UNIVERSITÄT ZU LÜBECK

From the Institute of Mathematics and Image Computing
of the University of Lübeck
Director: Prof. Dr. Jan Modersitzki

Analysis of 3D/2D Image Registration and a New Registration Approach Based on Residual Neural Networks

Dissertation
for Fulfillment of
Requirements
for the Doctoral Degree
of the University of Lübeck

from the Department of Computer Sciences and
Technical Engineering

Submitted by

Pia Franziska Schulz
from Rendsburg

Lübeck, 2025

First referee:
Second referee:
Date of oral examination:
Approved for printing:

Abstract

Image registration is an essential component of image processing with a wide range of applications, particularly in the medical field. The goal of image registration is to determine optimal deformations that align two or more images. Our contributions to this field are twofold.

First, we analyze 3D/2D registration problems. A particular challenge with such registration problems is that a 3D deformation is sought based on 2D data. Since typically regularization is utilized to ensure the existence of reasonable solutions, the question arises of which regularization is appropriate for 3D/2D registration.
We address this question and prove the suitability of a class of second-order regularizers. We additionally prove that first-order regularizers are generally not appropriate. Furthermore, we show that our working assumptions apply for common settings of 3D/2D registration and we also extend our results to registration problems of other dimensions. Our analysis contributes to a more comprehensive understanding of image registration problems.

Second, we present the new residual neural network-based registration approach RNR. This method enables the registration of multiple consecutive images. Moreover, the method ensures diffeomorphic deformations under certain conditions. This is a desirable property in many applications, as diffeomorphisms are invertible and sustain image features.
We show that RNR is theoretically sound and provide a comprehensive validation. In particular, we demonstrate that the method allows for larger deformations than comparative approaches and has competitive speed. Finally, we apply the method to build a breathing model, which in turn forms the basis for respiratory surface electromyography modeling. The latter is a highly relevant field that is used, for example, to improve mechanical ventilation for patients.

Zusammenfassung

Bildregistrierung ist eine wesentliche Komponente der Bildverarbeitung mit einem breiten Anwendungsspektrum, insbesondere im medizinischen Bereich. Das Ziel der Bildregistrierung besteht darin optimale Deformationen zu berechnen, welche zwei oder mehr Bilder in Übereinstimmung bringen. Die Beiträge dieser Arbeit umfassen zwei Themen.

Zunächst analysieren wir 3D/2D Registrierungsprobleme. Die Schwierigkeit dieser Aufgabe besteht darin, dass eine 3D Deformation gesucht wird, ein Teil der Daten jedoch nur 2D Informationen enthält. Da typischerweise Regularisierer verwendet werden, um die Existenz von sinnvollen Deformationen sicherzustellen, stellt sich die Frage, welche Regularisierung für die 3D/2D Registrierung geeignet ist.
Wir befassen uns mit dieser Frage und beweisen, dass eine Klasse von Regularisierern zweiter Ordnung geeignet ist. Wir zeigen zudem, dass Regularisierer erster Ordnung dies im Allgemeinen nicht sind. Darüber hinaus weisen wir nach, dass die hierfür getroffenen Annahmen für gängige Methoden der 3D/2D Registrierung gelten. Außerdem erweitern wir unsere Ergebnisse auf Registrierungsprobleme in anderen Dimensionen. Mit unserer Analyse tragen wir zu einem umfassenden Verständnis von Registrierungsproblemen bei.

Im zweiten Teil dieser Arbeit stellen wir die neue Registrierungsmethode RNR vor, welche auf Residual Neural Networks basiert. RNR ermöglicht die Registrierung mehrerer zeitlich aufeinanderfolgender Bilder. Zudem garantiert die Methode unter gewissen Bedingungen diffeomorphe Deformationen. Dies ist in vielen Anwendungen eine wünschenswerte Eigenschaft, da Diffeomorphismen invertierbar sind und bestimmte Bildmerkmale erhalten.
Wir zeigen, dass unsere Methode theoretisch fundiert ist und liefern eine umfangreiche Validierung. Insbesondere weisen wir nach, dass RNR größere Deformationen ermöglicht als Vergleichsmethoden und eine vergleichbare Geschwindigkeit hat. Abschließend wenden wir die Methode an, um ein Atemmodell zu erstellen, welches wiederum die Grundlage für die Modellierung von Oberflächen-Elektromyographie-Signalen der Atmung bildet. Letzteres ist ein hochrelevanter Themenbereich, der beispielsweise zur Verbesserung der künstlichen Beatmung eingesetzt wird.

Acknowledgments

I am very grateful for the great support I received from many people during the creation of this thesis.

First of all, I would like to thank my supervisor Jan Modersitzki for his advice and guidance. I am thankful that he gave me the opportunity to write my doctoral thesis at the Institute of Mathematics and Image Computing in Lübeck. I learned a lot from our conversations, both scientifically and personally.
Furthermore, I would like to thank Prof. Rafecas for chairing and Prof. Rößler for taking the effort of being a referee.

A huge thanks goes to my colleague Saskia, who quickly became a good friend. I am really happy that we could experience the adventure *doctorate* with all its ups and downs together. Having her by my side in the last years was of incredible value to me. I also want to thank Florian for our enjoyable joint work and for always being available to answer all my questions.
Thanks also to all my other present and former colleagues from MIC and MEVIS for giving me a great time in Lübeck.

Finally, I would like to thank my friends and family for their unconditional support. They were a sympathetic ear by my side and provided me with an environment where I could clear my head from daily work.
Special thanks to my parents and my brother, who always believe in me.
And most of all, a loving thank you to Michi for always being there and taking care of me. Without that support, this thesis would have certainly not been possible.

I am also thankful to live in a time when translators such as DeepL make it easier to write in a language other than the native tongue.

Publications

[1] Saskia Neuber, **Pia F. Schulz**, Sven Kuckertz, and Jan Modersitzki, "Segmentation-inspired Image Registration", in *BVM Workshop*, pp. 205–210, Springer, 2024.

[2] **Pia F. Schulz**, Florian Mannel, and Jan Modersitzki, "Appropriate Order of Regularization in 3D/2D Image Registration", in *Proc. SSVM 2025*, pp. 377–390, Springer, 2025.

[3] Ole Gildemeister, Johannes Bostelmann, **Pia F. Schulz**, Andra Oltmann, Phillip Rostalski, Jan Modersitzki, and Jan Lellmann, "Time-Continuous Sliding Motion Image Registration Using Stationary Velocity Fields for Respiratory Motion Interpolation ", in *Proc. SSVM 2025*, pp. 404–416, Springer, 2025.

[4] **Pia F. Schulz**, Andra Oltmann, Johannes Bostelmann, Ole Gildemeister, Franz Wegner, Jan Lellmann, Philipp Rostalski, and Jan Modersitzki, "Image Registration for a Dynamic Breathing Model", in *BVM Workshop*, pp. 5–11, Springer, 2025.

[5] Andra Oltmann, Ole Gildemeister, Johannes Bostelmann, **Pia F. Schulz**, Jan Graßhoff, Franz Wegner, Alex Frydrychowicz, Jan Lellmann, Jan Modersitzki, Tobias Knopp, and Philipp Rostalski, "Simulation of Diaphragmatic Motor Unit Action Potentials Throughout the Respiratory Cycle Using a Dynamic Breathing Model", in *Proc. EMBC 2025*, accepted

Contents

CHAPTER 1

Introduction

Image registration is a fundamental task in image processing; see e.g. [6, 7, 8] and references therein. The aim of image registration is to find geometrical correspondences between two or more images [6]. This task requires a solid theoretical foundation as well as practical approaches [9]. A solid theoretical analysis is necessary to develop reasonable registration methods, which are in turn needed in applications.
In this thesis, we contribute both to the theoretical analysis of image registration problems and to the development of practical registration methods.

In this first chapter, we introduce the topic of image registration as well as the contributions of this thesis. More precisely, in Sec. 1.1, the general image registration problem and different approaches to solve it are discussed. Then in Sec. 1.2, we present the topics addressed in this thesis and our contributions to them. In Sec. 1.3, we describe the structure of this work.

1.1 Image Registration

In the following, we give an introduction to image registration. For a more detailed description, we refer e.g. to [8, 9, 10].

Given two images that were taken, for example, at different times or with different devices, the goal of image registration is to determine a reasonable deformation such that a transformed version of the first image resembles the second one [11]. Of particular interest is the registration of two images [12], but there are also approaches for multiple images; see e.g. [13, 14]. An exemplary application of a registration with 3D torso images acquired using magnetic resonance imaging (MRI) is shown in Fig. 1.1.
Image registration has a wide range of applications, particularly in the medical field [15]. Medical applications range from computer-assisted diagnosis to computer-aided therapy and surgery [6]. For example, image registration can be used for the treatment verification of pre- and post-intervention images. Registration is also useful for the fusion

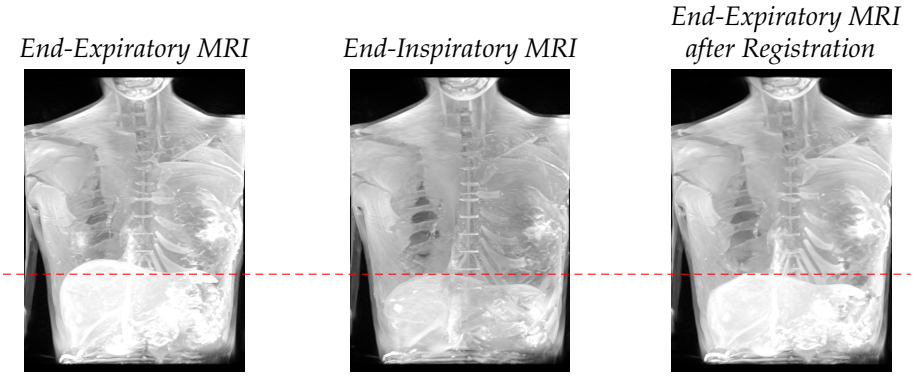


Figure 1.1: *Exemplary registration of 3D MRI torso images: The end-expiratory image is deformed to resemble the end-inspiratory image. The deformation is calculated with the new registration approach presented in Ch. 4. The red dashed line marks the position of the diaphragm as an indication for the image alignment. For details on the registration method and the visualized data see Ch. 4.*

of the complementary information provided by multimodal imaging techniques such as computer tomography (CT) and MRI [9]. In addition to the medical field, image registration also has applications in other areas where imaging techniques are employed, such as biology, chemistry, astronomy and criminology [15].

Therefore, image registration has been intensively researched in the last decades. For an overview of the development of image registration over the years, we refer to the survey articles [7, 8, 16, 17, 18, 19, 20, 21, 22, 23], which have been published consecutively in the listed order from 1998 to the present. Additionally, there are a number of comprehensive introductions to image registration; see e.g. [6, 9, 10, 11, 15, 24].

Furthermore, image registration is closely related to the problem of determining an optical flow; see e.g. [25]. Optical flow approaches model an image as a function in space and time and are typically based on the assumption that image intensities are preserved over time [25]. Seminal works were presented by Horn and Schunck [26] and Lucas and Kanade [27] in 1981.

1.1.1 From Optimization-Based to Learning-Based Registration

In general, image registration methods can be roughly divided into *classical optimization-based* and *learning-based approaches*. The survey articles [7, 16, 17, 18, 19, 22] provide an overview of classical optimization-based registration, whereas reviews of learning-based registration are presented in [8, 19, 20, 21, 22, 23], with both article lists arranged in order of publication. The advantages and disadvantages of classical optimization- and learning-based registration methods are discussed widely in the literature; see e.g. [8, 22, 28, 29, 30]. Below we summarize the ideas as well as the main advantages and disadvantages of both categories.

In classical optimization-based approaches the registration is performed by minimizing an energy functional for an image pair [29]. The functional takes the similarity of the images into account and thereby determines a deformation to solve the registration problem. In addition, a regularization term is typically included in the functional to ensure existence of a reasonable solution [15].

Classical optimization-based registration methods are generally well-established and supported by solid mathematical theory; see e.g. [9]. Furthermore, such methods are robust and generalizable, which makes them suitable for a wide range of applications [30]. As the optimization process is specific to each image pair, high registration accuracy is usually achieved [30, 31].
The main disadvantage of classical registration methods is that most of them are slow and computationally expensive [8, 30, 31]. Although parallelized GPU implementations enable a significant acceleration of these methods, most classical methods are only implemented in a slower CPU version [30]. In addition, the registration quality depends on the local minimum achieved, which in turn mainly depends on the selected optimization strategy and initialization [29]. Thus, good registration accuracy is only obtained with appropriate optimization methods.

In the last decade, learning approaches based on neural networks have rapidly advanced and show promising results for the image registration task [8, 31]. Learning-based registration methods take a pair of images as input to a neural network and output a deformation that aligns the images. The network is trained by optimizing a global objective functional on a training dataset, which contains many image pairs. After training, the network can be directly applied to new image pairs without further optimization [8]. Learning-based registration methods can be broadly categorized as supervised or unsupervised [8]. Initially, mainly supervised approaches emerged [8]. Supervised registration methods such as [32, 33] use ground truth transformations during the training process. Since, in contrast to other applications of learning-based approaches, ground truth is not usually available for image registration problems [19], it is determined either manually or using a classical registration method [30]. However, generating ground truth deformations in this way is time consuming and means that supervised methods are not able to outperform classical approaches. Therefore, primarily unsupervised methods were developed in recent years, as these do not require ground truth deformations; see e.g. [34, 35]. Instead, similar to classical approaches, the objective functional in these methods takes the similarity between the images into account [8].
The main advantage of learning-based methods is that registration is very fast once the network has been trained [8, 31]. Usually the registration takes less than a second [28]. Furthermore, it is possible to include additional information such as anatomical landmarks into the training data, which also improves the registration of unseen image pairs for which this information is not available [30]. In addition, the global objective functional is optimized with respect to many image pairs. According to [8], this can be considered as implicit regularization, which compensates for noisy or misleading gradients that can lead to suboptimal local minima. Moreover, learning-based methods commonly utilize the concept of automatic differentiation, see e.g. [36], which allows to easily and exactly determine gradients that are required for optimization and are difficult to calculate manually [29].
Probably the most significant disadvantage of learning-based methods is that they in general require a large amount of training data [30]. Additionally, so far learning-based methods do not have good generalization capabilities [30, 31]. For example, in [30] networks are trained on one brain dataset, but these networks do not perform well on other datasets of the same anatomy. A major challenge with learning-based methods is choosing an appropriate network structure, as the architectural choice significantly affects the registration quality [31]. Also the tuning of hyperparameters, which typically occur in greater numbers than in classical methods, is still a difficulty [30]. Moreover, as with classical optimization-based registration approaches, the registration accuracy is depen-

dent on the selected optimization method. However, in contrast to optimization-based methods, classic termination conditions are generally not used. Instead, the optimization is usually performed for a fixed number of iterations, see e.g. [34, 35], which does not guarantee that a minimum will be reached.

At the present time, both classical optimization- and learning-based approaches are topics of investigation in the image registration community. Although learning-based methods outperform classical methods in several experiments [35, 37], there are many other datasets for which classical methods still perform better in terms of accuracy [29, 30, 38].

A mixture of optimization- and learning-based approaches is given by methods based on so-called *implicit neural representations* (INRs), also known as neural fields [39]. A new INR-based registration method will be presented later in this work.
In contrast to the learning-based approaches described above, the idea behind INR-based registration approaches is not to use neural networks to predict a deformation, but rather to represent a deformation [38]. Therefore, INR-based registration can be considered a specific type of parametric registration; see e.g. [6, 9] for classical parametric registration approaches. As with classical optimization-based methods, INR-based methods perform an optimization for each image pair [8]. More precisely, for each image pair a network is trained, which takes as input a spatial coordinate and outputs a function value [8]. INRs are used either to directly represent a deformation [31, 39, 40] or to represent a velocity field, which is then integrated to obtain a deformation [38, 39].
One type of neural networks used in many INR-based registration methods are *residual neural networks* (ResNets); see e.g. [28, 41]. An introduction to ResNets and the motivation for using them in registration methods is given in Sec. 4.1.
Registration methods based on INRs combine advantages from classical optimization- and learning-based approaches: As with classical approaches, high registration accuracy can be achieved [39] and no large datasets are required for training [42], since a network is trained specifically for each image pair. As with other learning-based approaches, the concept of automatic differentiation enables easy and precise calculation of gradients [38]. Moreover, INR-based methods yield continuous (generally also differentiable) deformations that can be evaluated directly at any position without the need for interpolation [8, 42]. Unlike the learning-based approaches discussed above, the same model can be used regardless of the image resolution [38]. Therefore, also the memory requirements are not limited by the resolution of the images [42].
One disadvantage of INR-based methods compared to the learning-based methods described above is that training a network for each image pair is computationally more intensive and time-consuming than the evaluation of a pre-trained network [42]. In addition, the registration quality of INR-based methods depends significantly on the selected network structure [31].
Despite these drawbacks, INR-based image registration has proven to be useful and has gained popularity in recent years; see e.g. [31, 38, 39, 40].

A frequently used approach is also to combine different forms of registration methods [43]; such as using a learning-based method described above to derive an initial registration, which is then refined using an INR-based method [44].

1.1.2 3D/2D Registration

A particular class of image registration problems is addressed by 3D/2D registration. The goal of 3D/2D registration is to align a 3D image with a 2D image [45]. As the considered images have different dimensionality, this type of registration poses particular

challenges that have not yet been fully resolved [45]. This is our motivation for considering 3D/2D registration in one part of this thesis.

Registration of 3D to 2D data often occurs in image-guided interventions, for instance in cardiology [46], neurology [47] or orthopedics [48]. For example, such problems arise in a clinical setting when pre-operative data is combined with intra-operative data to improve safety and accuracy of the intervention [49]. Usually, 3D scans such as CT or MRI volumes are acquired pre-operatively in order to plan the treatment [49, 50]. During the intervention generally only 2D scans such as cine MRI slices, ultrasound (US) images or projective X-ray scans are available, for example due to time or dose constraints [49, 50]. Obviously, using the 3D data in combination with the intra-operative 2D data allows for easier and better guidance and monitoring during the surgery, which leads to reduced invasiveness and increased accuracy [49]. 3D/2D image registration enables this by aligning the 3D data to the 2D data.

