defined by a similarity measure. In this work we employ the sum of squared differences (Section 3.3.1) between the images and the mutual information (Section 3.3.2) which measures the statistical information shared by the images. The required regularity of the transform is usually a requirement pertaining its smoothness. In our case we work most of the time with the so-called curvature regularizer (Section 3.4.2) which penalizes variations in the second order derivative of the deformation function. We also shortly introduce the somewhat simpler diffusion regularizer (Section 3.4.1) which penalizes first order derivatives instead.

These two terms, the distance measure and the regularizer, have to be weighted against each other by the so-called stiffness parameter, to decide whether a better match or a smoother transform is more important. The weighting depends on the scaling and value range of both terms which themselves depend on the input data. By examining the behavior of the registration result with respect to the stiffness parameter and the input data we propose some rescalings that make a choice of this parameter better predictable and more intuitive for the user.

As the type of optimization algorithm has a significant influence on the registration result we compare several methods. The most important aspect in the optimization seems to be how the regularizer is integrated into the formulation. Due to its global influence the regularizer can pose problems in the non-linear optimization, however, the regularization terms considered in this work are only quadratic terms which can be solved directly. The non-linear optimization schemes explored here, therefore, are a semi-implicit gradient descent method, which treats the regularizer implicitly, and several variants of Newton type methods that all work with the correct Hessian matrix for the regularizer. The Newton type methods differ mainly in the way they treat the Hessian of the distance measure. In the case of the sum of squared differences an analytically derived approximation can be used. For a more general applicability that also works with the mutual information distance measure, numeric approximations like an L-BFGS scheme are applied. All methods are evaluated in a single- and multi-level context. The more generally applicable L-BFGS scheme proves to perform almost on par with the analytical approximation for the distance measure. In very seldom cases the numerical estimation of the Hessian causes problems that degrade the results. As an alternative we propose a similar scheme that only approximates the Hessian of the distance measure with a scaled identity matrix. While the single-level performance is not quite as good as the L-BFGS it is very close in the multi-level environment and was a bit more robust in our experiments.

But even for the best non-rigid registration, some problems offer just too many ambiguities to yield a robust and good result. To improve these situations we incorporate additional prior information into the registration formulation. The first kind of additional information we use are known point-to-point correspondences. These landmarks specify a priori known parts of the deformation. Consequently, we propose to remove these regions from the domain over which the registration is computed and instead treat them as boundary regions. This way the computational effort actually gets less, if more known correspondences are specified. The additional information is shown to be able to constrain the registration in a way that improves registration results to agree better with user expectations.

The second approach for integrating prior information presented in this work, is a bit more involved. If gold standard deformations for a certain application are available these can be used to build statistical models of the most common kinds of deformations. In our case we examine two types of model. A PCA model computed directly on the deformation fields and a PCA model computed on the Laplacian of the deformation fields. The first approach is susceptible to rigid transforms contained in the learning data. These are the result of misalignments in the rigid registration that is performed before the gold standard registrations are computed. This is to some extent fixed by adding additional translational components to the PCA model. As the second approach generates the model only on second derivatives of the deformation fields it is robust to rigid motions in the learning data by default. The models are added to the registration formulation as additional energy terms. Any part of a deformation that cannot be represented by the model is quadratically penalized. In the optimization algorithm these terms can be either treated similarly to the distance measure, or we can make use of the linear nature of a PCA transform to solve for them similarly to the regularization term. While the second approach is numerically better it also demands a lot more computational effort and does not yield significant advantages in our experiments.

The model based regularization is applied in the practical application scenario of MR/PET attenuation correction. In PET there exists the necessity of performing attenuation correction, which requires an attenuation map of the patient. In PET/CT scanners the CT can be used as the source of the attenuation map. In a MR/PET hybrid scanner, however, the MR data cannot be used directly to compute an attenuation map, as the values measured by the MR scanner are not related to attenuation in any way. One possible way to derive an attenuation map from the MR image nonetheless is to perform an atlas registration by registering a template CT image to the MR image and use the deformed CT as attenuation map. This multi-modal inter patient CT to MR registration is quite difficult. Allowing large deformations in the non-rigid registration can lead to numerous mis-registrations. A more constrained non-rigid registration, however, will often be too inflexible to match the atlas sufficiently to the patient dataset. We alleviate this by adding the model based regularizers. These allow large deformations that coincide with deformations learned from mono-modal registration data and inhibit deformations deviating from this learned information. Experiments show an increase in overall accuracy and robustness of the atlas registrations.

The algorithm presented in this work is therefore suited to numerous medical applications. For standard registration problems the user can specify the desired smoothness of the result in a simple and intuitive way. If the result does not agree with the users expectations it can be refined by manually adding landmarks that constrain the registration. In specific application scenarios the algorithm can be additionally constrained with information from gold standard training data, to yield robust and reliable results.

Appendix A

Matrix Properties

In this appendix the properties of the linear system matrices resulting from the discretization of the regularizers from Section 3.4.1 and 3.4.2 and, depending on the optimization method, the distance measures are discussed. The main point of interest here is the positive definiteness as it is this property that decides whether most of the standard iterative linear solver schemes like the Krylov subspace methods or Multigrid can be applied.