In recent years, 3D/2D image registration has gained widespread attention [45]. For an overview of 3D/2D registration approaches, we refer to the survey articles [45, 49, 51]. All approaches have in common that the 3D and 2D data have to be brought into dimensional correspondence [49]. This can be achieved either by transforming the 3D data into 2D (using projection) or by transforming the 2D data into 3D (using reconstruction or back-projection). While registration methods based on the former strategy maximize the similarity between 2D images, methods based on the latter approach compare 3D images [49]. Transforming 2D data into 3D is a non-trivial task [52], whereas projection of 3D data to 2D requires less model assumptions. In this thesis, we therefore focus on the projective approach, i.e. the determination of a 3D deformation that aligns the projected 3D image with a provided 2D image.

A particular case of 3D/2D registration is *volume-to-slice registration*; see e.g. [53] for a review of volume-to-slice registration methods. In this case the 2D image is given by a slice of a 3D volume instead of, for example, a projective image as in X-ray imaging. Several applications require volume-to-slice registration [53]. For example, in [50], the registration of 3D CT volumes and 2D US slices is considered in the context of liver surgeries.

1.2 Contributions of this Thesis

In this thesis, we contribute to the theoretical foundation of image registration and to the development of practical registration methods:

1. We investigate which type of regularization is appropriate for 3D/2D registration. Our theoretical analysis contributes to a more comprehensive understanding of 3D/2D registration.

2. We develop a new registration approach based on residual neural networks, called RNR. This method enables the computation of multiple deformations describing the motion of an object at successive time points.

Parts of the contents have already been published by the author; see [2] and [4].
Next, we describe the motivation and our contributions to each of the topics in detail. Works on these topics precede ours are discussed within the respective chapters.

1.2.1 Appropriate Regularization in 3D/2D Registration

As one part of this thesis, we analyze the impact of the regularization on the registration of 3D to 2D data.

A regularization term is usually included in the energy functional that is optimized to solve the image registration problem. This is because regularization ensures the existence of reasonable solutions [15]. While regularization is well understood for the registration of 2D to 2D data and 3D to 3D data, see e.g. [54], little is known about regularization for 3D to 2D registration.
In the following, we assume that solutions of the registration problem exist and focus on their plausibility. As outlined above, 3D/2D registration is characterized by the fact that a 3D deformation is sought, but one of the images is only 2D. In many applications it is reasonable to assume that this 2D data is representative for the 3D domain [50]. Hence, the regularizer should ensure that the information given on the 2D set is transferred to the 3D domain. In order to realize this, the regularization should be chosen such that the resulting 3D deformation is sufficiently smooth. An appropriate class of deformations is formed by continuously differentiable deformations; see e.g. [50, 55].

In this work, we address the question of which type of regularization guarantees continuously differentiable deformations in 3D/2D registration. More precisely, we analyze the impact of the regularization order, i.e. the order of the derivative in the regularization term. The reason for this is that the order of the regularizer significantly affects the deformation. Our main result is that specific second-order regularizers are sufficient to guarantee continuous differentiability of the deformations (Th. 3.13), while first-order regularizers are generally not (Th. 3.15).

In detail, our contributions are the following:

- To highlight the impact of the regularization in 3D/2D registration, we reproduce a numerical experiment of Heldmann and Papenberg [50]. In this experiment, two 3D/2D registrations are performed, one with first-order regularization and one with second-order regularization (Exp. 3.1). The experiment shows that the regularization order has a significant influence on the registration result.
- We formulate easily verifiable assumptions that enable us to make provable statements about the regularization in 3D/2D registration (Ass. 3.1–3.3).
- Based on the aforementioned assumptions, we prove that second-order regularization guarantees continuously differentiable deformations (Th. 3.13). The main tool in the proof is the Sobolev embedding theorem.
- We prove that first-order regularizers generally do not guarantee continuously differentiable deformations. More precisely, we provide a large class of examples with first-order regularization that fails to generate continuously differentiable deformations (Th. 3.15).
- We show that our working assumptions are satisfied for common practical 3D/2D registration problems (Sec. 3.4).
- We prove generalizations of the previous results to registration problems of dimensions other than 3D/2D. In particular, we show that specific second-order regularizers are also required for 3D/1D and 2D/1D registration (Th. 3.19 and Th. 3.20).

For the cases of 3D/3D and 2D/2D registration, however, already specific first-order regularizers are appropriate (Th. 3.21). For registration settings with higher dimension, we argue that higher order regularizers may be necessary (Sec. 3.5).

1.2.2 A New Registration Method Based on Residual Neural Networks

In this thesis, we also introduce the new registration approach RNR. The purpose of this method is to determine multiple deformations that represent a movement at successive time points. In addition, RNR guarantees diffeomorphic deformations under certain conditions. This is a desired property for many applications, since diffeomorphisms are invertible and preserve properties in the deformed image; see [29] for an introduction to diffeomorphic image registration.

One application of such a method, which we also consider in this work, comes from the field of respiratory surface electromyography (sEMG) modeling. Respiratory sEMG modeling is a highly relevant field that is used, for instance, to improve mechanical ventilation of patients [5]. To model respiratory sEMG signals, the breathing movement itself must first be modeled. More precisely, the 3D torso deformation at multiple time points between end-expiration and end-inspiration are needed [5].
In the end-expiratory and end-inspiratory phase, 3D images can be recorded while the patient is holding their breath. However, in between these phases, i.e. during the breathing process, the acquisition of 3D data is difficult and costly [56]. Therefore, a common alternative is to take 2D images during breathing [56]. In order to determine the 3D deformation at the time points between end-expiration and end-inspiration, a registration method is required that takes into account several successive time points.

Our new registration method RNR allows to solve such registration problems. RNR includes data from multiple successive time points and computes deformations that reasonably build on each other. The method is an INR-based approach, more precisely a ResNet is used; see Sec. 4.2.
In this work, the method is thoroughly investigated, including its application for respiratory sEMG modeling.

In detail, our contributions are the following:

- We present the new registration method RNR (Sec. 4.2). This method is developed by adapting the INR-based registration approach of Ben Amor et al. [41]. We adjust the network structure and include data from multiple time points to meet the requirements of the considered registration task.
- We analyze whether the existence of solutions to the minimization problem within RNR is guaranteed. We prove that an explicit regularization term is required (Th. 4.1). In addition, we provide an appropriate regularization term, which ensures existence of solutions (Th. 4.3).
- We analyze whether the deformations determined with RNR are diffeomorphic. More precisely, we show that the analytical solution approximated by RNR is diffeomorphic (Th. 4.6). For the discretized solution, we present conditions which guarantee diffeomorphisms (Th. 4.7). These conditions are included as a penalty in our registration method.
- We introduce an implementation of RNR in Python based on the code basis of Wolterink et al. [38] (Sec. 4.5).

- We present a comprehensive experimental validation of RNR on multiple datasets (Exp. 4.11–4.20). For the evaluation, we visualize registration results and compute different quantitative evaluation measures.
- We validate the plausibility of RNR. More precisely, we verify that the method is stable with respect to the initialization of the network parameters, examine the impact of different registration parameters and check that the termination condition for the optimization within RNR is reasonable (Exp. 4.21–4.23).
- We compare RNR with state-of-the-art registration methods on a publicly available dataset containing lung CT data (Exp. 4.24).
- We apply the method to build a breathing model that can be used for respiratory sEMG modeling. We slightly adapt the implementation of RNR for this application (Sec. 4.7.3). Moreover, we show, using qualitative and quantitative results, that a reasonable torso deformation is obtained with RNR (Sec. 4.7.4).

1.3 Outline

This thesis is structured as follows:

- In Ch. 2, we review a framework for variational image registration and summarize strategies for the validation of registration results. This chapter forms the basis for the subsequent investigations.
- In Ch. 3, we analyze which type of regularization is suitable for 3D/2D registration. This chapter builds on the author's publication [2].
- In Ch. 4, we present our new registration approach RNR. Parts of this chapter are based on the author's publication [4], while other parts have not yet been published.
- In Ch. 5, we conclude the thesis. We summarize the main findings as well as open questions and promising directions for further research.

CHAPTER 2

A Framework for Image Registration

In this chapter, we introduce a framework for image registration and for validating the registration results. This forms the basis for the topics addressed in the following chapters.

In Sec. 2.1, we summarize fundamental notations required in this thesis. This includes the term image and spaces from functional analysis. In Sec. 2.2, we formulate image registration as a variational problem, which is a common approach. This formulation is used both in the analysis of 3D/2D registration in Ch. 3 and in the new registration approach presented in Ch. 4. In Sec. 2.3, we describe strategies for validating registration results, as this is an indispensable step when presenting new registration methods and is therefore also required in Ch. 4.

2.1 Preliminary Notations

In order to introduce the image registration framework this work is based on, we must first clarify what we mean by an image. In this thesis, we consider 2D and 3D gray scale images as defined in the following.

Definition 2.1 (Image, adapted from [57, Def. 1])**.** Let $d \in \{2, 3\}$ and let $\Omega \subset \mathbb{R}^d$ be a bounded domain, i.e. a non-empty, bounded, connected and open subset. An *image* is a continuously differentiable function $I : \mathbb{R}^d \to \mathbb{R}$ that is compactly supported in Ω. We denote the set of images on Ω by $\mathrm{Img}(\Omega)$.

We define images as one time differentiable, as this regularity is required for the similarity measures introduced in Sec. 2.2.1. However, in certain applications higher or lower regularity assumptions may also be appropriate. For example, in Ch. 3 we will assume higher regularity for the 3D image and lower for the 2D image; see Sec. 3.2 for details.

In the following, we assume that d and Ω are as in Def. 2.1, if not stated otherwise. In image registration, the image domain Ω is usually a rectangle or a rectangular cuboid [9].

In most applications, *discrete images* are used; see e.g. [10, 11, 15]. In general, a discrete image is a measurement of a continuous property of an object, such as photon density [11, Sec. 2.1]. Therefore, a discrete image is an array containing pixel/voxel values. As usual, we assume that a discrete image results from evaluation of a continuous image on a grid in the image domain Ω; see e.g. [9, Sec. 3.1.2] for a detailed introduction into discretization of images. Inversely, a continuous image can be obtained from a discrete image using an interpolation strategy; see e.g. [11, Sec. 3] for an overview of interpolation strategies.

Throughout this thesis, we will use various standard notations in order to describe and analyze deformations resulting from image registration. The most important function spaces are summarized in Tab. 2.1. For detailed definitions and further background on functional analysis, we refer to [58].
Furthermore, to keep the notation for integrals short, we omit the integration variable in many integrals, i.e. we often write $\int_\Omega f\,dx$ instead of $\int_\Omega f(x)\,dx$.

Table 2.1: *Overview of the main function spaces utilized in this thesis. The spaces $C^k(\Omega,\mathbb{R}^n)$, $C_c^k(\Omega,\mathbb{R}^n)$, ... are defined analogously to the listed spaces for functions with values in $\mathbb{R}^n$. For further notations see back matter.*

Notation	Explanation
$C^k(\Omega)$	k-times continuously differentiable functions
$C_c^k(\Omega)$	k-times continuously differentiable compactly supported functions
$C^{k,\alpha}(\Omega)$	Functions with Hölder continuous derivatives of order k
$L^p(\Omega)$	Lebesgue p-integrable functions
$W^{k,p}(\Omega)$, $H^k(\Omega)$	Sobolev space of order k with exponent p, i.e. functions in $L^p(\Omega)$ with weak derivatives up to order k in $L^p(\Omega)$, $p=2$ for $H^k(\Omega)$
$W_0^{k,p}(\Omega)$, $H_0^k(\Omega)$	Sobolev space with zero boundary values
$W_{\text{loc}}^{k,p}(\Omega)$, $H_{\text{loc}}^k(\Omega)$	Local Sobolev functions

2.2 Variational Formulation

In this section, image registration for two images is introduced as a variational problem. This approach can also be extend to align multiple images; see e.g. [14].

Given two images $T, R \in \text{Img}(\Omega)$, the goal of image registration is to align these images by determining a *reasonable deformation* $y : \Omega \to \mathbb{R}^d$ such that $T[y] := T \circ y$ and R become *similar* on the image domain Ω; see e.g. [9, Sec. 3.3]. We refer to T as the *template image* and R as the *reference image*. For the mapping y, we will use the terms *deformation* and *transformation* interchangeably.
Note that there is no general definition for the reasonability of a deformation. Instead this property depends on the application. We will give an example in Sec. 4.7, where we model the deformation of the torso during breathing.
The similarity between the deformed template and reference image is measured using a data fitting term, which we will discuss in detail in Sec. 2.2.1. Different applications require different definitions of the term similar. For example, if T and R show the same

object but from different perspectives, one might want to find a deformation y such that image intensities of $T[y]$ and R are equal. However, if the images were taken by different imaging devices, it may not be reasonable to aim for matching image intensities. Instead, in this case it may be meaningful to consider images as similar if, for example, the image gradients match; see Sec. 2.2.1 for details.

A typical approach is to formulate image registration as a variational problem; see e.g. [7, 8, 9]. Here, the deformation is determined by minimizing an *energy functional* J,

$$J[y] := D[y] + \alpha_{\mathrm{S}} S[y] + \alpha_{\mathrm{P}} P[y], \tag{1}$$

where D is a data fitting term, S is a regularizer, P is a penalty term and $\alpha_{\mathrm{S}} > 0, \alpha_{\mathrm{P}} \geq 0$. The purpose of the data fitting term D is to minimize the distance between the images, the regularizer S ensures existence of solutions and the penalty term P can be used to penalize undesirable solutions. We provide a more detailed introduction to these components in Sec. 2.2.1.
The parameters $\alpha_{\mathrm{S}}, \alpha_{\mathrm{P}}$ are referred to as regularization and penalty parameter, respectively. These balance the influence of the components in (1). Appropriate choices for the parameters depend on the application and are commonly adjusted manually [59].

Often, the desired deformation y is split into an *initial transformation* $y_0 : \Omega \to \mathbb{R}^d$ and an *update* $u : \Omega \to \mathbb{R}^d$, i.e. $y = y_0 + u$; see e.g. [1, 38, 60]. In these cases, the deformation y_0 is either the identity or results from an initial alignment, such as an affine preregistration [11]. Accordingly, the variational problem is then formulated in the update u instead of in the deformation y, i.e. an energy $J[u]$ is treated instead of $J[y]$. We will use this approach later in Ch. 3.

2.2.1 Components of the Variational Formulation

Next, we discuss the functionals in (1) individually.

Data Fitting Term

The data fitting term D is a functional that measures the similarity between the transformed template image $T[y]$ and the reference image R; see e.g. [8, 9]. We will denote D also as distance measure or similarity measure. In addition, in this thesis we assume that D reaches only values in $\mathbb{R}_{\geq 0}$, i.e. the distance is nonnegative, which is fulfilled for typical choices of D; see e.g. [11].

There is a variety of data fitting terms, each of which is suitable for certain applications; see e.g. [8, 11].
Based on [9, Sec. 6] and [11, Sec. 7], we review some commonly used similarity measures that are also utilized in this thesis: the *sum of squared differences* (SSD) / *mean squared error* (MSE), *normalized cross-correlation* (NCC) and *normalized gradient field* (NGF).
For given images $T, R \in \mathrm{Img}(\Omega)$ these data fitting terms are defined by

$$D^{\mathrm{SSD}}[y] := \frac{1}{2}\|T[y] - R\|^2_{L^2(\Omega)}, \qquad D^{\mathrm{MSE}}[y] := \frac{D^{\mathrm{SSD}}[y]}{\mathrm{vol}(\Omega)}, \tag{2}$$

$$D^{\mathrm{NCC}}[y] := 1 - \frac{\langle T[y], R\rangle^2_{L^2(\Omega)}}{\|T[y]\|^2_{L^2(\Omega)}\|R\|^2_{L^2(\Omega)}}, \tag{3}$$

$$D^{\mathrm{NGF}}[y] := \int_\Omega 1 - \left(\mathrm{n}_{T[y]}(x)^\top \mathrm{n}_R(x)\right)^2 dx, \tag{4}$$

where $\mathrm{vol}(\Omega)$ denotes the volume of Ω and $\mathrm{n}_{T[y]}$, n_R are the normalized gradient fields of the images defined as in (5). We briefly comment on each of the data fitting terms.
The SSD similarity measure is a straightforward way to compare images. It compares the image intensities in the deformed template image $T[y]$ and reference image R by calculating the squared L^2-norm of the image difference. SSD has been proven to be robust and very effective for images of one modality [15]. In addition, its minimization is related to least-squares problems, which are well understood; see e.g. [61]. However, the SSD similarity measure is based on the assumption that $T[y] = R$ and is therefore not suited for comparing images from different modalities, such as MRI and CT images.
The MSE results from SSD by normalization. Thus, it has the same advantages and disadvantages as the SSD similarity measure. While the SSD is mainly used in classical optimization-based registration [11], the MSE is its counterpart, which is mainly used in learning-based approaches [8].
The NCC is another standard measure in image registration. It aims to maximize the correlation between the images. To comprehend this, we consider the discretized form of the NCC. For this, we regard the images as vectors containing the pixel/voxel values and replace the L^2 scalar product and norm with the Euclidean scalar product and the Euclidean norm. Due to $\langle T[y], R\rangle_2 = |T[y]||R| \cos \sphericalangle(T[y], R)$ the NCC contains the cosine of the angle between $T[y]$ and R. More precisely, the NCC value becomes small when the cosine reaches the value 1 or -1. Therefore, a minimal NCC value is achieved when there is a linear dependence between the images, i.e. $T[y] = \lambda R$ for some $\lambda \in \mathbb{R}$. In this sense, NCC is an extension of the SSD/MSE similarity measure. It has proven to be reliable in many applications; see e.g. [3, 31]. However, like the SSD/MSE data fitting term, this measure cannot compare images from modalities where the gray values differ not only in terms of scaling [11].
The NGF similarity measure is able to compare images from different modalities with differences in the gray values for corresponding structures [11]. Hence, it is a common choice for multimodal image registration; see e.g. [62]. It aims to align the gradient fields of $T[y]$ and R. This is motivated by the observation that the content of an image is also represented by changes in intensity, i.e. the image gradient. As the gradient also measures the strength of the intensity changes, which is unwanted for multimodal image registration, normalization of the gradient is required. Thus, the NGF similarity measure is based on the *normalized gradient field*, which is given by

$$\mathrm{n}_I(x) := \mathrm{n}_{I,\eta}(x) := \frac{\nabla I(x)}{\sqrt{\|\nabla I(x)\|_2^2 + \eta^2}} \tag{5}$$

for an image $I \in \mathrm{Img}(\Omega)$. Here, the so-called edge parameter $\eta \geq 0$ determines what is regarded as edge ($\|\nabla I(x)\|_2 > \eta$) or noise ($\|\nabla I(x)\|_2 < \eta$) and therefore plays an important role in the parameter tuning for the optimization process. Note that for the computation of the NGF data fitting term the images need to be differentiable, while this is not necessary for the evaluation of SSD/MSE and NCC. Therefore, the analysis of registration problems using NGF may differ from the analysis of problems using SSD/MSE and NCC. In the investigation of 3D/2D registration in Ch. 3, the additional regularity requirement yields that NGF is not covered, while SSD/MSE and NCC fulfill the working assumptions; see Sec. 3.4 for details.

Performing image registration by using only the data fitting term in the energy functional given by (1) is an ill-posed problem; see [15]. A problem is *well-posed* in the sense of Hadamard [63] if there exists a solution, the solution is unique and depends continu-

ously on the data. If one of these criteria is not fulfilled it is called *ill-posed*. Regularization addresses the ill-posedness and is thus important and inevitable.

Regularization Term

The regularization term S is used to ensure the existence of solutions, but also to incorporate prior knowledge about the expected solution and to increase the robustness against noise [15]. Uniqueness of solutions cannot be expected in general for image registration problems and is therefore not the focus of the regularization term; see e.g. [15, 57].

As the particular choice of the regularizer considerably affects the solution of the registration problem [15], a wide range of regularization terms exists; see e.g. [9, 59] for an overview of regularization strategies.
Based on [11, Sec. 8], we summarize some typical regularizers that are also used in this thesis: the *elastic*, *diffusion* and *curvature regularizer*.
These regularizers are defined by

$$S^{\text{elastic}}[y] := \frac{1}{2}\int_\Omega \mu \sum_{j=1}^{d} \|\nabla y_j\|_2^2 + (\lambda+\mu)(\operatorname{div} y)^2\, dx, \tag{6}$$

$$S^{\text{diffusion}}[y] := \frac{1}{2}\int_\Omega \sum_{j=1}^{d} \|\nabla y_j\|_2^2\, dx, \tag{7}$$

$$S^{\text{curvature}}[y] := \frac{1}{2}\int_\Omega \sum_{j=1}^{d} (\Delta y_j)^2\, dx. \tag{8}$$

The elastic regularizer is physically inspired and based on the assumption that the depicted objects deform like elastic materials. It has been first introduced in 1973 by Fischler and Elschläger [64] and for image registration in 1981 by Broit [65] and is probably the most commonly used regularizer in image registration [9]. The elastic regularizer contains first-order derivatives and therefore belongs to the class of so-called *first-order regularizers*. The parameters μ, λ with $\mu > 0$, $\lambda \geq -\mu$ in (6) describe material properties of the underlying elastic model and are referred to as Lamé constants.
In contrast to the physically motivated elastic regularization, the idea behind diffusion regularization is to minimize oscillations of the deformation. This regularizer has been proposed for optical flow problems by Horn and Schunk [26]. The diffusion regularizer also belongs to the class of first-order regularizers and results from the elastic regularization term by setting $\mu = -\lambda = 1$. Note that there is no coupling between the components of the deformation. This allows for fast numerical schemes; see e.g. [66].
The curvature regularizer was proposed for image registration by Fischer and Modersitzki [67]. It is based on second-order derivatives and thus typically results in smoother transformations compared to elastic- or diffusion-based registration [11]. There is a close relation between curvature-based registration and the computation of so-called thin-plate-splines, which allow for landmark-based registration; see e.g. [9]. As thin-plate-splines are designed to minimize the bending energy of the deformation, this connection offers an illustrative interpretation of the curvature-based approach [67]. According to [67], a further reason for using the curvature regularizer is that affine linear transformations are contained in the kernel. Therefore, curvature-based registration is less dependent on the initial position of the template and the reference image than, for example, elastic- or diffusion-based registration [9].

In addition to explicit regularization through the regularization term in (1), implicit regularization is used as well to restrict the space of admissible deformations. Implicit regularization arises, for example, from parameterization of the deformation with B-Splines [68] or from a coarse-to-fine multilevel approach [69]. Later in Ch. 4, we will use a combination of an explicit and an implicit approach for the regularization.

Penalty Term

The penalty term P can be used to penalize undesirable solutions. In contrast to a regularizer, which is needed to ensure solutions of the registration problem, the penalty is an add-on that can serve as a tool for incorporating user knowledge [15]. For instance, a penalty term can be utilized to preserve certain image structures [1].
In Ch. 4, we will use two different penalty terms whose purpose we briefly summarize here. The first is a gradient-based term that favors diffeomorphic deformations, which are desirable as they are invertible and preserve properties in the deformed image; see Sec. 4.4–4.5 for details. The second is tailored to the calculation of a breathing model and avoids unrealistic deformations of the rib cage by penalizing scalings of the rib cage in the vertical axis; see Sec. 4.7.

An alternative to using penalty terms for controlling the deformation is provided by constraints. Unlike the penalty approach, where unwanted solutions are penalized, undesirable deformations are completely excluded here; see e.g. [15, 19] for further details.

2.2.2 3D/2D Registration

In contrast to image registration where the template and the reference image have the same dimensionality, in 3D/2D registration the template image T is a 3D image and the reference image R is a 2D image. Accordingly, the desired deformation y is also a 3D mapping. In order to bring the deformed image $T[y]$ and R into dimensional correspondence, we transform the 3D image $T[y]$ into a 2D image; see Sec. 1.1.2 for alternative approaches. The goal of 3D/2D registration is to determine a deformation y such that $P[T \circ y]$ and R become similar. Here, P maps the 3D image $T[y] = T \circ y$ to a 2D image using some projection; see e.g. [70].

Variational formulations are also used to solve 3D/2D registration problems; see e.g. [71]. However, when solving 3D/2D registration problems with a variational approach, the data fitting term measures the image similarity between two 2D images, while the regularization term operates in 3D. Since we examine the influence of the regularization order in variational 3D/2D registration in Ch. 3, we will introduce the setting of 3D/2D registration in more detail there.

2.3 Validation Criteria

Validation of registration results is indispensable when introducing new registration methods. Before a new method is used in practice, for example for medical applications, it is mandatory to ensure that the method produces reasonable results [6]. Therefore, we will also present a comprehensive validation of our new registration method RNR later in Ch. 4.

Validation of image registration is a non-trivial task [18]. Since the goal of image registration is to compute a deformation, meaningful validation should evaluate its quality.

However, a direct evaluation is generally not possible, as ground truth deformations are usually not available [19].

A commonly used strategy is to assess the registration quality through visual inspection based on qualitative results [6]. A standard approach is to depict the template, the reference and the deformed template image next to each other, as in the introductory example; see Fig. 1.1. This allows a direct comparison of the undeformed and deformed images; see e.g. [11]. Another method is the visualization of difference images, which are created by subtracting the template from the reference image or vice versa. Difference images enable the illustration of subtle differences between the template and the reference images; see e.g. [11]. Further strategies include displaying contour or segmentation overlays, which enable a comparison of corresponding structures in the images [6]. A more detailed description of the visualization approaches used in the validation of RNR will be presented in Sec. 4.6.1.

There exists also a variety of quantitative measures to evaluate the registration quality; see e.g. [8]. In this thesis, we will use the following typical quantitative measures: the *mean squared error* (MSE), *target registration error* (TRE) and the *Dice score*. We briefly introduce each of them here based on [8].
The MSE, see (2), calculates the difference between the image intensities in the deformed template and the reference image. As discussed above this is a reasonable measure to assess the similarity of images taken by the same imaging device; see Sec. 2.2.1. Analogously, the other distance measures presented in Sec. 2.2.1 can be used for the validation of registration results. One advantage of using such a distance measure for the evaluation is that it relies solely on the images and does not require the extraction of additional features, as is the case with the evaluation measures presented below. Note however that for the comparison of different registration methods, caution must be exercised when using a distance measure as evaluation criterion. Assume that a method uses a certain data fitting term for the calculation of the deformation. Then, during validation, a smaller value of this term is to be expected compared to a method based on a different data fitting term. But this does not necessarily mean that the registration quality of the first method is better with respect to other evaluation measures.

A common approach to circumvent this problem is to assess the registration quality based on landmarks [8]. Landmarks are corresponding spatial positions in the template and reference image. In general, these landmarks are not utilized during the registration, but only for validation, allowing the results of different registration methods to be compared. The TRE computes the distance between such corresponding landmarks in the template and reference image. More precisely, when landmark pairs $(t_j, r_j)_{j=1,\dots,N}$ with $t_j, r_j \in \Omega$ are available, the mean TRE is defined by

$$\mathrm{TRE}[y] := \frac{1}{N} \sum_{j=1}^{N} |y(r_j) - t_j|. \tag{9}$$

The TRE is often regarded as the gold standard for evaluating registration results and is widely used to compare registration methods [8]. Therefore, we will also use the TRE to compare our RNR method with state-of-the-art registration approaches; see Sec. 4.6.5. A disadvantage of this measure is that its computation is based on the availability of corresponding landmarks. These landmarks are usually annotated manually by experts, but there are also algorithms for automatic landmark detection; see [8] and references

therein. The usability of the TRE for the validation of registration results is highly dependent on the accuracy of the landmark detection. Note also that, unlike the MSE, the TRE considers only a sparse representation of the deformation. It does not provide any information about the quality of the registration result in the area between the landmarks.

Another frequently used tool for the validation of registration results is the Dice score [72, 73]. In contrast to the TRE, the Dice score considers a broader part of the images by measuring the alignment of corresponding image sections. More precisely, for two sets A and B it is defined as

$$\text{Dice}[A, B] := \frac{2|A \cap B|}{|A| + |B|}. \tag{10}$$

Thus, it is one when the two sets overlap completely and zero when the sets are disjoint. In order to validate registration results, the Dice score is computed for corresponding segmentations in the template and reference image; see e.g. [3, 4]. Later in Sec. 4.7, we will use this measure to assess the alignment of different torso parts, such as the lungs and the diaphragm, achieved with RNR. The Dice score provides an easily calculable value that indicates how well two segmentations overlap. It is particularly useful when the focus of an application is on aligning specific parts of the images. However, note that the significance of the Dice score as an evaluation criterion depends on the quality of the segmentations. Since image segmentation is still a difficult task [74], these segmentations may be inaccurate. In addition, similar to the TRE, the Dice score only provides information about the registration quality in a very specific area. Another disadvantage is that it cannot always be interpreted uniformly. While a small misalignment has little effect on the Dice score for large structures, even a slight misalignment can impact the Dice score considerably for thin structures. In Sec. 4.7, we will observe this effect, for example, for the volumetric lungs and the rather flat diaphragm.

Summarizing, validation of registration results is a challenging task and each evaluation measure has different advantages and disadvantages. Therefore, a reasonable strategy is to base the validation on several different qualitative and quantitative measures in order to obtain a more comprehensive impression of the registration results.

CHAPTER 3

Appropriate Regularization in 3D/2D Image Registration

In this chapter, we analyze the influence of the regularization order in variational 3D/2D image registration. More precisely, we address the following problem:

Problem 1. Specify which type of regularization ensures continuously differentiable deformations in variational 3D/2D image registration.

We will show that certain second-order regularizers are sufficient, while first-order regularizers are generally insufficient.

To illustrate the relevance of solving Problem 1, we exemplary consider the volume-to-slice registration task depicted in Fig. 3.1. Here the template is a 3D MRI image, while the reference is a 2D Cine MRI slice image. The goal is to find a 3D deformation such that the deformed template image is aligned with the reference image on the 2D slice. For simplicity, we focus on the deformation of the lung, which is highlighted in green. Since the reference image is given just on the 2D slice, the image similarity between the template and the reference image can only be measured on this slice. Therefore, the image comparison does not provide any information about the deformation of the lung outside the slice. This makes the computation of plausible 3D deformations a hard task. However, the lung movement on the considered slice is also representative for a region around this slice. To model this, smooth deformations are reasonable, as the smoothness allows to transfer information about the deformation from the slice to the outside. But the higher the regularity we assume for the deformation, the smaller the space of admissible deformations. Continuously differentiable deformations are a compromise, while still being meaningful [55, 75].
Since it is the role of the regularizer to ensure reasonable solutions, the question arises of which type of regularization guarantees continuously differentiable deformations in 3D/2D registration.

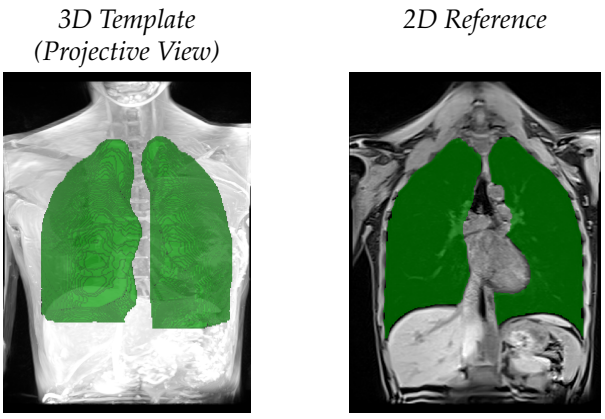


Figure 3.1: *Visualization of data defining a 3D/2D registration problem: The template is a 3D MRI image and the reference is a 2D Cine MRI slice image. The lung motion on the slice is representative for a region around this slice. A reasonable class of transformations to model the lung motion are continuously differentiable deformations, which motivates solving Problem 1.*

An experiment that further highlights the impact of the regularization in 3D/2D registration is presented in detail in the next section; see Exp. 3.1. This experiment builds on the work of Heldmann and Papenberg [50] and suggests that second-order regularization is required to guarantee continuously differentiable deformations.
However, to the best of our knowledge, a rigorous investigation of Problem 1 was still missing prior to our work [2], on which this chapter is based.

In the following, we address Problem 1 by proving that specific second-order regularizers are sufficient to guarantee continuous differentiability of the deformations, while first-order regularizers are generally not. In particular, the work of [50], which is summarized in the subsequent section, is extended in various directions:

1. We consider a more general 3D/2D registration setting that incorporates volume-to-slice registration, as investigated in [50].
2. We rigorously prove that certain second-order regularization terms guarantee continuously differentiable deformations (Th. 3.13), and first-order regularization fails to generate continuously differentiable deformations on a large class of examples (Th. 3.15). These statements explain the numerical experiments in [50].
3. We also show that commonly used data fitting and regularization terms, including those considered in [50], satisfy the assumptions on which our proofs are based.

This chapter is organized as follows: First, in Sec. 3.1 we summarize previous contributions on Problem 1. In particular, we outline the work of [50] and explain how our analysis builds on their results. Moreover, we emphasize the significance of solving Problem 1 ones more by discussing a numerical example that uses different types of regularizers. In Sec. 3.2, we introduce the 3D/2D image registration setting of this chapter and formulate assumptions on the data fitting and regularization terms. Based on this, we present our main results regarding first- and second-order regularization in Sec. 3.3. For both cases detailed proofs are provided here. Afterwards, in Sec. 3.4 we show that the assumptions on which the proofs are based hold for common data fitting and regularization terms in 3D/2D registration. Finally, in Sec. 3.5 we discuss generalizations of

our results to registration problems of other dimensions than 3D/2D. In particular, we show that the results from the previous sections also apply for 3D/1D and 2D/1D registration, while already specific first-order regularizers are appropriate in the common cases of 3D/3D and 2D/2D registration, and higher order regularizers may be necessary in registration settings with higher dimension. We conclude the chapter with a summary of our contributions as well as ideas for future investigations in Sec. 3.6.

This chapter builds on the author's publication [2]. Therein, the author conceived and developed the ideas and wrote the manuscript in close cooperation with Dr. Florian Mannel, while the numerical experiments were implemented by the author alone.

3.1 Related Work

In this section, we summarize how previous works have contributed to solving Problem 1, i.e. ensuring appropriate deformations in 3D/2D registration. In addition, we further stress the relevance of the question of what kind of regularization guarantees continuously differentiable deformations.

Though guaranteeing suitable deformations is highly relevant, to the best of our knowledge there is little work investigating the impact of the regularization order on the deformation in 3D/2D registration.

3.1.1 Approaches Based on Additional Information

Many 3D/2D registration procedures avoid the problem of choosing a suitable explicit regularization by parameterizing the deformation. Using parameterizations it is easy to determine whether the resulting deformation is continuously differentiable. For example, in rigid 3D/2D registration methods [76, 77] the resulting deformation is naturally continuously differentiable due to the selected parameterization. Another common approach is the use of spline-based transformations, such as in [78]. Here, it depends on the type of splines used whether the resulting deformation is continuously differentiable or not. However, the disadvantage of parametric image registration approaches is that the choice of the basis functions is in general artificial and must be justified [9].
Another possibility to circumvent Problem 1 is to incorporate knowledge about the deformation provided by the imaging method. For example, Striewski and Wirth [71] presented a nonparametric registration approach for optical microscopy images. An advantage of this application is that the out-of-focus blur in the 2D data contains additional information about the underlying deformation. Using this information only, appropriate deformations can be computed and it is not necessary to ensure continuously differentiable deformations. Thus, Problem 1 does not need to be taken into account here. However, registration of optical microscopy images is a special case. In applications working with images from other modalities, for example X-ray images, such additional information about the deformation cannot be extracted and thus guaranteeing continuously differentiable deformations is desirable.

3.1.2 Numerical Investigation of the General Variational Setting

The seminal work of Heldmann and Papenberg [50] is the first that analyzes the impact of regularization in variational 3D/2D registration without additional information about

the deformation. Therein, the main focus is on a numerical investigation of Problem 1. We summarize their contributions in more detail next.