Let $\mathbf{A} \in \mathbb{R}^{n \times n}$ be the system matrix as indicated above. We want to show that $\mathbf{A}$ is positive definite (PD) under certain, easy to check conditions. A common definition for positive definiteness is

$$\mathbf{x}^T \mathbf{A} \mathbf{x} > 0 \quad \forall \|\mathbf{x}\| \neq 0 \tag{A.1}$$

A necessary and sufficient condition for $\mathbf{A}$ being PD is that there exist n real valued, positive Eigenvalues for $\mathbf{A}$, i. e.

$$\lambda_i \in \mathbb{R} \quad \wedge \quad \lambda_i > 0 \quad \forall\ \mathbf{A}\mathbf{v}_i = \lambda_i \mathbf{v}_i \quad \|\mathbf{v}_i\| = 1 \quad i = 1, \ldots, n. \tag{A.2}$$

This is a sufficient condition for (A.1), as

$$\begin{aligned} \mathbf{x}^T \mathbf{A} \mathbf{x} &= \mathbf{x}^T (\mathbf{v}_1, \ldots, \mathbf{v}_n)^T \operatorname{diag}(\lambda_1, \ldots, \lambda_n)(\mathbf{v}_1, \ldots, \mathbf{v}_n)\mathbf{x} \\ &= \mathbf{y}^T \operatorname{diag}(\lambda_1, \ldots, \lambda_n)\mathbf{y} \\ &= \sum_{i=1}^{n} \lambda_i y_i^2 \\ &> 0 \quad \forall \|\mathbf{y}\| = \|(\mathbf{v}_1, \ldots, \mathbf{v}_n)\mathbf{x}\| = \|\mathbf{x}\| \neq 0. \end{aligned}$$

The converse, i. e. that (A.1) implies (A.2) is also true, but requires are more elaborate proof, and is not necessary for the work presented here. A weaker requirement than PD, is the positive semi-definiteness

$$\mathbf{x}^T \mathbf{A} \mathbf{x} \geq 0. \tag{A.3}$$

A matrix being positive semi-definite (PSD) is thus either PD or singular. Similarly to (A.2) a necessary and sufficient condition for $\mathbf{A}$ being PSD, are real valued Eigenvalues ≥ 0, i. e.

$$\lambda_i \in \mathbb{R} \quad \wedge \quad \lambda_i \geq 0 \quad \forall\, \mathbf{A}\mathbf{v}_i = \lambda_i \mathbf{v}_i \quad i = 1, \ldots, n. \tag{A.4}$$

The proof relating (A.4) to (A.3) is identical to (A) with $>$ replaced by $\geq$.

It also holds that adding a PD matrix $\mathbf{A}$ and a PSD matrix $\mathbf{B}$ will yield a PD matrix in turn, as

$$\mathbf{x}^T(\mathbf{A} + \mathbf{B})\mathbf{x} = \underbrace{\mathbf{x}^T\mathbf{A}\mathbf{x}}_{>0} + \underbrace{\mathbf{x}^T\mathbf{B}\mathbf{x}}_{\geq 0} > 0 \qquad \forall \|\mathbf{x}\| > 0 \tag{A.5}$$

Assuming that $\mathbf{A}$ has n real valued Eigenvalues, it can be shown that $\mathbf{A}$ is PD if it is diagonally dominant, i. e.

$$a_{ii} \geq \sum_{\substack{j=1 \\ j\neq i}}^{n} |a_{ij}| \qquad \forall\, 1 \leq i \leq n \tag{A.6}$$

We now consider an arbitrary Eigenvalue λ and its associated Eigenvector $\mathbf{v}$. Without loss of generality it is possible to choose a scaling of $\mathbf{v}$ and an index i such that

$$v_i \geq |v_j| > 0 \qquad \forall\, 1 \leq j \leq n \tag{A.7}$$

Using this maximum entry i in the Eigenvector and the diagonal dominance (A.6) it is possible to show that

$$\begin{aligned} \lambda v_i &= \sum_{j=1}^{n} a_{ij} v_j \geq a_{ii} v_i - \sum_{\substack{j=1 \\ j\neq i}}^{n} |a_{ij}||v_j| \\ &\geq a_{ii} v_i - \sum_{\substack{j=1 \\ j\neq i}}^{n} |a_{ij}| v_i = \Big(a_{ii} - \sum_{\substack{j=1 \\ j\neq i}}^{n} |a_{ij}| \Big) v_i \qquad \text{by (A.7)} \end{aligned}$$

and therefore

$$\lambda \geq a_{ii} - \sum_{\substack{j=1 \\ j\neq i}}^{n} |a_{ij}| \geq 0 \qquad \text{by (A.6).} \tag{A.8}$$

where a_{ij} is the element in the i-th row and j-th column of the matrix $\mathbf{A}$. As this can be shown for each of the n Eigenvalues, we have shown that (A.4) holds and therefore $\mathbf{A}$ is at least PSD.

Additionally, if it is possible to prove that $\mathbf{A}$ is non-singular i. e. $\lambda_i \neq 0$ for all Eigenvalues, then we can conclude that $\mathbf{A}$ is PD. This is, for example necessary for the random walker system matrix. In this case the non-singularity of the diagonally dominant matrix can be shown by requiring two more properties (the following proof is similar to [Schw 06] pp. 499). The first is the strict diagonal dominance in one line, i. e.

$$\exists i \quad a_{ii} > \sum_{\substack{j=1\\j\neq i}}^{n} |a_{ij}|. \tag{A.9}$$

Additionally, it is necessary to require that $\mathbf{A}$ is not reducible. A reducible matrix is defined as

$$\begin{aligned} \exists N_1 \neq \emptyset, N_2 \neq \emptyset \quad N_1 \cup N_2 = N, \quad N_1 \cap N_2 = \emptyset \\ a_{ij} = 0 \quad \forall\, i \in N_1, j \in N_2, \end{aligned} \tag{A.10}$$

where $N = \{1, \ldots, n\}$ is the set of all row/column indices of $\mathbf{A}$. In other words there exists a permutation matrix $\mathbf{P}$ such that

$$\mathbf{P}^T\mathbf{A}\mathbf{P} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ 0 & \mathbf{A}_{22} \end{bmatrix}. \tag{A.11}$$

The non-singularity of $\mathbf{A}$ is then proven by contradiction. Suppose that there exists an Eigenvalue $\lambda = 0$. It follows that, for a single row of $\mathbf{Av} = \lambda\mathbf{v}$

$$\begin{aligned} \sum_{j=1}^{n} a_{ij}v_j &= \lambda v_i = 0 \\ a_{ii}v_i &= -\sum_{\substack{j=1\\j\neq i}}^{n} a_{ij}v_j \\ |a_{ii}||v_i| &= |\sum_{\substack{j=1\\j\neq i}}^{n} a_{ij}v_j| \;\leq\; \sum_{\substack{j=1\\j\neq i}}^{n} |a_{ij}||v_j|. \end{aligned} \tag{A.12}$$

Furthermore choose $\mathbf{v}$ such that $\|\mathbf{v}\|_\infty = 1$, i. e. $|v_j| \leq 1$ with $1 \leq j \leq n$. For $N_1 := \{i \in N \;:\; |v_i| = 1\} \neq \emptyset$ and $i \in N_1$ results

$$\begin{aligned} \sum_{\substack{j=1\\j\neq i}}^{n} |a_{ij}| &\leq |a_{ii}| && \text{by (A.6)} \\ &= |a_{ii}||v_i| \\ &\leq \sum_{\substack{j=1\\j\neq i}}^{n} |a_{ij}||v_j| && \text{by (A.12)} \end{aligned} \tag{A.13}$$

If $N_1 = N$, then $|v_j| = 1 \quad \forall\, 1 \leq j \leq n$ and equation (A.13) implies that

$$\sum_{\substack{j=1\\j\neq i}}^{n} |a_{ij}| = |a_{ii}| \qquad \forall 1 \leq i \leq n, \tag{A.14}$$

which contradicts (A.9). Therefore $N_1 \neq N$ and we define

$$N_2 := N \setminus N_1 \neq \emptyset. \tag{A.15}$$

From equation (A.13) it can be derived that

$$\sum_{\substack{j=1\\j\neq i}}^{n} |a_{ij}| \leq \sum_{\substack{j=1\\j\neq i}}^{n} |a_{ij}||v_j| \qquad \text{and} \qquad \sum_{\substack{j=1\\j\neq i}}^{n} |a_{ij}|(1 - |v_j|) \leq 0 \tag{A.16}$$

From (A.15) it follows that for all indices $j \in N_2$ and therefore $j \notin N_1$ the inequality $1 - |v_j| > 0$ holds. In order to satisfy (A.16) it is therefore necessary that $|a_{ij}| = 0 \quad \forall\ j \in N_2$. As this contradicts the irreducibility (A) the assumption that $\lambda = 0$ has to be wrong.

These results can now be applied to the regularizer matrices used in this work. The discretized system matrices of the diffusion (see Section 3.4.1) and the curvature (see Section 3.4.2) regularizer are both diagonally dominant and symmetric. They are thus PSD. In the semi-implicit gradient descent optimization scheme (Section 4.2), they are added to a (positive) multiple of an identity matrix. As the combination of a PSD matrix (regularizer) and a PD matrix (identity) yields a PD matrix according to (A.5), the system matrix of the semi-implicit gradient descent scheme is indeed PD. The same is true for the inexact newton approaches (Section 4.3) as each of the approximations of the Hessian of the distance measure $\mathbf{H}$ presented there is PD.

Appendix B

Stencil Notation

The system matrices of the registration methods presented in this work have in common that they are very large and sparse. For this type of matrix it is impractical to put them down as a whole. In fact as most of the matrix is filled with 0 anyway, giving every entry explicitly would be very redundant. Instead this kind of matrix, resulting from discretizing a local operator on a regular grid is usually given in the so-called stencil notation (see e. g. [Brig 00]). Basically the stencil is a direct representation of the discrete local operator, much like a filter mask. For example, let us consider a 2-D problem of dimension $n \times m$ to which an operator described by the stencil

$$\begin{bmatrix} s_1 & s_2 & s_3 \\ s_4 & s_5 & s_6 \\ s_7 & s_8 & s_9 \end{bmatrix} \tag{B.1}$$

is applied. If the variables on the discrete grid on which this operation takes place are sorted first column- and then row-wise, then the rows i of the corresponding matrix $A \in \mathbb{R}^{nm,nm}$ with elements a_{ij} are defined as

$$\begin{aligned} a_{i,i-m-1} &= s_1 & a_{i,i-m} &= s_2 & a_{i,i-m+1} &= s_3 \\ a_{i,i-1} &= s_4 & a_{i,i} &= s_5 & a_{i,i+1} &= s_6 \\ a_{i,i+m-1} &= s_7 & a_{i,i+m} &= s_8 & a_{i,i+m+1} &= s_9 \end{aligned} \tag{B.2}$$