Heldmann and Papenberg [50] focus on volume-to-slice registration, like in the motivating example in the introduction of this chapter. For this setting the authors of [50] provide a concrete solution to Problem 1. They claim the following: Second-order regularization leads to continuously differentiable deformations, while first-order regularization yields non-differentiable deformations.
In order to justify this statement, as proof-of-concept, specific first- and second-order regularization schemes are investigated. These are diffusion as first-order regularization and curvature as second-order regularization. Since the focus in [50] is on the discretization of the considered registration problem, numerical experiments are performed to compare the deformations resulting from registration using diffusion and curvature regularization. The presented numerical results confirm their statement that second-order regularization is required to obtain continuously differentiable deformations.

As the numerical experiments of Heldmann and Papenberg [50] illustrate the impact of different regularizers on the registration result very well, we reproduce one of their experiments here to motivate solving Problem 1. Therefore, in Exp. 3.1 we perform two volume-to-slice registrations. To emphasize kinks in the deformation we use images without corners and edges, which therefore differ from those in [50]. Nevertheless, our findings are the same as in [50].

Experiment 3.1 (Impact of regularization order)**.** To emphasize the impact of the regularizer in 3D/2D registration, we perform two volume-to-slice registrations, one with first-order regularization and one with second-order regularization. Fig. 3.2 depicts the template and reference image as well as the deformed template images resulting from both registrations. For each image, a 3D visualization is shown in the first row, while two orthogonal slices of the 3D image are presented in the second and third row. The images are grayscale, but for clearer visualization the intensity values are displayed in color.
The 3D template image depicts a ball; see first column of Fig. 3.2. The reference image is a 2D slice of the image in the second column of Fig. 3.2. More precisely, this slice is the red-bordered subset of the x_2-x_3-plane shown in the third row. The balls corresponding to the template and the reference have the same radius but different distributions of the image intensities.
In order to perform the volume-to-slice registration, we minimize an energy functional of the form

$$J[y] = D[y] + \alpha S[y],$$

where D is a data fitting term measuring the image similarity only on the red-bordered slice in Fig. 3.2, S is a regularizer and $\alpha > 0$ is a weighting parameter. As in [50], we use SSD as similarity measure for simplicity. Also analogous to [50], we utilize diffusion as first-order regularization and curvature as second-order regularization. The regularization parameter α is chosen such that after registration the same value in the data fitting term is achieved for both choices of the regularizer. The implementation is based on the FAIR toolbox [11] using Matlab 9.13.0 [79]; for the sake of brevity, we will not go into further details of the implementation here.
The deformed template images obtained from registration with diffusion and curvature

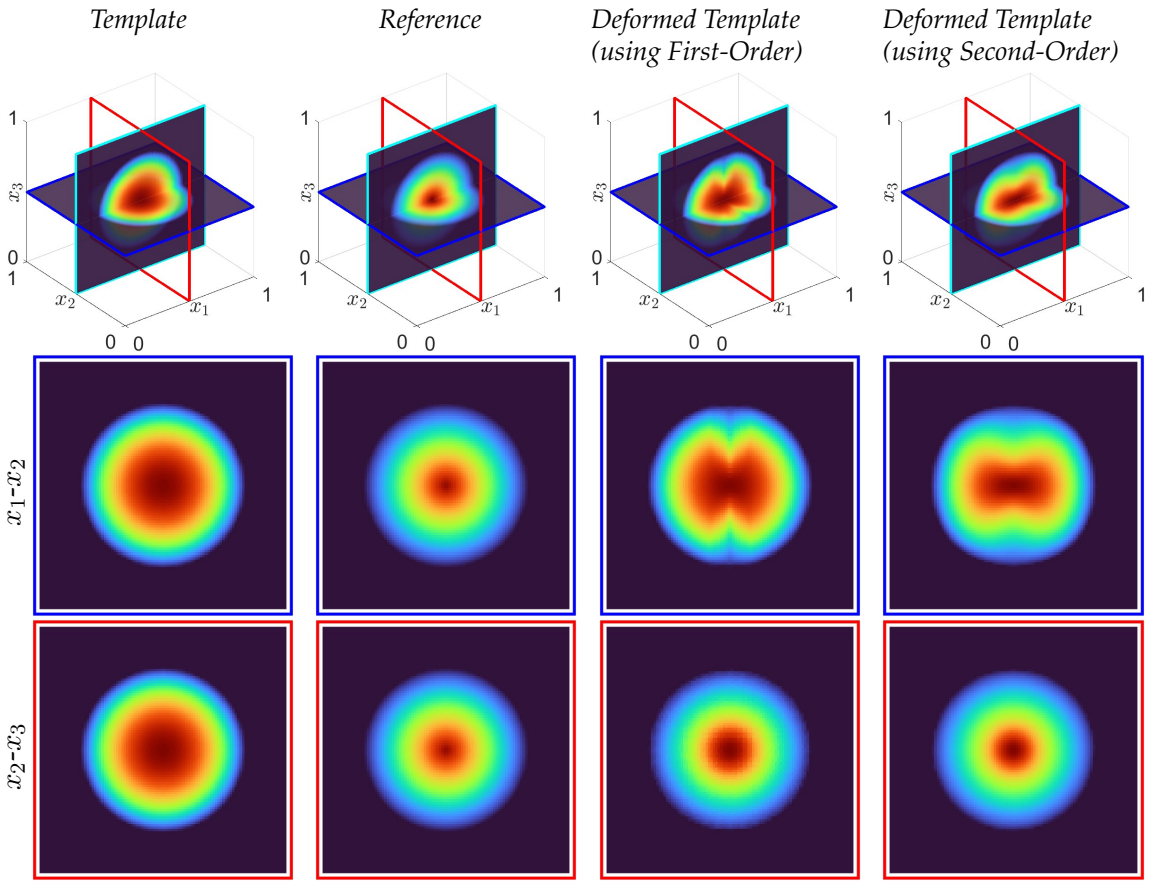


Figure 3.2: *Initial and deformed images originating from a registration problem based on [50], which illustrates the impact of the regularization in volume-to-slice registration: The template image (first column) is deformed to match the reference image (second column). The image similarity is measured on the red-bordered slice in the x_2-x_3-plane and thus the data fitting term aims to align the images on this subset only (third row). Using second-order regularization (fourth column) yields more transfer of the available 2D information to the 3D domain than using first-order regularization (third column). For further details see Exp. 3.1.*

regularization are visualized in the last two columns of Fig. 3.2. The results of both registration approaches hardly differ on the red-bordered slice, on which the image similarity is measured. However, the 3D visualizations as well as the blue-bordered pictures show that the regularization has a major impact on the registration, as the results obtained by the two registrations are very different.
When using first-order diffusion regularization, the template image is only deformed in the vicinity of the red-bordered 2D slice, i.e. the information about the deformation on this slice is not transferred well to the 3D volume. In addition, the visualizations in the third column of Fig. 3.2 suggest that the underlying calculated deformation has kinks and may not be continuously differentiable. In contrast, the deformation obtained using second-order curvature regularization yields much more transfer of the available information on the red-bordered 2D slice to the 3D volume. Furthermore, the pictures in the last column of Fig. 3.2 suggest that the calculated deformation is continuously differentiable.

In conclusion, these considerations show that the regularizer has a significant influence on the registration. Additionally, this experiment provides a 3D/2D registration problem where continuously differentiable deformations are reasonable. And finally, this experiment also confirms the statement by Heldmann and Papenberg [50] that second-order regularization is required to obtain continuously differentiable deformations.

However, a comprehensive proof of this assertion is not provided in [50], as it is not the focus of the paper. Regardless of this gap, subsequent work on 3D/2D image registration is based on [50] and uses second-order regularization [80, 81]. In the following, we will fill this gap.

3.2 3D/2D Registration Setting

In this section, we introduce the 3D/2D image registration setting this chapter is based on in detail. First, we present the problem of variational image registration that we will investigate. We then formulate assumptions on the data fitting and the regularization terms, on which our main results in the next section are based. These assumptions apply for common similarity measures and regularizers, as we discuss in Sec. 3.4.

3.2.1 Variational Formulation

In Sec. 2.2, we introduced variational image registration for 3D/3D and 2D/2D registration and briefly discussed the adaptations required for formulating the 3D/2D setting. Now we specify the latter and present the 3D/2D registration setting under consideration in detail.

In 3D/2D registration the template image T is a 3D image, while the reference image R is a 2D image; see Sec. 2.2.2. According to Def. 2.1, this means that $T : \mathbb{R}^3 \to \mathbb{R}$ and $R : \mathbb{R}^2 \to \mathbb{R}$ are compactly supported functions on two domains of interest $\Omega \subset \mathbb{R}^3$ and $\Sigma \subset \mathbb{R}^2$. Based on the embedding $\mathbb{R}^2 \to \mathbb{R}^3$, we deviate slightly from Def. 2.1 and suppose for the remainder of this chapter that $\Sigma \subset \mathbb{R}^3$ and thus also $R : \mathbb{R}^3 \to \mathbb{R}$. An exemplary registration setting is visualized in Fig. 3.3.
Without loss of generality, we further assume that Σ is a 2D affine subset of Ω (justified

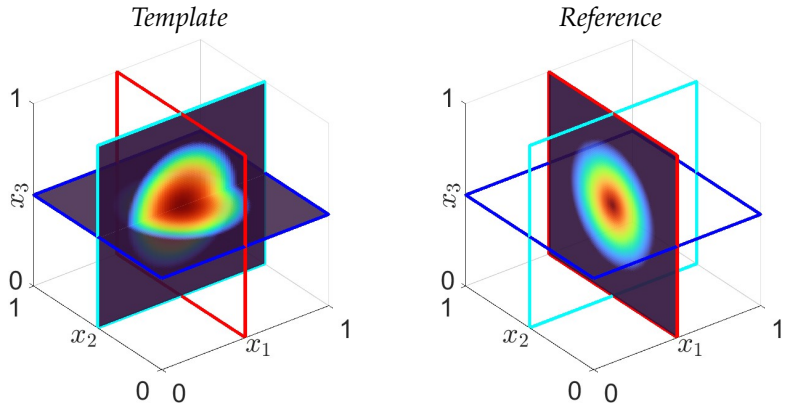


Figure 3.3: *Exemplary 3D/2D registration setting considered in this chapter. The template is a 3D image, while the reference is a 2D image embedded in $\mathbb{R}^3$; see Sec. 3.2.1 for further details.*

again by an appropriate embedding). In addition, we assume that the domain Ω is a bounded rectangular cuboid. This does not pose restrictions for practical 3D/2D registration registration problems, as 3D image domains are usually rectangular cuboids.
Regarding the regularity assumption on the images we deviate further from Def. 2.1 and suppose $T \in C^2(\Omega)$ and $R|_\Sigma \in L^2(\Sigma)$. Thus, we assume higher regularity on T and lower regularity on R compared to Def. 2.1.

As introduced in Sec. 2.2.2, the objective of 3D/2D image registration is to establish a deformation $y : \Omega \to \mathbb{R}^3$ that aligns the deformed 3D template image $T[y]$ with the 2D reference image R. Since we follow a variational approach, the deformation y results from the minimization of an energy functional in the form of (1). More precisely, in this chapter we consider deformations $y = y_0 + u$, where $y_0 : \Omega \to \mathbb{R}^3$ is a fixed initial deformation and $u : \Omega \to \mathbb{R}^3$ is an additional update deformation; see also Sec. 2.2. The function u is calculated my minimizing $J : H^k(\Omega, \mathbb{R}^3) \to \mathbb{R}$ given by

$$J[u] = D_{y_0}[u] + \alpha S[u], \tag{11}$$

where D_{y_0} is a data fitting term that depends on the fixed deformation y_0, and S is a regularizer. We denote by $k \in \mathbb{N}$ an integer such that D_{y_0} and S are both defined on the Sobolev space $H^k(\Omega, \mathbb{R}^3)$. As will become clear later in Ass. 3.2, the number k indicates the order of regularization. Thus, we use $k = 1$ for first-order regularization and $k = 2$ for second-order regularization.

For the remainder of this chapter, without loss of generality, we set the regularization parameter to $\alpha = 1$. In addition, we assume that the initial deformation y_0 is given and sufficiently smooth so that the smoothness of the deformation y is directly related to the smoothness of the deformation u. In applications y_0 may result from preregistration or is simply the identity on $\mathbb{R}^3$.
For the update deformation u, we suppose the Dirichlet boundary conditions

$$u = 0 \quad \text{on } \partial\Omega.$$

This ensures that the final deformation y respects the boundary values given by the initial deformation y_0. We impose this condition to keep the analysis brief. In practice, however, this restriction is compensated by the initial deformation y_0, so that there are no restrictions associated with this condition. Due to the Dirichlet boundary conditions, when minimizing (11) we consider deformations in

$$\mathbb{H}^k := H^k(\Omega, \mathbb{R}^3) \cap H_0^1(\Omega, \mathbb{R}^3),$$

which is the subspace of $H^k(\Omega, \mathbb{R}^3)$ of functions with zero boundary values; see Tab. 2.1.

In this chapter, we will not address the question of whether a local minimizer $u \in \mathbb{H}^k$ exists. Instead, we assume that this is the case and focus on the question of how regular the deformation u is, i.e. whether the solution is reasonable for 3D/2D registration. For existence results in variational image registration, we refer e.g. to [82, 83, 84].

In the next section, we summarize assumptions related to the data fitting and regularization terms.

3.2.2 Assumptions on the Data Fitting and Regularization Terms

Next, we present further assumptions that are used to establish the main results of this chapter in Sec. 3.3. A brief overview of how each of the assumptions is incorporated into the proofs is already given at the end of this section. Later in Sec. 3.4, we will show that the assumptions are fulfilled for commonly used data fitting and regularization terms.

First, we introduce the assumptions on the data fitting term.

Assumption 3.1 (Data fitting term)**.**

a) The data fitting term D_{y_0} is Gateaux differentiable on $\mathbb{H}^k$. That means, for any $u \in \mathbb{H}^k$, the functional

$$F : \mathbb{H}^k \to \mathbb{R}, \qquad F[v] := d_v D_{y_0}[u]$$

is linear, where $d_v D_{y_0}[u] := \lim_{t\to 0} \frac{D_{y_0}[u+tv]-D_{y_0}[u]}{t}$.

b) At any $u \in \mathbb{H}^k$, the Gateaux derivative $v \mapsto d_v D_{y_0}[u]$ can be extended from $\mathbb{H}^k$ to $H_0^1(\Omega, \mathbb{R}^3)$ and its extension is a bounded linear functional. That is, for any $u \in \mathbb{H}^k$, the functional

$$F : H_0^1(\Omega, \mathbb{R}^3) \to \mathbb{R}, \qquad F[v] := d_v D_{y_0}[u]$$

is well-defined and satisfies $F \in H^{-1}$, where H^{-1} denotes the topological dual space of $H_0^1(\Omega, \mathbb{R}^3)$.

The remaining assumptions relate to the regularization term. While the next condition refers to all regularizers regardless of their order, the latter condition requires at least second-order regularization.

Assumption 3.2 (Regularization term)**.** The regularizer S is given by

$$S[u] = \frac{1}{2} a[u, u],$$

where $a : \mathbb{H}^k \times \mathbb{H}^k \to \mathbb{R}$ is a continuous and coercive bilinear form defined by

$$a[u, v] := \langle Bu, Bv \rangle_{L^2(\Omega,\mathbb{R}^\ell)},$$

with a kth-order differential operator $B : \mathbb{H}^k \to L^2(\Omega, \mathbb{R}^\ell)$ for some $\ell \in \mathbb{N}$. We call S a *regularization of order* k.

In order to formulate the other assumption on the regularizer, we need to introduce the Legendre-Hadamard condition. This condition implies, in particular, the coercivity of a certain bilinear form considered in the proof of Th. 3.13. Coercivity is necessary to show existence of a solution using the Lax-Milgram theorem.

Definition 3.2 (Legendre-Hadamard condition [85], adapted to constant matrices)**.** For a 4-tensor A, which we understand as a matrix $A \in \mathbb{R}^{m^2 \times n^2}$, and matrices $\xi, \theta \in \mathbb{R}^{m \times n}$, we define

$$\xi^\top A \theta := \sum_{\alpha,\beta=1}^{n} \sum_{i,j=1}^{m} A_{ij}^{\alpha\beta} \xi_{i\alpha} \theta_{j\beta}. \tag{12}$$

With this, $A \in \mathbb{R}^{m^2 \times n^2}$ satisfies the *Legendre-Hadamard condition* if there exists a constant $\lambda > 0$ such that

$$\xi^\top A \xi \geq \lambda \|\xi\|_F^2 \quad \forall \xi = a \otimes b \tag{13}$$

with the matrix $\xi \in \mathbb{R}^{m \times n}$ defined through vectors $a \in \mathbb{R}^n$, $b \in \mathbb{R}^m$ by $\xi_{i\alpha} = a_\alpha b_i$.

The following example provides a matrix, which fulfills the Legendre-Hadamard condition. We will refer to this example later in this chapter, as the matrix is related to the second-order curvature regularizer.

Example 3.3. Let $m = n = 3$ and define $A \in \mathbb{R}^{3^2 \times 3^2}$ through

$$A_{ij}^{\alpha\beta} := \begin{cases} 1, & \text{if } \alpha = \beta \text{ and } i = j, \\ 0, & \text{else.} \end{cases}$$

The matrix A satisfies the Legendre-Hadamard condition (13), since

$$\sum_{\alpha,\beta,i,j=1}^{3} A_{ij}^{\alpha\beta} \xi_{i\alpha} \xi_{j\beta} = \sum_{\alpha,i=1}^{3} (\xi_{i\alpha})^2 = \|\xi\|_F^2 \quad \forall \xi \in \mathbb{R}^{3,3}.$$

Based on the Legendre-Hadamard condition, we formulate the following additional assumption, which requires at least second-order regularization.

Assumption 3.3 (Second-order regularization term)**.** There holds $k \geq 2$ and there exists $A \in \mathbb{R}^{3^2 \times 3^2}$ satisfying the Legendre-Hadamard condition as well as the symmetry condition $A_{ij}^{\alpha\beta} = A_{ji}^{\beta\alpha}$ such that the differential operator B is given by

$$B : \mathbb{H}^k \to L^2(\Omega, \mathbb{R}^3), \quad Bu = A \cdot \nabla^2 u.$$

In $A \cdot \nabla^2 u$ we understand A as a matrix $A \in \mathbb{R}^{3 \times 3^3}$ and the Hessian $\nabla^2 u$ as a vector $\nabla^2 u \in \mathbb{R}^{3^3}$, i.e. for $i = 1, 2, 3$ there holds $(Bu)_i = \sum_{\alpha,\beta,j=1}^{3} A_{ij}^{\alpha\beta} \partial_{x_\alpha x_\beta} u_j$.