However, this is only applicable for the rows not corresponding to a point in the 2-D domain not lying on a boundary. For example on the lower boundary of the first dimension the assignment of $a_{1,1-m-1} = a_{1,-m}$ would be invalid. At the boundaries the stencil therefore has to be deformed according to the specified boundary conditions. For example at the "left" boundary of the domain and with von Neumann boundary conditions the stencil

$$\begin{bmatrix} 0 & s_2 + s_1 & s_3 \\ 0 & s_5 + s_4 & s_6 \\ 0 & s_8 + s_7 & s_9 \end{bmatrix} \tag{B.3}$$

would result. If Dirichlet boundary conditions were chosen s_1, s_4 and s_7 would simply be set to 0, i. e.

$$\begin{bmatrix} 0 & s_2 & s_3 \\ 0 & s_5 & s_6 \\ 0 & s_8 & s_9 \end{bmatrix} \tag{B.4}$$

These deformed stencils are applied as specified by equation (B.2), except for the zero entries in the stencil.

The stencil from (B.1) with Dirichlet boundary conditions would therefore describe the matrix depicted in Figure B.

$$\begin{pmatrix}
s_5 & s_6 & 0 & & \cdots & & 0 & s_7 & s_8 & s_9 & 0 & \cdots \\
s_4 & s_5 & s_6 & 0 & & \cdots & & 0 & s_7 & s_8 & s_9 & 0 & \cdots \\
0 & s_4 & s_5 & s_6 & 0 & & \cdots & & 0 & s_7 & s_8 & s_9 & 0 & \cdots \\
\\
s_3 & 0 & & \cdots & & 0 & s_4 & s_5 & s_6 & 0 & & \cdots & & 0 & s_7 & s_8 & s_9 & 0 & \cdots \\
s_2 & s_3 & 0 & & \cdots & & 0 & s_4 & s_5 & s_6 & 0 & & \cdots & & 0 & s_7 & s_8 & s_9 & 0 & \cdots \\
s_1 & s_2 & s_3 & 0 & & \cdots & & 0 & s_4 & s_5 & s_6 & 0 & & \cdots & & 0 & s_7 & s_8 & s_9 & 0 & \cdots \\
0 & s_1 & s_2 & s_3 & 0 & & \cdots & & 0 & s_4 & s_5 & s_6 & 0 & & \cdots & & 0 & s_7 & s_8 & s_9 & 0 & \cdots \\
\\
& & & \cdots & & 0 & s_1 & s_2 & s_3 & 0 & & \cdots & & 0 & s_4 & s_5 & s_6 & 0 & & \cdots & & 0 & s_7 & s_8 & s_9 \\
& & & & \cdots & & 0 & s_1 & s_2 & s_3 & 0 & & \cdots & & 0 & s_4 & s_5 & s_6 & 0 & & \cdots & & 0 & s_7 & s_8 \\
& & & & & \cdots & & 0 & s_1 & s_2 & s_3 & 0 & & \cdots & & 0 & s_4 & s_5 & s_6 & 0 & & \cdots & & 0 & s_7 \\
\\
& & & & & & & & & & \cdots & & 0 & s_1 & s_2 & s_3 & 0 & & \cdots & & 0 & s_4 & s_5 & s_6 & 0 \\
& & & & & & & & & & & \cdots & & 0 & s_1 & s_2 & s_3 & 0 & & \cdots & & 0 & s_4 & s_5 & s_6 \\
& & & & & & & & & & & & \cdots & & 0 & s_1 & s_2 & s_3 & 0 & & \cdots & & 0 & s_4 & s_5
\end{pmatrix}$$

Figure B.1: Schematic of a matrix described by a 2-D stencil with Dirichlet boundary conditions (see (B.1)).

Appendix C

Notation

a	scalar
$\mathbf{v}$	vector
$\mathbf{A}$	matrix
$\boldsymbol{v}(\mathbf{x})$	a vector valued function
$\mathbf{x}$	a continuous vector valued variable
$\mathrm{KL}(p_1, p_2)$	the Kullback-Leibler divergence
$\mathrm{diag}(\mathbf{A})$	a matrix consisting of the main diagonal of the matrix $\mathbf{A}$
$f \circ g$	function composition
$\mathbb{R}$	real numbers
∇f	gradient of function f
Δf	the Laplacian of function f
$f \star g$	convolution of function f with g
$\mathsf{d}f(\boldsymbol{u};\boldsymbol{\eta})$	Gâteaux derivative of f with respect to the function $\boldsymbol{u}$ using the testfunction $\boldsymbol{\eta}$
Ω	computational domain for a registration
$\|\Omega\|$	the size of the domain Ω
$\partial\Omega$	the boundary of the domain Ω
$\mathcal{U}$	Hilbert space of functions $\Omega \mapsto \Omega$
$\\|\boldsymbol{u}\\|_{\mathcal{U}}$	norm for $\boldsymbol{u} \in \mathcal{U}$
$\langle \boldsymbol{u}, \boldsymbol{v} \rangle_{\mathcal{U}}$	inner product for $\boldsymbol{u}, \boldsymbol{v} \in \mathcal{U}$
Φ	registration transform
$\mathbf{I}$	identity matrix
F	fixed image
M	moving image
$\boldsymbol{u}$	deformation
$M_{\boldsymbol{u}}$	deformed moving image i. e. $M_{\boldsymbol{u}} = M(\mathbf{x} - \boldsymbol{u}(\mathbf{x}))$
$\mathcal{E}(F, M, \boldsymbol{u})$	registration energy
$\mathcal{D}(F, M_{\boldsymbol{u}})$	distance measure
$\mathcal{D}_{\text{MI}}(F, M_{\boldsymbol{u}})$	mutual information distance measure
$\mathcal{D}_{\text{SSD}}(F, M_{\boldsymbol{u}})$	SSD distance measure

$\mathcal{R}(\boldsymbol{u})$	regularizer energy
$\mathcal{R}_{\text{Diff}}(\boldsymbol{u})$	diffusion regularizer energy
$\mathcal{R}_{\text{Curv}}(\boldsymbol{u})$	curvature regularizer energy
$\mathcal{P}(\boldsymbol{u})$	PCA model regularizer energy
$\mathcal{P}_{\Delta}(\boldsymbol{u})$	curvature PCA model regularizer energy
$\mathbf{H}_{\text{SSD}}$	Hessian of the sum of squared differences distance measure
$\mathbf{H}_{\text{MI}}$	Hessian of the mutual information distance measure
$\mathbf{A}_{\Delta}$	system matrix of the discretized diffusion regularizer
$\mathbf{A}_{\Delta^2}$	system matrix of the discretized curvature regularizer
τ	step size in optimization algorithms
s	number of pixels in the discretized domain Ω
d	dimensionality of the problem
κ	weighting factor for the standard regularizer (see Section 3.5.1)
i_F	gray value in the fixed image F
i_M	gray value in the moving image M
$\mathbf{i}$	combined/overlaid gray value $\mathbf{i} = (i_F, i_M)$
$\mathrm{E}[x]$	expectation of the random variable x
$\mathrm{Var}[x]$	variance of the random variable x
p	probability density function
p_F	probability density function of the fixed image gray value distribution
p_M	probability density function of the moving image gray value distribution
$p_{F,M}$	joint probability density function

Appendix D

Abbreviations

1-D	one-dimensional
2-D	two-dimensional
3-D	three-dimensional
AX	C-arm X-ray angiographic imaging
BFGS	quasi-Newton method due to Broyden, Fletcher, Goldfarb and Shanno
CG	conjugate gradient
CR	correlation ratio
CT	computed tomography
DICOM	digital imaging and communications in medicine
DOF	degrees of freedom
DSA	digital subtraction angiography
FFT	fast fourier transform
GPU	graphics processing unit
HU	Hounsfield unit
KL	Kullback-Leibler
MAE	mean absolute error
MI	mutual information
MR	magnetic resonance
MSE	mean squared error
PCA	principal component analysis
PD	positive definite
PDE	partial differential equation
PDF	probability density function
PET	positron emission tomography
PSD	positive semi-definite
SDM	statistic deformation model
SE	sensitivity
SPECT	single photon emission computed tomography
SSD	sum of squared differences

SVD	singular value decomposition
TPS	thin-plate splines
TRE	target registration error

Bibliography

[Albr 08] T. Albrecht, M. Luthi, and T. Vetter. "A statistical deformation prior for non-rigid image and shape registration". In: *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pp. 1–8, Anchorage, AK, June 2008.

[Beau 95] S. S. Beauchemin and J. L. Barron. "The computation of optical flow". *ACM Computing Surveys*, Vol. 27, pp. 433–466, Sep. 1995.

[Book 89] F. L. Bookstein. "Principal Warps: Thin-Plate Splines and the Decomposition of Deformations". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 11, No. 6, pp. 567–585, jun 1989.

[Brig 00] W. L. Briggs, V. E. Henson, and S. F. McCormick. *A multigrid tutorial.* Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2 Ed., 2000.

[Bro 96] M. Bro-Nielsen and C. Gramkow. "Fast Fluid Registration of Medical Images". In: K. Hohne and R. Kikinis, Eds., *Proceedings of the 4th International Conference on Visualization in Biomedical Computing*, pp. 267–276, Springer, London, UK, 1996.

[Bron 99] I. Bronstein, K. Semendjajew, G. Musil, and H. Mühlig. *Taschenbuch der Mathematik.* Harri Deutsch, Frankfurt am Main, 4 Ed., 1999.

[Clar 06] U. Clarenz, M. Droske, S. Henn, M. Rumpf, and K. Witsch. *Computational methods for nonlinear image registration*, Chap. 1, pp. 81–101. Vol. 10 of *Mathematics in Industry*, Springer, Berlin Heidelberg, 2006.

[Daum 09] V. Daum, D. A. Hahn, J. Hornegger, and T. Kuwert. "PCA Regularized Nonrigid Registration for PET/MRI Attenuation Correction". In: W. M. Wells III, S. Joshi, and K. Pohl, Eds., *Proceedings of the MICCAI Workshop on Probabilistic Models For Medical Image Analysis*, pp. 127–138, London, UK, Sep. 2009.

[Doss 08] O. Dössel. *Bildgebende Verfahren in der Medizin: Von der Technik zur medizinischen Anwendung.* Springer, Berlin, 1 Ed., 2008.

[Dros 04] M. Droske and M. Rumpf. "A Variational Approach to Non-Rigid Morphological Registration". *SIAM Journal on Applied Mathematics*, Vol. 64, No. 2, pp. 668–687, 2004.

[Dros 05] M. Droske. *On Variational Problems and Gradient Flows in Image Processing.* PhD thesis, Universität Duisburg-Essen, Fachbereich Mathematik, 2005.