In order to get a first idea how Ass. 3.1–3.3 are used in Sec. 3.3 to prove the main results of this chapter, we now briefly comment on each assumption:
The first part of Ass. 3.1 is utilized together with Ass. 3.2 in order to state that any local minimizer of the energy functional J satisfies a necessary optimality condition; see Th. 3.11. This necessary optimality condition forms the basis for the analysis of both first-order and second-order regularization. The second part of Ass. 3.1 and Ass. 3.3 are

relevant only for our investigation regarding second-order regularization. With these assumptions the necessary optimality condition induces higher regularity on the deformation u, from which continuous differentiability for u follows; see Th. 3.13.

3.3 Results for the Regularization Order in 3D/2D Registration

Based on the setting introduced in the previous section, we can now formulate the main results of this chapter. These results address Problem 1 and thus the question of which kind of regularization guarantees continuously differentiable deformations.
For this purpose, we first summarize functional analysis statements in Sec. 3.3.1, which are relevant in the proofs later this section. Then, in Sec. 3.3.2 we state a necessary optimality condition that forms the basis for the analysis of both first-order regularization and second-order regularization. Thereafter, we focus on second-order regularization in Sec. 3.3.3 and on first-order regularization in Sec. 3.3.4. In particular, in Sec. 3.3.3 we rigorously prove that second-order regularization guarantees continuously differentiable deformations in 3D/2D registration. However, in Sec. 3.3.4 we show that this does not hold for first-order regularization. To this end, we present a class of registration problems that necessarily yield deformations, which are not continuously differentiable.

3.3.1 Preliminaries

First, we summarize theorems from functional analysis that we will use to prove our main results. We adhere to the order in which the statements are used later.

Therefore, we begin with stating the Lax-Milgram theorem, which in particular provides a unique solution for the necessary optimality condition for a local minimizer of (11).

Theorem 3.4 (Lax-Milgram theorem [86, Sec. 6.2.1, Th. 1])**.** Let U be a real Hilbert space. Assume that $b : U \times U \to \mathbb{R}$ is a bilinear mapping, for which there exist constants $c_1, c_2 > 0$ such that

1. $|b[u, v]| \leq c_1 \|u\|_U \|v\|_U \quad \forall u, v \in U,$ (Continuity)
2. $b[u, u] \geq c_2 \|u\|_U^2 \quad \forall u \in U.$ (Coercivity)

Finally, let $f : U \to \mathbb{R}$ be a bounded linear functional on U.
Then there exists a unique element $u \in U$ such that

$$b[u, v] = f[v] \quad \forall v \in U.$$

Next, we present three theorems utilized in Sec. 3.3.3 in order to prove that second-order regularization guarantees continuously differentiable deformations.
In this proof, the following result will provide coercivity of a certain bilinear form. The coercivity result is then used to apply the Lax-Milgram theorem in order to show existence of unique solutions of a system of equations considered there.

Theorem 3.5 (Gårding's inequality [85, Th. 1.6])**.** Assume that $A \in \mathbb{R}^{m^2 \times n^2}$ satisfies the Legendre-Hadamard condition (13) for some positive constant λ and the symmetry condition $A_{ij}^{\alpha\beta} = A_{ji}^{\beta\alpha}$. Then

$$\int_{\mathbb{R}^n} \nabla u^\top A \nabla u \, dx \geq \lambda \int_{\mathbb{R}^n} |\nabla u|^2 \, dx \quad \forall u \in H^1(\mathbb{R}^n, \mathbb{R}^m),$$

where $\nabla u^\top A \nabla v$ is defined as in (12).

The next result will provide higher regularity for the deformation u.

Theorem 3.6 (Higher regularity, adapted from [85, Th. 2.13, Rem. 2.16])**.** Let the set $U \subset \mathbb{R}^n$ be an open domain. Assume $A \in \mathbb{R}^{m^2 \times n^2}$ satisfies the Legendre-Hadamard condition (13). Furthermore, let $f \in H^\ell_{\text{loc}}(U, \mathbb{R}^m)$ for an integer $\ell \geq 0$. Suppose that $u \in H^1_{\text{loc}}(U, \mathbb{R}^m)$ is a solution of the weak formulation

$$\int_U \nabla u^\top A \nabla v \, dx = \langle f, v \rangle_{L^2(U, \mathbb{R}^m)} \quad \forall v \in H^1_0(U, \mathbb{R}^m).$$

Then $u \in H^{\ell+2}_{\text{loc}}(U, \mathbb{R}^m)$.

With the subsequent embedding theorem of Sobolev spaces into Hölder spaces, we will deduce continuous differentiability from the higher regularity of u.

Theorem 3.7 (Sobolev embedding theorem [58, 10.13])**.** Let the set $U \subset \mathbb{R}^n$ be open and bounded with Lipschitz boundary. Moreover, let $\ell \geq 1$ be an integer and let $1 \leq p < \infty$. In addition, let $m \geq 0$ be an integer and let $0 \leq \lambda \leq 1$. Then the following holds:

1. If $\ell - \frac{n}{p} = m + \lambda$ and $0 < \lambda < 1$, then the embedding

   $$\text{Id} : W^{\ell,p}(U) \to C^{m,\lambda}(\bar{U})$$

   exists and is continuous.

2. If $\ell - \frac{n}{p} > m + \lambda$, then the embedding

   $$\text{Id} : W^{\ell,p}(U) \to C^{m,\lambda}(\bar{U})$$

   exists and is continuous and compact.
   Here $C^{m,0}(\bar{U}) := C^m(\bar{U})$.

The previous embedding theorem is optimal in the sense that no embeddings of the types asserted exist, if the parameter values n, m, ℓ, p, λ do not satisfy the restrictions imposed in Th. 3.7; see [87, 4.40].

Lastly, we present the statements that will be used in Sec. 3.3.4 to prove that first-order regularization generally does not guarantee continuously differentiable deformations.

The utilized theorems all deal with second-order partial differential equations. For this reason, as in [86, Sec. 6.3.1], we assume in the following that L is an elliptic operator having the form

$$Lu := -\sum_{i,j=1}^{n} a_{ij}\partial_{x_i x_j} u + \sum_{i=1}^{n} b^i \partial_{x_i} u + cu$$

acting on functions $u : \bar{U} \to \mathbb{R}$, where $U \subset \mathbb{R}^n$ is open and bounded. We assume that the coefficient functions $a_{ij}, b^i, c : \bar{U} \to \mathbb{R}$ are continuous and the symmetry condition $a_{ij} = a_{ji}$ is fulfilled.
Moreover, we assume that L fulfills the uniform ellipticity condition [86, Sec. 6.1.1], i.e. there exists a constant $\theta > 0$ such that

$$\sum_{i,j=1}^{n} a_{ij}(x)\xi_i\xi_j \geq \theta|\xi|^2$$

for a.e. $x \in U$ and all $\xi \in \mathbb{R}^n$.

The first result we recapitulate in this context will be used to show that the deformation u is infinitely differentiable in the interior of the 3D volume Ω.

Theorem 3.8 (Infinite differentiability in the interior [86, Sec. 6.3.1, Th. 3])**.** Let $U \subset \mathbb{R}^n$ be open and bounded. Assume $f \in C^\infty(U)$. Suppose $u \in H^1(U)$ is a weak solution of the elliptic partial differential equation

$$Lu = f \quad \text{in } U.$$

Then

$$u \in C^\infty(U).$$

The following theorem will provide a statement about where the maximum of the deformation u lies in Ω.

Theorem 3.9 (Strong maximum principle [86, Sec. 6.4.2, Th. 4])**.** Let $U \subset \mathbb{R}^n$ be open, bounded and connected. Assume $u \in C^2(U) \cap C(\bar{U})$. For the coefficient function c in the operator L, suppose $c \geq 0$ in U.

1. If $Lu \leq 0$ in U and u attains a nonnegative maximum over $\bar{U}$ at an interior point, then u is constant within U.
2. Similarly, if $Lu \geq 0$ in U and u attains a nonpositive minimum over $\bar{U}$ at an interior point, then u is constant within U.

As the last step in the proof for first-order regularization, we will use Hopf's Lemma, which is stated next.

Theorem 3.10 (Hopf's lemma [86, Sec. 6.4.2])**.** Let $U \subset \mathbb{R}^n$ be open and bounded. Let $u \in C^2(U) \cap C^1(\bar{U})$. For the coefficient function c in the operator L, assume $c = 0$ in U. Suppose further

$$Lu \leq 0 \quad \text{in } U,$$

and there exists a point $x^0 \in \partial U$ such that

$$u(x^0) > u(x) \quad \forall x \in U.$$

Assume finally that U satisfies the interior ball condition at x^0; i.e. there exists an open ball $V \subset U$ with $x^0 \in \partial V$. Then

$$\frac{\partial u}{\partial \nu}(x^0) > 0,$$

where ν is the outer unit normal to V at x^0.

With the theorems presented in this section, we can state and prove the main results of this chapter.

3.3.2 Necessary Optimality Condition

The analysis of both the first-order and the second-order regularization is based on the following necessary optimality condition for a local minimizer of (11). Therefore, we consider this condition in advance. This necessary optimality condition is then used in the proofs in the subsequent two sections.

Theorem 3.11 (Necessary optimality condition)**.** Let $k \in \mathbb{N}$, let Ass. 3.1a) and Ass. 3.2 hold and let $u \in \mathbb{H}^k$ be a local minimizer of J defined by (11). Then

$$a[u, v] = -F[v] \quad \forall v \in \mathbb{H}^k \tag{14}$$

with $a[u, v]$ defined as in Ass. 3.2 and $F[v] = d_v D_{y_0}[u]$.

Proof. Since $d_v S[u] = a[u, v]$, the statement is a direct consequence of the necessary optimality condition

$$d_v J[u] = 0 \quad \forall v \in \mathbb{H}^k$$

for a minimizer $u \in \mathbb{H}^k$ of J defined by (11). □

Remark 3.12. For given F, the optimality condition (14) has a unique solution $u \in \mathbb{H}^k$. This follows from the Lax-Milgram theorem (Th. 3.4). Here, the continuity and the coercivity of the bilinear form a hold by Ass. 3.2. The linearity and boundedness of F follow from Ass. 3.1a) and (14).

3.3.3 Main Result for Second-Order Regularization

In this section, we present our first main result, which addresses Problem 1 by focusing on second-order regularization.
We show that second-order regularization guarantees continuous differentiability of the deformation in the considered 3D/2D registration setting. Later in Sec. 3.5, we will also extend this result to registration problems with dimensions other than 3D/2D.

Theorem 3.13 (Second-order regularization)**.** Assume $k = 2$. Let Ass. 3.1–3.3 hold and let $u \in \mathbb{H}^k$ be a local minimizer of the functional J defined by (11). Then $u \in C^1(\Omega, \mathbb{R}^3) \cap C(\bar{\Omega}, \mathbb{R}^3)$.

Proof. As $k = 2$, it holds $\mathbb{H}^k = H^2(\Omega, \mathbb{R}^3) \cap H_0^1(\Omega, \mathbb{R}^3)$. We show $u \in C^1(\Omega, \mathbb{R}^3)$ and $u \in C(\bar{\Omega}, \mathbb{R}^3)$ separately.

First, we prove $u \in C^1(\Omega, \mathbb{R}^3)$. For this, we show $u \in H^3_{\text{loc}}(\Omega, \mathbb{R}^3)$. Then the first part of the Sobolev embedding theorem (Th. 3.7) yields $u \in C^1(\Omega, \mathbb{R}^3)$.
The idea for the proof of $u \in H^3_{\text{loc}}(\Omega, \mathbb{R}^3)$ is to split the necessary condition (14), which is

$$\langle Bu, Bv \rangle_{L^2(\Omega,\mathbb{R}^3)} = -F[v] \quad \forall v \in \mathbb{H}^k,$$

into a system of equations

$$\begin{aligned} &\langle \tilde{z}, Bv \rangle_{L^2(\Omega,\mathbb{R}^3)} = -F[v] && \forall v \in \mathbb{H}^k, \\ &\langle Bz, \tilde{v} \rangle_{L^2(\Omega,\mathbb{R}^3)} = \langle \tilde{z}, \tilde{v} \rangle_{L^2(\Omega,\mathbb{R}^3)} && \forall \tilde{v} \in C_c^\infty(\Omega, \mathbb{R}^3) \quad \text{(i.e. } Bz = \tilde{z} \text{ a.e. in } \Omega) \end{aligned}$$

and convert the latter into weak formulations with first-order derivatives.
More precisely, we assume that $z, \tilde{z} \in H_0^1(\Omega, \mathbb{R}^3)$ solve the system

$$\int_\Omega \nabla \tilde{z}^\top A \nabla v \, dx = F[v] \qquad \forall v \in H_0^1(\Omega, \mathbb{R}^3), \tag{15}$$

$$\int_\Omega \nabla z^\top A \nabla \tilde{v} \, dx = -\langle \tilde{z}, \tilde{v} \rangle_{L^2(\Omega,\mathbb{R}^3)} \quad \forall \tilde{v} \in H_0^1(\Omega, \mathbb{R}^3). \tag{16}$$

Here, we use the notation introduced in (12), i.e. $\nabla \tilde{z}^\top A \nabla v = \sum_{\alpha,\beta,i,j=1}^3 A_{ij}^{\alpha\beta} \partial_{x_\beta} \tilde{z}^j \partial_{x_\alpha} v^i$ for $\tilde{z}, v \in H_0^1(\Omega, \mathbb{R}^3)$ and analogously for $z, \tilde{v} \in H_0^1(\Omega, \mathbb{R}^3)$.
We conduct the proof of $u \in H^3_{\text{loc}}(\Omega, \mathbb{R}^3)$ by showing the following:

1) The system given by (15) and (16) has unique solutions $z, \tilde{z} \in H_0^1(\Omega, \mathbb{R}^3)$,
2) $z \in H^3_{\text{loc}}(\Omega, \mathbb{R}^3)$, and
3) z fulfills the necessary optimality condition (14).

Since by Rem. 3.12 the condition (14) has only one solution, it must then hold $z = u$ and thus $u \in H^3_{\text{loc}}(\Omega, \mathbb{R}^3)$, so in conclusion $u \in C^1(\Omega, \mathbb{R}^3)$.

Proof of 1): We use the Lax-Milgram theorem (Th. 3.4) to show the existence of the unique solutions $z, \tilde{z} \in H_0^1(\Omega, \mathbb{R}^3)$. For this, we define a bilinear form by the left-hand sides of (15) and (16),

$$b : H_0^1(\Omega, \mathbb{R}^3) \times H_0^1(\Omega, \mathbb{R}^3) \to \mathbb{R}, \quad b[z, v] := \int_\Omega \nabla z^\top A \nabla v \, dx.$$

The bilinear form b is continuous, since

$$|b[z,v]| \leq c_1 \|z\|_{H^1(\Omega,\mathbb{R}^3)} \|v\|_{H^1(\Omega,\mathbb{R}^3)} \quad \forall z, v \in H_0^1(\Omega,\mathbb{R}^3)$$

with the constant $c_1 > 0$ depending only on A. In addition, by Th. 3.5 and the Poincaré inequality [58, 6.7] there exist constants $c_2, c_3 > 0$ independent of z such that

$$b[z,z] \geq c_2 \int_\Omega |\nabla z|^2 \, dx \geq c_3 \|z\|^2_{H^1(\Omega,\mathbb{R}^3)} \quad \forall z \in H_0^1(\Omega,\mathbb{R}^3).$$

Thus, b is also coercive. Note that Th. 3.5 requires that A fulfills the Legendre-Hadamard condition and the symmetry condition $A_{ij}^{\alpha\beta} = A_{ji}^{\beta\alpha}$, which is both provided by Ass. 3.3. In order to use the Lax-Milgram theorem, the functionals on the right-hand side of (15) and (16) must be bounded and linear. For (15) this applies due to Ass. 3.1b). Therefore, according to the Lax-Milgram theorem, there is a unique solution $\tilde{z} \in H_0^1(\Omega,\mathbb{R}^3)$ of (15). Consequently, the functional on the right-hand side of (16) is also bounded and linear. Thus, by the Lax-Milgram theorem, there is also a unique solution $z \in H_0^1(\Omega,\mathbb{R}^3)$ of (16).

Proof of 2): Since $\tilde{z} \in H_0^1(\Omega,\mathbb{R}^3)$, we can apply Th. 3.6 on (16). From this we directly get $z \in H^3_{\text{loc}}(\Omega,\mathbb{R}^3)$.

Proof of 3): To show that z fulfills the necessary condition (14), we use that by Ass. 3.3 the operator $B : \mathbb{H}^k \to L^2(\Omega,\mathbb{R}^3)$ in the bilinear form defining the regularizer is given by $Bu = A \cdot \nabla^2 u$, i.e.

$$(Bu)_i = \sum_{\alpha,\beta,j=1}^{3} \partial_{x_\alpha}(A_{ij}^{\alpha\beta} \partial_{x_\beta} u_j) \quad \forall i = 1,2,3.$$

First, note that (15) implies

$$\langle \tilde{z}, Bv \rangle_{L^2(\Omega,\mathbb{R}^3)} = -F[v] \quad \forall v \in \mathbb{H}^k. \tag{17}$$

In addition, with (16) and $z \in H^3_{\text{loc}}(\Omega,\mathbb{R}^3) \cap H_0^1(\Omega,\mathbb{R}^3)$, it holds

$$\langle Bz, \tilde{v} \rangle_{L^2(\Omega,\mathbb{R}^3)} = \langle \tilde{z}, \tilde{v} \rangle_{L^2(\Omega,\mathbb{R}^3)} \quad \forall \tilde{v} \in H_0^1(\Omega,\mathbb{R}^3).$$

Since the set $C_c^\infty(\Omega,\mathbb{R}^3)$ of smooth functions with compact support in Ω is dense in $H_0^1(\Omega,\mathbb{R}^3)$, it also holds

$$\langle Bz, \tilde{v} \rangle_{L^2(\Omega,\mathbb{R}^3)} = \langle \tilde{z}, \tilde{v} \rangle_{L^2(\Omega,\mathbb{R}^3)} \quad \forall \tilde{v} \in C_c^\infty(\Omega,\mathbb{R}^3).$$

The fundamental lemma of the calculus of variations [58, 4.22] implies $Bz = \tilde{z}$ a.e. in Ω. Inserting this into (17) yields

$$\langle Bz, Bv \rangle_{L^2(\Omega,\mathbb{R}^3)} = -F[v] \quad \forall v \in \mathbb{H}^k,$$

thus z fulfills the necessary condition (14).