[Duch 76] J. Duchon. “Interpolation des fonctions de deux variables suivant le principe de la flexion des plaques minces”. *R. A. I. R. O. Analyse Mathematique*, Vol. 10, No. 12, pp. 5–12, 1976.

[Evan 91] A. C. Evans, W. Dai, L. Collins, P. Neelin, and S. Marrett. “Warping of a computerized 3-D atlas to match brain image volumes for quantitative neuroanatomical and functional analysis”. In: M. H. Loew, Ed., *Medical Imaging V: Image Processing*, pp. 236–246, Bellingham, WA, 1991.

[Fisc 03a] B. Fischer and J. Modersitzki. “Combining landmark and intensity driven registrations”. *Proceedings in Applied Mathematics and Mechanics*, Vol. 3, No. 1, pp. 32–35, 2003.

[Fisc 03b] B. Fischer and J. Modersitzki. “Curvature based image registration”. *Journal of Mathematical Imaging and Vision*, Vol. 18, No. 1, pp. 81–85, 2003.

[Fisc 99] B. Fischer and J. Modersitzki. “Fast inversion of matrices arising in image processing”. *Numerical Algorithms*, Vol. 22, No. 1, pp. 1–11, Oct. 1999.

[Gaff 09] S. Gaffling, F. Jäger, V. Daum, M. Tauchi, and E. Lütjen-Drecoll. “Interpolation of Histological Slices by Means of Non-rigid Registration”. In: H.-P. Meinzer, T. M. Deserno, H. Handels, and T. Tolxdorff, Eds., *Bildverarbeitung für die Medizin 2009*, pp. 267–271, Berlin, 2009.

[Gaff 11] S. Gaffling, V. Daum, and J. Hornegger. “Landmark-constrained 3-D Histological Imaging: A Morphology-preserving Approach”. In: P. Eisert, J. Hornegger, and K. Polthier, Eds., *VMV 2011: Vision, Modeling & Visualization*, pp. 309–316, Eurographics Association, Berlin, Germany, 2011.

[Habe 05] E. Haber and J. Modersitzki. “Beyond Mutual Information: A simple and robust alternative”. In: H.-P. Meinzer, H. Handels, A. Horsch, and T. Tolxdorff, Eds., *Bildverarbeitung für die Medizin 2005*, pp. 350–354, Springer, Berlin, 2005.

[Habe 06] E. Haber and J. Modersitzki. “A Multilevel Method for Image Registration”. *SIAM Journal on Scientific Computing*, Vol. 27, No. 5, pp. 1594–1607, 2006.

[Hahn 06] D. A. Hahn, G. Wolz, Y. Sun, J. Hornegger, F. Sauer, T. Kuwert, and X. Xu. “A Practical Salient Region Feature Based 3D Multi-Modality Registration Method for Medical Images”. In: J. M. Reinhardt and J. P. Pluim, Eds., *Proceedings of SPIE on Medical Imaging*, pp. 870–879, San Diego CA, USA, March 2006.

[Hahn 10] D. A. Hahn, V. Daum, and J. Hornegger. “Automatic Parameter Selection for Multi-Modal Image Registration”. *IEEE Transactions on Medical Imaging*, Vol. 29, No. 5, pp. 1140–1155, 2010.

[Hart 02] T. Hartkens, D. L. G. Hill, A. D. Castellano-Smith, D. J. Hawkes, C. R. Maurer, A. Martin, W. Hall, H. Liu, and C.L.Truwit. “Using Points and Surfaces to Improve Voxel-Based Non-rigid Registration”. In: *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2002*, pp. 565 – 572, 2002.

[Hart 04] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambridge, second Ed., 2004.

[Henn 03] S. Henn. "A Levenberg-Marquardt Scheme for nonlinear image registration". *BIT Numerical Mathematics*, Vol. 43, No. 4, pp. 743–759, 2003.

[Herm 02a] G. Hermosillo. *Variational Methods for Multimodal Image Matching.* PhD thesis, University of Nice Sophia Antipolis, France, 2002.

[Herm 02b] G. Hermosillo, C. Chef d'Hôtel, and O. Faugeras. "Variational Methods for Multimodal Image Matching". *International Journal of Computer Vision*, Vol. 50, No. 3, pp. 329–343, 2002.

[Hofm 08] M. Hofmann, F. Steinke, V. Scheel, G. Charpiat, J. Farquhar, P. Aschoff, M. Brady, B. Schölkopf, and B. J. Pichler. "MRI-Based Attenuation Correction for PET/MRI: A Novel Approach Combining Pattern Recognition and Atlas Registration". *Journal of Nuclear Medicine*, Vol. 49, No. 11, pp. 1875–1883, Oct. 2008.

[Hofm 09] M. Hofmann, B. Pichler, B. Schölkopf, and T. Beyer. "Towards quantitative PET/MRI: a review of MR-based attenuation correction techniques". *European Journal of Nuclear Medicine and Molecular Imaging, European Journal of Nuclear Medicine and Molecular Imaging*, Vol. 36, No. Supplement 1, pp. 93–104, March 2009.

[Homk 06] L. Hömke. "A multigrid method for anisotropic PDEs in elastic image registration". *Numerical Linear Algebra with Applications*, Vol. 13, No. 2-3, pp. 215–229, mar 2006.

[Horn 81] B. Horn and B. Schunck. "Determining optical flow". *Artificial Intelligence*, Vol. 17, No. 1-3, pp. 185 – 203, aug 1981.

[Horn 96] J. P. Hornak. "The Basics of MRI, a hypertext book on magnetic resonance imaging.". online, 1996. `www.cis.rit.edu/htbooks/mri`.

[Huan 04] X. Huang, Y. Sun, D. Metaxas, F. Sauer, and C. Xu. "Hybrid Image Registration based on Configural Matching of Scale-Invariant Salient Region Features". In: *IEEE Workshop on Image and Video Registration, IVR'04*, pp. 167 – 167, Washington DC, USA, July 2004.

[John 02] H. Johnson and G. Christensen. "Consistent landmark and intensity-based image registration". *IEEE Transactions on Medical Imaging*, Vol. 21, No. 5, pp. 450 – 461, May 2002.

[Kabu 04] S. Kabus, T. Netsch, B. Fischer, and J. Modersitzki. "B-spline registration of 3D images with Levenberg-Marquardt optimization". In: M. J. Fitzpatrick and M. Sonka, Eds., *Medical Imaging 2004: Image Processing*, pp. 304–313, SPIE, 2004.

[Kalm 03] E. M. Kalmoun and U. Rüde. "A Variational Multigrid for Computing the Optical Flow". Tech. Rep., Department of Computer Science, Friedrich-Alexander University Erlangen-Nürnberg, Erlangen, 2003.

[Ke 04] Y. Ke and R. Sukthankar. "PCA-SIFT: a more distinctive representation for local image descriptors". In: *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, pp. 506–513, Pittsburgh, PA, USA, jun 2004.

[Keer 10] V. Keereman, Y. Fierens, T. Broux, Y. D. Deene, M. Lonneux, and S. Vandenberghe. "MRI-Based Attenuation Correction for PET/MRI Using Ultrashort Echo Time Sequences". *Journal of Nuclear Medicine*, Vol. 51, No. 5, pp. 812–818, 2010.

[Kim 08] M.-J. Kim, M.-H. Kim, and D. Shen. "Learning-based deformation estimation for fast non-rigid registration". In: *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW '08. IEEE Computer Society Conference on*, pp. 1–6, IEEE Computer Society, Los Alamitos, CA, USA, June 2008.

[Klei 07] 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, Dec. 2007.

[Knop 06] Z. F. Knops, J. B. A. Maintz, M. A. Viergever, and J. P. W. Pluim. "Normalized mutual information based registration using k-means clustering and shading correction". *Medical Image Analysis*, Vol. 10, No. 3, pp. 432–439, jun 2006.

[Lee 97] S. Lee, G. Wolberg, and S. Y. Shin. "Scattered Data Interpolation with Multilevel B-Splines". *IEEE Transactions on Visualization and Computer Graphics*, Vol. 3, No. 3, pp. 228–244, 1997.

[Maes 97] 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. 187–198, Apr. 1997.

[Matt 79] H. Matthies and G. Strang. "The solution of nonlinear finite element equations". *International Journal for Numerical Methods in Engineering*, Vol. 14, No. 11, pp. 1613–1626, 1979.

[Miko 05] K. Mikolajczyk and C. Schmid. "A performance evaluation of local descriptors". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 27, No. 10, pp. 1615–1630, 2005.

[Mode 04] J. Modersitzki. *Numerical Methods for Image Registration.* Oxford University Press, Oxford, 2004.

[Mura 95] W. Murase and M. Lindenbaum. "Partial eigenvalue decomposition of large images using spatial temporal adaptive method". *IEEE Transactions on Image Processing*, Vol. 4, No. 5, pp. 620–629, 1995.

[Noce 80] J. Nocedal. "Updating Quasi-Newton Matrices with Limited Storage". *Mathematics of Computation*, Vol. 35, No. 151, pp. 773–782, July 1980.

[Noce 99] J. Nocedal and S. J. Wright. *Numerical Optimization. Springer Series in Operations Research*, Springer, 1999.

[Parz 62] E. Parzen. "On the estimation of probability density function and mode". *Annals of Mathematical Statistics*, Vol. 33, No. 3, pp. 1065–1076, 1962.

[Paul 10] J. Paulus, R. Bock, V. Daum, and J. Hornegger. "Non-Rigid Registration to Capture Optic Nerve Head Variability". In: H.-P. Meinzer, T. M. Deserno, and T. Tolxdorff, Eds., *Bildverarbeitung für die Medizin 2010 - Algorithmen, Systeme, Anwendungen*, pp. 221–225, Heidelberg, 2010.

[Pete 08] K. B. Petersen and M. S. Pedersen. "The Matrix Cookbook". Technical University of Denmark, Oct. 2008. Version 20081110.

[Plui 00] J. P. W. Pluim, J. B. A. Maintz, and M. A. Viergever. “Interpolation artefacts in mutual information-based image registration”. *Computer Vision and Image Understanding*, Vol. 77, No. 9, pp. 211–232, Feb. 2000.

[Rams 05] J. Ramsay and B. Silverman. *Functional Data Analysis. Springer Series in Statistics*, Springer, New York, second Ed., 2005.

[Ritt 10] M. Ritt, R. Janka, M. P. Schneider, P. Martirosian, J. Hornegger, W. Bautz, M. Uder, and R. E. Schmieder. “Measurement of kidney perfusion by magnetic resonance imaging: comparison of MRI with arterial spin labeling to para-aminohippuric acid plasma clearance in male subjects with metabolic syndrome”. *Nephrology Dialysis Transplantation*, Vol. 25, No. 4, pp. 1126–1133, 2010.