It remains to show $u \in C(\bar{\Omega},\mathbb{R}^3)$. Since $u \in \mathbb{H}^k \subset H^2(\Omega,\mathbb{R}^3)$ this directly follows with the second part of the Sobolev embedding theorem (Th. 3.7), which yields $u \in C^{0,\lambda}(\bar{\Omega},\mathbb{R}^3)$ for $\lambda < \frac{1}{2}$ and thus in particular $u \in C(\bar{\Omega},\mathbb{R}^3)$.

Altogether we have shown that $u \in C^1(\Omega,\mathbb{R}^3) \cap C(\bar{\Omega},\mathbb{R}^3)$. □

Remark 3.14. The statement of Th. 3.13 still holds if we relax Ass. 3.3 and consider certain operators B with non-constant coefficient matrix A. However, typical regularizers in 3D/2D image registration have constant coefficients.

3.3.4 Main Result for First-Order Regularization

Next, we present our second main result, which addresses Problem 1 for first-order regularization. We show that first-order regularization generally does not guarantee continuous differentiability of the deformation. For this purpose, we provide a class of registration problems whose deformations satisfy $u \notin C^1(\Omega, \mathbb{R}^3) \cap C(\bar{\Omega}, \mathbb{R}^3)$. As we argue in Rem. 3.16, this means u is not continuously differentiable or that u is continuous in Ω but not in $\bar{\Omega}$. Both scenarios are undesirable for practical applications. An extension of the result to registration problems with dimensions other than 3D/2D will be presented in Sec. 3.5.

The class of registration problem considered here is volume-to-slice registration with common choices for the data fitting term and the first-order regularization, namely SSD and diffusion; see Sec. 2.2.1. As we will show in Sec. 3.4, Ass. 3.1a) and Ass. 3.2 hold in this case, so any local minimizer $u \in \mathbb{H}^k$ of (11) fulfills the necessary condition (14) with

$$d_v D_{y_0}[u] = \int_\Sigma \Big(T[y](x) - R(x)\Big) \left[\langle \nabla T[y](x), v(x)\rangle_{\mathbb{R}^3}\right] d\mathcal{H}^2(x),$$

$$d_v S[u] = \langle \nabla u, \nabla v\rangle_{L^2(\Omega,\mathbb{R}^3)} = \sum_{j=1}^{3} \int_\Omega \nabla u_j^\top \nabla v_j \, dx.$$

Without loss of generality, we suppose that the cuboid Ω is $(-1,1)^3$ and the 2D set Σ is $\{0\} \times (-1,1)^2$.

Theorem 3.15 (First-order regularization)**.** Let $k = 1$ and consider the setting of volume-to-slice registration introduced above.
Suppose that $M := \{x \in \Sigma : T[y_0](x) \neq R(x) \wedge \nabla T[y_0](x) \neq 0\}$ has positive 2D Hausdorff measure and let $u \in \mathbb{H}^k$ be a local minimizer of the functional J defined by (11). Then $u \notin C^1(\Omega, \mathbb{R}^3) \cap C(\bar{\Omega}, \mathbb{R}^3)$.

Proof. The underlying space of J is $\mathbb{H}^k = H_0^1(\Omega, \mathbb{R}^3)$. We show $u \notin C^1(\Omega, \mathbb{R}^3) \cap C(\bar{\Omega}, \mathbb{R}^3)$. To argue by contradiction, we assume $u \in C^1(\Omega, \mathbb{R}^3) \cap C(\bar{\Omega}, \mathbb{R}^3)$. With this u is differentiable and thus for each $x \in \Omega$ and opposite directions $\nu_1 = -\nu_2$ it holds

$$\frac{\partial u}{\partial \nu_1}(x) = -\frac{\partial u}{\partial \nu_2}(x).$$

We will show that there exists $x^0 \in \Sigma$ for which the equation is not fulfilled, which provides the contradiction.

For this, we first use the 2D set Σ to split $\Omega \setminus \Sigma$ in two connected domains Ω_1 and Ω_2 as depicted in Fig. 3.4. Due to $\Sigma \subset \partial\Omega_i$, with $i \in \{1,2\}$, for $v \in H_0^1(\Omega_i, \mathbb{R}^3)$ it holds $v|_\Sigma = 0$ a.e. and thus $d_v D_{y_0}[u] = 0$. For a function in $H_0^1(\Omega_i, \mathbb{R}^3)$, we consider the extension by zero, which yields a function in $H_0^1(\Omega, \mathbb{R}^3)$. Then, from the necessary condition (14) it

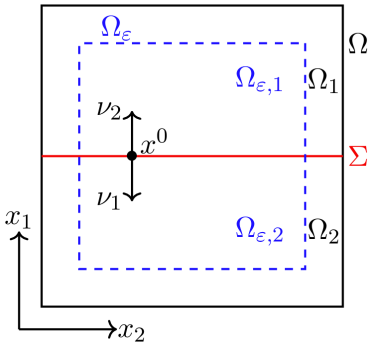


Figure 3.4: *In-plane visualization of the decomposition of the 3D set Ω used in the proof of Th. 3.15.*

follows

$$\langle \nabla u, \nabla v \rangle_{L^2(\Omega_i, \mathbb{R}^3)} = 0 \quad \forall v \in H_0^1(\Omega_i, \mathbb{R}^3).$$

Due to the interior regularity result in Th. 3.8 the above identity implies $u \in C^\infty(\Omega_i, \mathbb{R}^3)$. In addition, u is neither constant on Ω_1 nor on Ω_2. Otherwise, the Dirichlet boundary conditions together with the continuity of u in $\bar{\Omega}$ would result in $u = 0$ on Ω. Thus, with (14) it would follow $d_v D_{y_0}[u] = -\langle \nabla u, \nabla v \rangle_{L^2(\Omega, \mathbb{R}^3)} = 0$ for all $v \in H_0^1(\Omega, \mathbb{R}^3)$, which contradicts the assumption on M. Thus, the strong maximum principle stated in Th. 3.9 ensures that $u|_{\bar{\Omega}_i}$ does neither attain a nonnegative maximum nor a nonpositive minimum in the interior Ω_i.
Without loss of generality, we assume that the maximum of u is greater than zero. The alternative is that the minimum of u is smaller than zero. In this case, however, the remainder of the proof can be argued analogously.
If the maximum of u is greater than zero, the maximum of u is attained only on $\partial\Omega_i$. Because of the Dirichlet boundary conditions this implies that there is $x^0 \in \Sigma \setminus \partial\Omega$ with $u(x^0) > u(x)$ for all $x \in \Omega_1$ and for all $x \in \Omega_2$. We define a slightly smaller cuboid $\Omega_\varepsilon := \{x \in \Omega : \operatorname{dist}(x, \partial\Omega) > \varepsilon\}$ and set $\Omega_{\varepsilon,i} := \Omega_i \cap \Omega_\varepsilon$, where $\varepsilon > 0$ is chosen such that $x_0 \in \Omega_\varepsilon$ (such a choice is possible because of $x_0 \notin \partial\Omega$). Note that $u \in C^1(\bar{\Omega}_\varepsilon, \mathbb{R}^3)$. In conclusion, Hopf's Lemma (Th. 3.10) yields

$$\frac{\partial u}{\partial \nu_i}(x^0) > 0 \quad \text{for } i = 1, 2,$$

where ν_i is the outer unit normal to Ω_i at x^0. This contradicts $\frac{\partial u}{\partial \nu_1}(x^0) = -\frac{\partial u}{\partial \nu_2}(x^0)$, which holds because $u \in C^1(\Omega, \mathbb{R}^3)$ and $\nu_1 = -\nu_2$. □

The following remark argues that for practical applications Th. 3.15 provides a class of registration problems whose deformations are not continuously differentiable or not suitable in other ways.

Remark 3.16. First, the condition on M in Th. 3.15 is no restriction for practical registration problems. To comprehend this, we assume that the condition on M is not satisfied, i.e. M has 2D Hausdorff measure zero. In this case, $u = 0$ fulfills $d_v D_{y_0}[u] = 0$ for all $v \in H_0^1(\Omega, \mathbb{R}^3)$ and thus is a solution of the necessary condition (14). Therefore, already

the initial deformation y_0 is a solution to the registration problem.
Second, we consider the fact that Th. 3.15 only yields $u \notin C^1(\Omega, \mathbb{R}^3) \cap C(\bar{\Omega}, \mathbb{R}^3)$ instead of the stronger result $u \notin C^1(\Omega, \mathbb{R}^3)$. However, for $u \notin C^1(\Omega, \mathbb{R}^3) \cap C(\bar{\Omega}, \mathbb{R}^3)$ at least one of the following cases applies:

- If $u \in C(\bar{\Omega}, \mathbb{R}^3)$, then $u \notin C^1(\Omega, \mathbb{R}^3)$.
- If, on the other hand, $u \notin C(\bar{\Omega}, \mathbb{R}^3)$, then either $u \notin C(\Omega, \mathbb{R}^3)$ (thus $u \notin C^1(\Omega, \mathbb{R}^3)$) or u is continuous in Ω but not in $\bar{\Omega}$.

Both cases, $u \notin C^1(\Omega, \mathbb{R}^3)$ and $u \in C(\Omega, \mathbb{R}^3) \setminus C(\bar{\Omega}, \mathbb{R}^3)$, are undesirable for practical 3D/2D registration problems.

3.4 Suitable Data Fitting and Regularization Terms

In this section, we verify that the conditions formulated in Sec. 3.2, on which our main results are based, apply in practice. For this, we assess whether commonly used data fitting and regularization terms in 3D/2D registration fulfill Ass. 3.1–3.3. In Sec. 3.4.1 we focus on popular similarity measures, while in Sec. 3.4.2 we consider different first- and second-order regularizers.

3.4.1 Suitable Data Fitting Terms

First, we focus on Ass. 3.1, i.e. requiring the data fitting term D_{y_0} to be Gateaux differentiable with the derivative at fixed $u \in \mathbb{H}^k$ being in H^{-1}. Note that the latter part, which is Ass. 3.1b), is only used for $k = 2$.

We will verify Ass. 3.1 for the data fitting terms introduced in Sec. 2.2.1. However, in 3D/2D registration the 3D template image has to be projected to the 2D set Σ in order to compare it with the 2D image R. That means the data fitting term includes a projection operator, which we will now introduce.

We denote the projection by P, i.e. P is a linear operator with $P \circ P = P$. Furthermore, we suppose the following.

Assumption 3.4 (Continuity of P)**.** The projection P is continuous as an operator from $H^1(\Omega)$ to $L^2(\Sigma)$.

In particular, we will employ this condition to show that for commonly used data fitting terms Ass. 3.1b) holds, i.e. that their Gateaux derivative is bounded on $H_0^1(\Omega, \mathbb{R}^3)$.

Ass. 3.4 is satisfied, for instance, by the two usual projection operators discussed in the examples below.

Example 3.17 (Cine MRI imaging)**.** A Cine MRI slice image contains 2D data of a single slice. Thus, when aligning 2D Cine MRI images with 3D MRI images, volume-to-slice registration is performed; see e.g. Fig. 3.1. Note that this is the type of registration also investigated by Heldmann and Papenberg [50]. The corresponding projection operator is defined by

$$P[I](x) := \begin{cases} I(x), & \text{if } x \in \Sigma, \\ 0, & \text{else} \end{cases} \tag{18}$$

for a given image $I : \mathbb{R}^3 \to \mathbb{R}$. This projection operator satisfies Ass. 3.4 due to the trace theorem [58, A8.6]. Also, if the projection is defined by multiple of such slice operators, Ass. 3.4 is still fulfilled.
Note that the projection operator (18) cannot be defined sensibly for functions in $L^2(\Omega)$, see [86, Sec. 5.10], and thus $H^1(\Omega)$ is required in Ass. 3.4.

Example 3.18 (X-ray imaging)**.** Another common type of 2D images occurring in 3D/2D image registration are X-ray images. For example, registration of 3D CT and 2D X-ray images is used in [70].
In X-ray imaging, the attenuation of radiation is measured. The corresponding projection operator is defined by an integral over the line from the radiation source to the detector element. Without loss of generality, we consider here a simplified form of the projection operator underlying X-ray imaging. As in the proof of Th. 3.15, we suppose that the cuboid Ω is $(-1,1)^3$ and the 2D set Σ is $\{0\} \times (-1,1)^2$. We consider the projection P, which takes for an image $I : \mathbb{R}^3 \to \mathbb{R}$ the mean in the x_1-direction, i.e.

$$P[I](x) := \frac{1}{2} \int_{[-1,1]} I(s, x_2, x_3)\, ds \quad \text{for } x \in \Omega.$$

This operator satisfies Ass. 3.4, since for $I \in H^1(\Omega)$ it holds

$$\begin{aligned} \|P[I]\|^2_{L^2(\Sigma)} &= \int_{[-1,1]^2} \left| \frac{1}{2} \int_{[-1,1]} I(x)\, dx_1 \right|^2 dx_2\, dx_3 \leq c \int_{[-1,1]^2} \left(\int_{[-1,1]} |I(x)|^2\, dx_1 \right) dx_2\, dx_3 \\ &= c\|I\|^2_{L^2(\Omega)} \leq c\|I\|^2_{H^1(\Omega)} \end{aligned}$$

for a constant $c \in \mathbb{R}$. For the first inequality, we used the Cauchy-Schwarz inequality.

Based on Ass. 3.4, in the following we discuss three common similarity measures: SSD, NCC and NGF; see Sec. 2.2.1. We examine whether these fulfill Ass. 3.1.

Sum of Squared Differences (SSD)

In 3D/2D registration, the SSD similarity measure, see (2), and its Gateaux derivative are given by

$$\begin{aligned} D_{y_0}[u] &= \frac{1}{2} \int_\Sigma \Big(P[T[y]](x) - R(x) \Big)^2\, d\mathcal{H}^2(x), \qquad \text{respectively,} \\ d_v D_{y_0}[u] &= \int_\Sigma \Big(P[T[y]](x) - R(x) \Big) P\big[\langle \nabla T[y](\cdot), v(\cdot) \rangle_{\mathbb{R}^3} \big](x)\, d\mathcal{H}^2(x) \end{aligned}$$

for $u, v \in H^k(\Omega, \mathbb{R}^3)$ with $k \in \{1, 2\}$, where $y = y_0 + u$. The Gateaux derivative can be determined by a straightforward calculation; see e.g. [11, Sec. 7.1.1] for an analogous computation.

Thus, Ass. 3.1a) is satisfied.
We now show that Ass. 3.1b) holds for $k = 2$, i.e. that $F \in H^{-1}$ for $F[v] = d_v D_{y_0}[u]$ with fixed $u \in \mathbb{H}^2$. For this, we need to show that F is bounded on $H^1_0(\Omega, \mathbb{R}^3)$. Thus, we fix $v \in H^1_0(\Omega, \mathbb{R}^3)$ and show $|F[v]| \leq C\|v\|_{H^1(\Omega,\mathbb{R}^3)}$ for a constant $C \in \mathbb{R}$.
Since $T \in C^2(\Omega)$ with compact support in Ω, the absolute values of T and its first two

derivatives are bounded, i.e. there exists a constant $c_1 \in \mathbb{R}$ with

$$|T(x)| + \|\nabla T(x)\|_{\mathbb{R}^3} + \|\nabla^2 T(x)\|_{\mathbb{R}^{3\times 3}} \leq c_1 \quad \forall x \in \mathbb{R}^3.$$

Hence, using $R|_\Sigma \in L^2(\Sigma)$, the continuity of $P : H^1(\Omega) \to L^2(\Sigma)$, $y \in H^2(\Omega, \mathbb{R}^3)$ and Hölder's inequality [58, 3.18], we find

$$\begin{aligned}
|F[v]| &\leq \left(\|P[T[y]]\|_{L^2(\Sigma)} + \|R\|_{L^2(\Sigma)}\right) \|P[\langle \nabla T[y], v\rangle_{\mathbb{R}^3}]\|_{L^2(\Sigma)} \\
&\leq c_2 \| \langle \nabla T[y], v\rangle_{\mathbb{R}^3} \|_{H^1(\Omega)} \\
&\leq c_2 \left(\| \langle \nabla T[y], v\rangle_{\mathbb{R}^3} \|_{L^2(\Omega)} + \| \left\langle \nabla^2 T[y] \nabla y, v \right\rangle_{\mathbb{R}^3} \|_{L^2(\Omega)} + \|\nabla v \nabla T[y]\|_{L^2(\Omega,\mathbb{R}^3)} \right) \\
&\leq c_3 \left(\|v\|_{L^2(\Omega,\mathbb{R}^3)} + \|\nabla y\|_{L^4(\Omega,\mathbb{R}^9)} \|v\|_{L^4(\Omega,\mathbb{R}^3)} + \|\nabla v\|_{L^2(\Omega,\mathbb{R}^9)} \right) \\
&\leq c_4 \left(\|v\|_{L^2(\Omega,\mathbb{R}^3)} + \|v\|_{H^1(\Omega,\mathbb{R}^3)} + \|\nabla v\|_{L^2(\Omega,\mathbb{R}^9)} \right) \leq C \|v\|_{H^1(\Omega,\mathbb{R}^3)}
\end{aligned}$$

for constants $c_2, c_3, c_4, C \in \mathbb{R}$. In the penultimate inequality, we also used that $H^1(\Omega, \mathbb{R}^3)$ is continuously embedded in $L^4(\Omega, \mathbb{R}^3)$ due to $\Omega \subset \mathbb{R}^3$ and the embedding theorem [58, 10.9]. This proves Ass. 3.1b) for $k = 2$.