[Roch 98] A. Roche, G. Malandain, X. Pennec, and N. Ayache. “The Correlation Ratio as a New Similarity Measure for Multimodal Image Registration”. In: *Medical Image Computing and Computer-Assisted Intervention, MICCAI 1998, 1st International Conference, Proceedings*, pp. 1115–1124, Springer Verlag, Cambridge, USA, Oct. 1998.

[Rohl 00] T. Rohlfing. *Multimodale Datenfusion für die bildgesteuerte Neurochirurgie und Strahlentherapie*. PhD thesis, Technical University Berlin, 2000.

[Rohr 01] K. Rohr, H. Stiehl, R. Sprengel, T. Buzug, J. Weese, and M. Kuhn. “Landmark-Based Elastic Registration Using Approximating Thin-Plate Splines”. *IEEE Transactions on Medical Imaging*, Vol. 20, No. 6, pp. 526–534, 2001.

[Ruec 99] D. Rueckert, L. I. Sonoda, C. Hayes, D. L. G. Hill, M. O. Leach, and D. J. Hawkes. “Non-rigid registration using free-form deformations: Application to breast MR images”. *IEEE Transactions on Medical Imaging*, Vol. 18, No. 8, pp. 712–721, 1999.

[Schw 06] H. R. Schwarz and N. Köckler. *Numerische Mathematik*. Teubner, Wiesbaden, 6 Ed., Oct. 2006.

[Spie 09] M. Spiegel, D. A. Hahn, V. Daum, J. Wasza, and J. Hornegger. “Segmentation of kidneys using a new active shape model generation technique based on non-rigid image registration”. *Computerized Medical Imaging and Graphics*, Vol. 33, No. 1, pp. 29–39, 2009.

[Strz 04] R. Strzodka, M. Droske, and M. Rumpf. “Image Registration by a Regularized Gradient Flow: A Streaming Implementation in DX9 Graphics Hardware”. *Computing*, Vol. 73, No. 4, pp. 373–389, 2004.

[Stud 99] C. Studholme, D. L. G. Hill, and D. J. Hawkes. “An overlap invariant entropy measure of 3D medical image alignment”. *Pattern Recognition*, Vol. 32, No. 1, pp. 71–86, 1999.

[Tikh 77] A. N. Tikhonov and V. Y. Arsenin. *Solutions of ill posed problems*. Wiley, New York, 1977.

[Ursc 06] M. Urschler, C. Zach, H. Ditt, and H. Bischof. “Automatic Point Landmark Matching for Regularizing Nonlinear Intensity Registration: Application to Thoracic CT Images”. In: R. Larsen, M. Nielsen, and J. Sporring, Eds., *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2006*, pp. 710 – 717, 2006.

[Wahb 90] G. Wahba. *Spline Models for Observational Data.* Vol. 59 of *CBMS-NSF Regional Conference Series in Applied Mathematics*, SIAM, 1990.

[Wang 00] Y. Wang and L. H. Staib. "Physical Model-Based Non-Rigid Registration Incorporating Statistical Shape Information". *Medical Image Analysis*, Vol. 1, pp. 35–51, 2000.

[Wees 97] J. Weese, G. Penney, P. Desmedt, T. Buzug, D. Hill, and D. Hawkes. "Voxel-based 2-D/3-D registration of fluoroscopy images and CT scans for image-guided surgery". *Information Technology in Biomedicine, IEEE Transactions on*, Vol. 1, No. 4, pp. 284–293, Dec. 1997.

[Weic 98] J. Weickert, B. M. ter Haar Romeny, and M. A. Viergever. "Efficient and reliable schemes for nonlinear diffusion filtering". *IEEE Transactions on Image Processing*, Vol. 7, pp. 398–410, 1998.

[Well 96] W. M. Wells III, P. Viola, H. Atsumi, S. Nakajima, and R. Kikinis. "Multimodal volume registration by maximization of mutual information". *Medical Image Analysis*, Vol. 1, No. 1, pp. 35–51, March 1996.

[West 97] J. West, J. M. Fitzpatrick, M. Y. Wang, B. M. Dawant, C. R. Maurer, Jr., R. M. Kessler, R. J. Maciunas, C. Barillot, D. Lemoine, A. Collignon, F. Maes, P. Suetens, D. Vandermeulen, P. A. van den Elsen, S. Napel, T. S. Sumanaweera, B. Harkness, P. F. Hemler, D. L. G. Hill, D. J. Hawkes, C. Studholme, J. B. A. Maintz, M. A. Viergever, G. Malandain, X. Pennec, M. E. Noz, G. Q. Maguire, Jr., M. Pollack, C. A. Pelizzari, R. A. Robb, D. Hanson, and R. P. Woods. "Comparison and Evaluation of Retrospective Intermodality Brain Image Registration Techniques". *Journal of Computer Assisted Tomography*, Vol. 21, No. 4, pp. 554–566, July 1997.

[Wout 06] J. Wouters, E. D'Agostino, F. Maes, D. Vandermeulen, and P. Suetens. "Non-rigid brain image registration using a statistical deformation model". In: J. M. Reinhardt and J. P. W. Pluim, Eds., *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, pp. 338–345, March 2006.

[Xue 09] Z. Xue and D. Shen. "A new statistically-constrained deformable registration framework for MR brain images". *International Journal of Medical Engineering and Informatics*, Vol. 1, No. 3, pp. 357 – 367, 2009.