Normalized Cross-Correlation (NCC)

Another common similarity measure in image registration is NCC; see also (3). For 3D/2D registration, the data fitting term D_{y_0} in the objective (11) and its Gateaux derivative are given by

$$\begin{aligned}
D_{y_0}[u] &= 1 - \frac{\langle P[T[y]], R\rangle^2}{\langle P[T[y]], P[T[y]]\rangle \, \langle R, R\rangle}, \qquad \text{respectively,} \\
d_v D_{y_0}[u] &= 2 \left(\frac{\langle P[T[y]], R\rangle^2 \, \langle P[T[y]], P\left[\langle \nabla T[y](\cdot), v(\cdot)\rangle_{\mathbb{R}^3}\right]\rangle}{\langle P[T[y]], P[T[y]]\rangle^2 \, \langle R, R\rangle} \right. \\
&\qquad \left. - \frac{\langle P[T[y]], R\rangle \, \langle P\left[\langle \nabla T[y](\cdot), v(\cdot)\rangle_{\mathbb{R}^3}\right], R\rangle}{\langle P[T[y]], P[T[y]]\rangle \, \langle R, R\rangle} \right)
\end{aligned}$$

for $u, v \in H^k(\Omega, \mathbb{R}^3)$ with $k \in \{1, 2\}$ and $y = y_0 + u$. Here, we use the notation

$$\langle I_1, I_2 \rangle := \int_\Sigma I_1(x) I_2(x) \, d\mathcal{H}^2(x)$$

for images I_1, I_2 with $I_1|_\Sigma, I_2|_\Sigma \in L^2(\Sigma)$. In addition, we assume that the restrictions $P[T[y]]|_\Sigma$ and $R|_\Sigma$ are not identically zero, which is reasonable for practical registration problems. The Gateaux derivative can be directly calculated as in the SSD case.

Ass. 3.1a) applies, since F is linear for $F[v] = d_v D_{y_0}[u]$ with fixed $u \in \mathbb{H}^k$. In addition, the proof of $F \in H^{-1}$ for $k = 2$ runs completely analogously to the case of SSD as similarity measure and is therefore not presented here for the sake of brevity. Thus, Ass. 3.1b) holds.

Normalized Gradient Field (NGF)

Unfortunately, for the NGF similarity measure as defined by (4), Ass. 3.1b) cannot be verified in the same way as for SSD or NCC and Th. 3.13 may not hold. Thus, the NGF similarity measure is beyond the scope of this work.

The reason is that the data fitting term of NGF involves gradients of the images. Therefore, stronger regularity for R and $T[y]$ is necessary compared to the case of SSD or NCC; see also Sec. 2.2.1. For the same reason, we have to require higher regularity for the deformation u, i.e. $u \in H^2(\Omega, \mathbb{R}^3)$, in order to prove the Gateaux differentiability of $D_{y_0}[u]$. This is not necessarily reasonable when using first-order regularization. However, even for second-order regularizers we were unable to show that Ass. 3.1b) is fulfilled for NGF due to the additional derivative in the data fitting term. Stronger assumptions on the projection operator P than those formulated in Ass. 3.4 could solve this problem, but this would further restrict the set of possible projection operators. We will leave a detailed investigation of the conditions under which NGF satisfies Ass. 3.1 for future work.

3.4.2 Suitable Regularization Terms

Next, we focus on Ass. 3.2 and Ass. 3.3 and verify that these are fulfilled by common regularization terms. More precisely, we discuss elastic and diffusion regularization (as first-order regularizers) and curvature (as second-order regularizer); see Sec. 2.2.1 for an introduction of these regularizers.

First-Order Regularization

First, we consider $k = 1$ and thus $\mathbb{H}^k = H_0^1(\Omega, \mathbb{R}^3)$.

A common regularization scheme of first-order is elastic regularization, which is given by (6). In this case, the regularizer is defined as in Ass. 3.2 with the differential operator $B : \mathbb{H}^k \to L^2(\Omega, \mathbb{R}^{10})$ given by

$$B = \begin{bmatrix} \sqrt{\mu} I_3 \otimes \nabla \\ \sqrt{\lambda + \mu}\, \operatorname{div} \end{bmatrix} \tag{19}$$

where $\mu > 0$, $\lambda \geq -\mu$ are the so-called Lamé constants, I_3 is the identity matrix in $\mathbb{R}^3$ and $\otimes$ denotes the Kronecker product.
In order to show Ass. 3.2, it is required to prove that the bilinear form $a : \mathbb{H}^k \times \mathbb{H}^k \to \mathbb{R}$, $a[u, v] = \langle Bu, Bv \rangle_{L^2(\Omega, \mathbb{R}^m)}$ with B as in (19) is continuous and coercive. The continuity of a follows directly from the Cauchy-Schwarz inequality, as there is a constant $C > 0$ such that for all $u, v \in \mathbb{H}^k$ holds

$$a[u, v] = \langle Bu, Bv \rangle_{L^2(\Omega, \mathbb{R}^{10})} \leq \|Bu\|_{L^2(\Omega, \mathbb{R}^{10})} \|Bv\|_{L^2(\Omega, \mathbb{R}^{10})} \leq C \|u\|_{H^1(\Omega, \mathbb{R}^3)} \|v\|_{H^1(\Omega, \mathbb{R}^3)}.$$

The coercivity of a results from Poincaré's inequality [58, 6.7], which can be applied since Ω is open and bounded. More precisely, for $u \in \mathbb{H}^k$ holds

$$a[u, u] = \|Bu\|^2_{L^2(\Omega, \mathbb{R}^{10})} \geq \mu \|\nabla u\|^2_{L^2(\Omega, \mathbb{R}^9)} \geq c \|u\|^2_{H^1(\Omega, \mathbb{R}^3)}$$

for a constant $c > 0$. Thus, Ass. 3.2 is satisfied for elastic regularization.

Diffusion regularization, which is the first-order regularization investigated in [50] as well as in Th. 3.15, is a special case of elastic regularization. Here, the regularizer is defined as in (19) with the choice $\mu = -\lambda = 1$. Thus, Ass. 3.2 also holds for diffusion regularization.

Second-Order Regularization

Now we consider the case $k = 2$ and thus $\mathbb{H}^k = H^2(\Omega, \mathbb{R}^3) \cap H_0^1(\Omega, \mathbb{R}^3)$.

Curvature regularization is a popular second-order regularization, which is also considered by [50]. With the differential operator $B : \mathbb{H}^k \to L^2(\Omega, \mathbb{R}^3)$ given by

$$B = I_3 \otimes \Delta,$$

the curvature regularizer has the form stated in Ass. 3.2. Continuity and coercivity of the corresponding bilinear form $a : \mathbb{H}^k \times \mathbb{H}^k \to \mathbb{R}$ again follow from the Cauchy-Schwarz inequality and Poincaré's inequality, since there exist constants $C, c > 0$ such that for all $u, v \in \mathbb{H}^k$ holds

$$\begin{aligned} a[u, v] &= \langle Bu, Bv \rangle_{L^2(\Omega,\mathbb{R}^3)} \leq \|Bu\|_{L^2(\Omega,\mathbb{R}^3)} \|Bv\|_{L^2(\Omega,\mathbb{R}^3)} \leq C\|u\|_{H^2(\Omega,\mathbb{R}^3)} \|v\|_{H^2(\Omega,\mathbb{R}^3)}, \\ a[u, u] &= \|Bu\|^2_{L^2(\Omega,\mathbb{R}^3)} \geq c\|u\|^2_{H^2(\Omega,\mathbb{R}^3)}. \end{aligned}$$

Hence, Ass. 3.2 is fulfilled.
In order to show that Ass. 3.3 holds, we define the matrix $A \in \mathbb{R}^{3^2 \times 3^2}$ through

$$A_{ij}^{\alpha\beta} := \begin{cases} 1, & \text{if } \alpha = \beta \text{ and } i = j, \\ 0, & \text{else.} \end{cases}$$

Then, $Bu = A \cdot \nabla^2 u$ with $A \cdot \nabla^2 u$ as defined in Ass. 3.3. In Ex. 3.3 we already showed that A satisfies the Legendre-Hadamard condition (13). In addition, A fulfills the symmetry condition $A_{ij}^{\alpha\beta} = A_{ji}^{\beta\alpha}$. Thus, Ass. 3.3 holds.
Consequently, Th. 3.13 is applicable and we obtain that curvature regularization yields continuously differentiable deformations in 3D/2D registration.

3.5 Generalizations of the Main Results

So far we have considered 3D/2D image registration. More precisely, we addressed the question of which regularization order guarantees continuous differentiability of the deformation in variational 3D/2D registration. Yet, many arguments can be extended to registration problems of other dimensions. In this section, we briefly discuss possible generalizations.
We will show that the previous results of this chapter also apply in the case of 3D/1D and 2D/1D registration, while already specific first-order regularizers are appropriate in 3D/3D and 2D/2D registration, and higher order regularizers may be necessary in registration settings with higher dimension.

To this end, we replace in the setting introduced in Sec. 3.2 the space $\mathbb{R}^3$ by $\mathbb{R}^n$ with $n \in \mathbb{N}$ and assume that Σ is of dimension $m \in \mathbb{N}$, with $m \leq n$. In particular, this adjustment also applies to Ass. 3.1–3.3.
Note that the cases $n = m = 3$ and $n = m = 2$ correspond to the usual cases of registration, which are 3D/3D and 2D/2D registration. Therefore, statements regarding these cases are especially relevant for practical applications. There are also some applications based on other values of n and m. 4D/4D registration, for example, occurs when a time component is added to 3D images. However, the time component is usually treated differently than the location components, so that 4D/4D registration does not fit into the

framework covered in this chapter [88, 89]. Nevertheless, for the sake of completeness, we will consider $n, m \in \mathbb{N}$ in general in the following. More precisely, we will take another look at the two main results presented in Sec. 3.3 under this adapted setting.

First, we reconsider Th. 3.13, which states that specific second-order regularizers guarantee continuously differentiable deformations in 3D/2D registration. We show that this statement holds equally when $n \leq 3$, so that the following theorem covers the cases of 3D/3D, 3D/2D, 3D/1D, 2D/2D and 2D/1D registration. For higher dimensions, i.e. $n > 3$, transferring the statement of Th. 3.13 becomes much more complicated, as we will discuss below.

Theorem 3.19 (Extension of Th. 3.13)**.** Assume $k = 2$. Let Ass. 3.1–3.3 hold and let $u \in \mathbb{H}^k$ be a local minimizer of the functional J defined by (11). If $n \leq 3$, then $u \in C^1(\Omega, \mathbb{R}^n) \cap C(\bar{\Omega}, \mathbb{R}^n)$.

Proof. The proof of Th. 3.13 for 3D/2D registration can be transferred directly to this more general setting, since all statements used in that proof remain valid for $n \leq 3$. This applies in particular to the validity of the embeddings

$$H^3_{\text{loc}}(\Omega, \mathbb{R}^n) \to C^1(\Omega, \mathbb{R}^n) \quad \text{and} \quad H^2(\Omega, \mathbb{R}^n) \to C(\bar{\Omega}, \mathbb{R}^n),$$

on which the proof of Th. 3.13 is primarily based. □

However, if $n > 3$, the proof of Th. 3.13 cannot be directly extended. In this case, the space $H^3_{\text{loc}}(\Omega, \mathbb{R}^n)$ is not continuously embedded in $C^1(\Omega, \mathbb{R}^n)$, since the Sobolev embedding theorem (Th. 3.7) is optimal. Thus, for $n > 3$, the proof of Th. 3.13 fails and the theorem may not apply.
Higher regularization may therefore be necessary for $n > 3$ in order to ensure continuously differentiable deformations. However, higher-order regularization also leads to higher derivatives in the necessary condition (14). Thus, we have to deal with a higher-order equation, which requires further investigation. Consequently, the question of which regularization order guarantees continuous differentiability of the deformation for the case $n > 3$ is subject for future work.

Next, we focus on an extension of Th. 3.15, which is the second main result of Sec. 3.3. In this theorem, we provided an example with first-order regularization that yields deformations $u \notin C^1(\Omega, \mathbb{R}^3) \cap C(\bar{\Omega}, \mathbb{R}^3)$. In contrast to Th. 3.19, the following theorem holds for all registration settings with $m < n$, i.e. in particular, that the statement is not limited to the case $n \leq 3$.

Theorem 3.20 (Extension of Th. 3.15)**.** If $m < n$, then a registration problem with first-order regularization that yields deformations $u \notin C^1(\Omega, \mathbb{R}^n) \cap C(\bar{\Omega}, \mathbb{R}^n)$ can be formulated analogously to Th. 3.15.

Proof. The example in Th. 3.15 is directly transferable. Then, $u \notin C^1(\Omega, \mathbb{R}^n) \cap C(\bar{\Omega}, \mathbb{R}^n)$ can be shown by contradiction, as in the proof of Th. 3.15.
The first step of this proof was to split Ω into two connected sets using Σ. If $m = n - 1$, this is possible in the same way. If $m < n - 1$, we extend Σ to a set $\tilde{\Sigma}$, which contains Σ and is of dimension $n - 1$. This higher dimensional space $\tilde{\Sigma}$ can then be used to split Ω

into two connected domains.
The subsequent argumentation in the proof of Th. 3.15 remains valid for $m \leq n-1$, since the statements on which the proof is based hold for $\mathbb{R}^n$, not just for $\mathbb{R}^3$. □

Note that Th. 3.19 covers the most common registration settings, which are 3D/3D and 2D/2D registration. Thus, specific second-order regularizers guarantee continuously differentiable deformations in these cases. However, Th. 3.20 does not cover 3D/3D and 2D/2D registration. This raises the question of whether first-order regularization is already sufficient to guarantee continuously differentiable deformations in these cases.
We show that under further conditions first-order regularizers are indeed sufficient. Afterwards, we discuss that these conditions are in general fulfilled for 3D/3D and 2D/2D registration cases.

Theorem 3.21. Assume $k = 1$ and let Ass. 3.1a) and Ass. 3.2 hold. Furthermore, suppose that for any $u \in \mathbb{H}^k$ there exists $f \in H^1(\Omega, \mathbb{R}^n)$ such that the Gateaux derivative $F[v] = d_v D_{y_0}[u]$ is given by

$$F[v] = \langle f, v \rangle_{L^2(\Omega,\mathbb{R}^n)} \quad \forall v \in \mathbb{H}^k. \tag{20}$$

Let $u \in \mathbb{H}^k$ be a local minimizer of the functional (11). If $n \leq 3$, then $u \in C^1(\Omega, \mathbb{R}^n)$.

Proof. The necessary optimality condition (14) yields

$$\langle Bu, Bv \rangle_{L^2(\Omega,\mathbb{R}^\ell)} = -\langle f, v \rangle_{L^2(\Omega,\mathbb{R}^n)} \quad \forall v \in \mathbb{H}^k = H_0^1(\Omega, \mathbb{R}^n).$$

Since $f \in H^1(\Omega, \mathbb{R}^n)$, the regularity result [86, Sec. 6.3, Th. 2] yields $u \in H^3_{\text{loc}}(\Omega, \mathbb{R}^n)$. Thus, by the first part of the Sobolev embedding theorem (Th. 3.7) holds $u \in C^1(\Omega, \mathbb{R}^n)$. Note that the latter again requires $n \leq 3$. □

The statement and the proof of Th. 3.21 show similarities to the result for second-order regularization in Th. 3.13. However, in the latter case it also holds $u \in C(\bar{\Omega}, \mathbb{R}^n)$, which is not provided by Th. 3.21. The reason for this is that the proof of $u \in C(\bar{\Omega}, \mathbb{R}^n)$ is based on the embedding $H^2(\Omega, \mathbb{R}^n) \to C(\bar{\Omega}, \mathbb{R}^n)$ and $u \in H^2(\Omega, \mathbb{R}^n)$ does not necessarily hold if first-order regularization is used. Therefore, $u \in H^2(\Omega, \mathbb{R}^n)$ might not be satisfied in the setting with first-order regularization, which is considered in Th. 3.21.

We now comment on the condition (20) for the Gateaux derivative of the data fitting term. In particular, we show that for the 3D/2D registration example previously considered in Th. 3.15 the condition is not fulfilled, while it holds for usual 3D/3D and 2D/2D registration problems.

Remark 3.22. The data fitting term used in the 3D/2D registration example in Th. 3.15 does not fulfill the condition (20). Essentially, the reason for this is that this data fitting term, which is defined by an integral over the 2D set Σ, cannot be represented with an integral over Ω as in (20). We refrain from presenting a detailed proof here. Nevertheless, since the condition (20) is not satisfied, Th. 3.21 is not applicable. This is consistent with the fact that the deformations resulting from registration as in Th. 3.15 are not continuously differentiable.
In contrast, commonly used data fitting terms in 3D/3D and 2D/2D registration fulfill

the condition (20) and thus first-order regularization is already sufficient here to guarantee continuously differentiable deformations. Exemplary, we consider the distance measures SSD and NCC, which were also examined in Sec. 3.4 for the case of 3D/2D registration. In these cases, the Gateaux derivatives of the data fitting terms are given by

$$
\begin{aligned}
d_v D_{y_0}^{\mathrm{SSD}}[u] &= \langle (T[y] - R), \langle \nabla T[y](\cdot), v(\cdot) \rangle_{\mathbb{R}^d} \rangle, \\
d_v D_{y_0}^{\mathrm{NCC}}[u] &= 2 \left(\frac{\langle T[y], R \rangle^2 \, \langle T[y], \langle \nabla T[y](\cdot), v(\cdot) \rangle_{\mathbb{R}^d} \rangle}{\langle T[y], T[y] \rangle^2 \, \langle R, R \rangle} - \frac{\langle T[y], R \rangle \, \langle \langle \nabla T[y](\cdot), v(\cdot) \rangle_{\mathbb{R}^d}, R \rangle}{\langle T[y], T[y] \rangle \, \langle R, R \rangle} \right)
\end{aligned}
$$

for $u, v \in H^1(\Omega, \mathbb{R}^d)$ with $d = 2, 3$ and $y = y_0 + u$. Here, $\langle \cdot, \cdot \rangle$ denotes the standard scalar product on $L^2(\Omega)$. Under appropriate assumptions on the images T and R, there exists for each $u \in \mathbb{H}^k$ a function $f \in H^1(\Omega, \mathbb{R}^n)$, such that the condition (20) is satisfied.

Summarizing, in this section we showed that specific first-order regularizers are sufficient in 3D/3D and 2D/2D registration to ensure continuously differentiable deformations. For 3D/2D, 3D/1D and 2D/1D registration, specific second-order regularizers are required. Higher order regularizers, which are not common in applications, may be necessary for continuous differentiability in registration settings with higher dimension.

3.6 Conclusion

In the previous sections, we have analyzed the impact of the regularization in variational 3D/2D registration. More precisely, we have dealt with the question of which regularizers guarantee continuously differentiable deformations; see Problem 1. In order to answer this question, we built on the numerical investigations of Heldmann and Papenberg [50].

Our main contributions in this chapter are the following: We proved that specific second-order regularizers guarantee continuously differentiable deformations in 3D/2D registrations, while first-order regularizers generally do not. In addition, we have verified that the assumptions on which our proofs are based apply for commonly used data fitting and regularization terms. Furthermore, we discussed possible extensions of the statements on second- and first-order regularization in 3D/2D registration to registration problems of other dimensions.

However, the investigation on the impact of the regularization in variational 3D/2D registration can still be expanded. One aspect is that there are exceptions in which our working assumptions are not satisfied. An example for a data fitting term that is not covered by our investigation is the NGF distance measure; see the discussion in Sec. 3.4. An example for a second-order regularizer that is not covered is the divergence-curl regularization; see e.g. [59]. An inspection of such cases is subject for future work.
Further open questions arise when other properties are required for the deformation y. An example of this is the question of how to extend our results to situations where smoother deformations are sought, such as $y \in C^2(\Omega, \mathbb{R}^3)$. Or, what changes are necessary in our investigation if we demand continuous differentiability of the deformation in only one particular direction. These properties can be reasonable in certain applications. In volume-to-slice registration, for example, it can be useful to consider deformations that are only continuously differentiable in the direction perpendicular to the 2D slice. In this way, the space of suitable deformations is increased, while still ensuring that information is transferred from the 2D slice into the 3D volume. These questions are also subject for future investigation.

CHAPTER 4

A New Image Registration Method Based on Residual Neural Networks

In this chapter, a new registration method, called *residual neural network-based registration* (RNR), is presented. RNR allows the computation of multiple deformations describing the motion of an object at successive time points. In order to provide deformations that build on each other in a meaningful way, data from these time points can be input into RNR.
At the end of this chapter, we will highlight the power of RNR by enhancing the modeling of respiratory surface electromyography (sEMG) signals. Respiratory sEMG modeling is an important field, as it is utilized, for example, to improve mechanical ventilation of patients [5]. In order to model respiratory sEMG signals, the breathing movement itself must first be modeled. We will use RNR to create such a breathing model based on 3D and 2D MRI data; see Fig. 4.1.

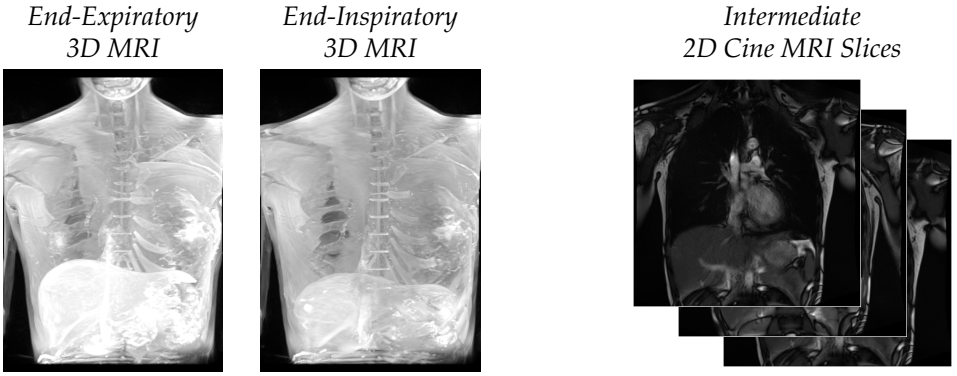


Figure 4.1: *Part of the data used to model the breathing movement with RNR: end-expiratory and end-inspiratory 3D MRI images and 2D Cine MRI slice images; see Sec. 4.7 for details.*

Mathematically, the registration task considered in this chapter is given as follows: Let $T \in \text{Img}(\Omega)$ and $R_1, ..., R_N \in \text{Img}(\Omega)$ be given images with an image domain $\Omega \subset \mathbb{R}^d$, $d \in \{2, 3\}$. We assume that the images $R_1, ..., R_N$ correspond to $N \in \mathbb{N}$ successive time points. Then the goal is to solve the following problem.

Problem 2. Find deformations $\phi_1, ..., \phi_N : \mathbb{R}^d \to \mathbb{R}^d$ such that

1. $T \circ \phi_n \approx R_n$, i.e. $T \circ \phi_n$ is similar to R, for all $n = 1, ..., N$,
2. $\phi_1, ..., \phi_N$ build on each other in a physically meaningful way,
3. $\phi_1, ..., \phi_N$ are diffeomorphisms.

Note that this task is formulated for a general dimension $d \in \{2, 3\}$, i.e. both 2D and 3D registration problems are addressed. A schematic visualization of Problem 2 for the 3D case is given in Fig. 4.2. For the breathing model at the end of this chapter we will expand the setting slightly further to perform a 3D registration which takes 2D data from intermediate time points into account.

We now briefly comment on each of the required conditions in Problem 2.
The condition $T \circ \phi_n \approx R_n$ assures that at each time point the deformed image $T \circ \phi_n$ is aligned with the corresponding image R_n. In particular, this means that there is one template image T and possibly several reference images $R_1, ..., R_N$.
With regard to the second condition, it should be noted that it is not possible to define physical meaningfulness in general, instead this property depends on the application. However, regardless of the application we can interpret $R_1, ..., R_N$ as discrete data of a time-dependent motion, and thus it is desirable that the deformations $\phi_1, ..., \phi_N$ represent this movement.
The last condition requires that the deformations are diffeomorphic, i.e. continuously differentiable and bijective with differentiable inverse. The motivation to restrict the space of suitable deformations to the class of diffeomorphisms is that these are invertible and in addition preserve properties in the deformed image such as connectedness of image regions or smoothness of curves in the image. These are desired characteristics in many applications [29].

Our approach for solving Problem 2 is based on an implicit neural representation (INR), i.e. we represent the deformations by a neural network; see Sec. 1.1.1 for an introduction

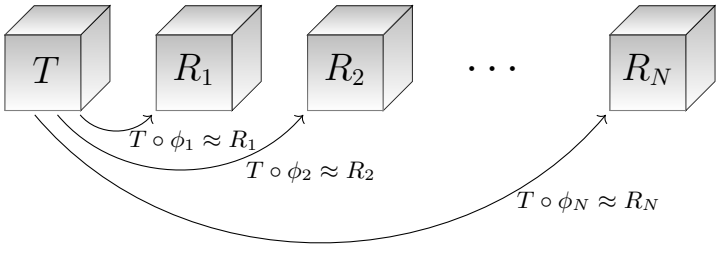


Figure 4.2: *Visualization of the registration problem handled by RNR. Here, the goal is to compute deformations ϕ_n such that the deformed template image $T \circ \phi_n$ is aligned with the corresponding reference image R_n, $n = 1, ..., N$. In addition, the deformations should build on each other in a meaningful way and be diffeomorphic; see Problem 2.*

to INRs. The network architecture we propose for this purpose is based on large deformation diffeomorphic metric mapping, a popular approach for diffeomorphic image registration; see e.g. [29, 90, 91].

In the following sections, we will present and validate the new registration method RNR in detail. On the one hand, we will show that the method is based on a solid theoretical foundation that guarantees, for example, the existence of solutions. One the other hand, we will also examine the method in various experiments, including its application in the field of sEMG modeling.

The structure of this chapter is as follows: First, in Sec. 4.1 we introduce the methods of image registration on which RNR is based. In particular, large deformation diffeomorphic metric mapping [29] and residual neural networks [92, 93] as well as the connection between these two popular approaches are presented. Also, a specific registration method [41] that combines these approaches and serves as a basis for our proposed method is summarized. Afterwards, in Sec. 4.2 we introduce RNR, our new registration method for solving Problem 2. In order to demonstrate that this registration method is well-founded, important properties of the method are examined in Sec. 4.3 and Sec. 4.4. While in Sec. 4.3 we formulate an appropriate regularization to guarantee existence of solutions, in Sec. 4.4 we specify conditions under which deformations resulting from RNR are diffeomorphic. Thereafter, in Sec. 4.5 we describe details on the implementation of RNR. A comprehensive validation of the method is then presented in Sec. 4.6. Here we provide several experimental results that give insight into what the deformations produced by RNR look like. Furthermore, RNR is compared with selected state-of-the-art registration approaches. In Sec. 4.7, we stress the applicability of our new registration method by generating a breathing model. Finally, in Sec. 4.8 we summarize the most important findings of this chapter and provide ideas for future research using RNR.

Sec. 4.2 and Sec. 4.7 in this chapter build on the author's publication [4], in which the author conceived and developed the ideas, performed the implementation and wrote the majority of the manuscript. The investigations presented in Sec. 4.3–4.6 have not been published prior to this work.

4.1 Related Work

In this section, we introduce image registration approaches and neural network architectures on which RNR is based.

First, in Sec. 4.1.1 a brief introduction to *large deformation diffeomorphic metric mapping*, or LDDMM for short, is given. LDDMM is a common approach for diffeomorphic registration. The method yields a time-dependent deformation, which allows access to deformations at intermediate time points. Therefore, LDDMM is a promising approach with regard to Problem 2. In Sec. 4.1.2, we describe *residual neural networks* (also referred to as a residual networks or ResNets) and their connection to diffeomorphic image registration. We will use ResNets later in our registration method to approximate deformations obtained by LDDMM. This way, we save computational effort compared to classical LDDMM methods. Lastly, in Sec. 4.1.3 we present the registration approach of Ben Amor et al. [41], which also utilizes ResNets to compute deformations. Our RNR method is an adaptation of this approach as we will explain later in Sec. 4.2.

4.1.1 Large Deformation Diffeomorphic Metric Mapping

One of the most popular approaches in diffeomorphic image registration is LDDMM, which is also the foundation of RNR. The LDDMM framework comprises a class of registration methods; see e.g. [29] and references therein. LDDMM was an innovation in image registration and builds on works from the 1990s [94, 95, 96]; see also [90, 91] for an introduction to LDDMM.

The idea of LDDMM methods is to generate a deformation by integrating a time-varying velocity field, such that the template and the reference image are connected by shortest paths. Since shortest paths are intersection-free, this approach guarantees a diffeomorphic deformation; see e.g. [90].

More precisely, in LDDMM an artificial time $t \in [0,1]$ is introduced and a time- and space-dependent velocity field $v : [0,1] \times \mathbb{R}^d \to \mathbb{R}^d$ is assumed. Based on this, a time-varying deformation $\phi : [0,1] \times \mathbb{R}^d \to \mathbb{R}^d$ is defined through the flow equation [96]

$$\begin{aligned} \partial_t \phi(t,x) &= v(t,\phi(t,x)) && \forall t \in [0,1], x \in \mathbb{R}^d, \\ \phi(0,x) &= x && \forall x \in \mathbb{R}^d. \end{aligned} \tag{21}$$

The final deformation $y : \mathbb{R}^d \to \mathbb{R}^d$ is given at $t = 1$, thus $y = \phi(1,\cdot)$. Fig. 4.3 visualizes the LDDMM setting in one dimension, i.e. $d = 1$.

Arguillere et al. [97] showed that this approach yields a diffeomorphic deformation with respect to the location; see also [94, 95]. In particular, they proved that a solution ϕ of (21) exists, is unique and $\phi(t,\cdot)$ is a diffeomorphism for each $t \in [0,1]$, if v fulfills the following conditions:

1. $v(t,\cdot) \in C_c^1(\mathbb{R}^d, \mathbb{R}^d)$ for all $t \in [0,1]$, and
2. $\int_0^1 \|v(t,\cdot)\|_{C_c^1(\mathbb{R}^d,\mathbb{R}^d)}\, dt := \int_0^1 \sup_{|\alpha|\le 1, x\in\mathbb{R}^d} \left|\frac{\partial^{\alpha_1+\cdots+\alpha_d}}{\partial x_1^{\alpha_1}\ldots\partial x_d^{\alpha_d}} v(t,x)\right| dt < \infty.$

Characterizing and thus name-giving for LDDMM is the property that the length of the shortest paths connecting the template and reference image defines a metric on the space

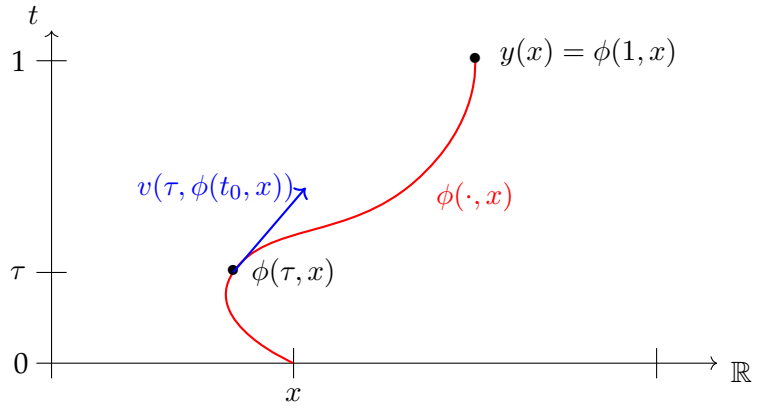


Figure 4.3: *Visualization of the LDDMM setting in one dimension. The deformation ϕ solving (21) yields for each $x \in \mathbb{R}$ a path $\phi(\cdot, x)$ (red) from the initial position x to the final position $y(x) = \phi(1,x)$. For each time point $\tau \in [0,1]$ the velocity vector $v(\tau, \phi(\tau,x))$ (blue) indicates the direction of the path in $\phi(\tau,x)$.*

of images [98]. This is an important characteristic as it allows to assess the similarity of various images.

The original LDDMM algorithm was proposed by Beg et al. [90] in 2005. In [90], a variational formulation is minimized in order to solve (21), which is regularized by the cost of the deformation, defined as the integral of the squared norm of the time-varying velocity field from which it arises.

Adaptations of Large Deformation Diffeomorphic Metric Mapping

One limitation of LDDMM algorithms is their computational complexity, which is caused by the time-dependence of the vector field v. In order to overcome this restriction several extensions to the original LDDMM algorithm were proposed; see e.g. [99, 100].

A popular and efficient adaptation of LDDMM is the concept of *stationary velocity fields* (SVFs), which was introduced by Arsigny et al. [101]. The idea of the SVF framework is similar to the classical LDDMM approach replacing the time-varying velocity field by a vector field $v : \mathbb{R}^d \to \mathbb{R}^d$, which is constant in time. The solution of the flow equation (21) is usually approximated using the scaling and squaring method for transformations, which allows for fast integration of stationary differential equations; see e.g. [101] for details. A serious disadvantage of SVFs is that they can only parameterize a subgroup of the deformations represented using LDDMM [29, 102]. Nevertheless, for many applications the SVF approach yields satisfactory registration results [31, 39, 40]. Together with the advantage of computational efficiency, this makes the concept of SVFs widely used alongside classical LDDMM [31, 101, 102].

There are multiple other extensions of the classical LDDMM approach, developed for different purposes. For instance, the symmetric normalization method (SyN) proposed by Avants et al. [103] formulates a symmetric image registration problem, in order to improve the symmetry between the forward and backward path of the diffeomorphism. More details about SyN as well as an overview about further LDDMM variants can be found in [29].

An alternative approach to these methods is to approximate the solution of the flow equation (21) using a ResNet, which we outline in the next section.

4.1.2 Residual Neural Network-Based Registration

Before introducing ResNets, we briefly summarize the general terminology of deep learning that is used in the following.

Introduction to Deep Learning

In recent years, deep learning has had a major impact on various fields of science [104]. Accordingly, it has also been established in image registration; see e.g. [8, 22].
The following introduction to the main deep learning notations used in this thesis is based on [104, 105]. In [105], deep learning is discussed from the perspective of applied mathematics, while in [104], it is introduced from the perspective of medical image processing.

A *neural network* is a function, which maps an input vector to an output vector using a combination of linear and nonlinear operations. More precisely, we define a neural

network with $L \in \mathbb{N}$ *layers* as a mapping $F : \mathbb{R}^{n_0} \to \mathbb{R}^{n_L}$ given by a concatenation

$$F := f_L \circ ... \circ f_2 \circ f_1,$$

where each layer $f_\ell : \mathbb{R}^{n_{\ell-1}} \to \mathbb{R}^{n_\ell}$, $\ell = 1, ..., L$, can be described as

$$f_\ell(x) = \sigma_\ell(W_\ell x + b_\ell).$$

Here, $W_\ell \in \mathbb{R}^{n_\ell \times n_{\ell-1}}$ is a matrix and $b_\ell \in \mathbb{R}^{n_\ell}$ is bias vector, both of which contain *weights* that determine layer ℓ. The mapping σ_ℓ is called *activation function* and is generally applied point-wise on each element of the vector. A schematic representation of an exemplary neural network is visualized in Fig. 4.4.
The weights in W_ℓ and b_ℓ are *learned/trained* by minimizing a certain *loss function*. In image registration, this loss function often contains a data fitting and a regularization term, such as the functional given by (1). The activation functions are chosen prior to the optimization process. Examples of commonly used activation functions include the sine as well as the Rectified Linear Unit (ReLU), which is given by

$$\mathrm{ReLU}(x) := \max\{0, x\}.$$

For an overview of activation functions in deep learning, we refer e.g. to [106].
The training of the network weights is usually performed using gradient descent-based methods, such as ADAM [107]. The step size is called *learning rate*. An *epoch* contains one or multiple optimization steps.
A neural network is called a *deep neural network* if it has more than two layers [108]. A special class of neural networks is given by *convolutional neural networks (CNNs)*. As the name suggest, these networks are characterized by layers that perform convolutions, which results in sparse and highly structured networks; see e.g. [105] for more details.

In image registration, neural networks are either used to predict or to represent a deformation; see Sec. 1.1.1. A particular type of network architecture that is useful for diffeomorphic registration are ResNets. An introduction to ResNets and their usability for diffeomorphic registration is presented next.

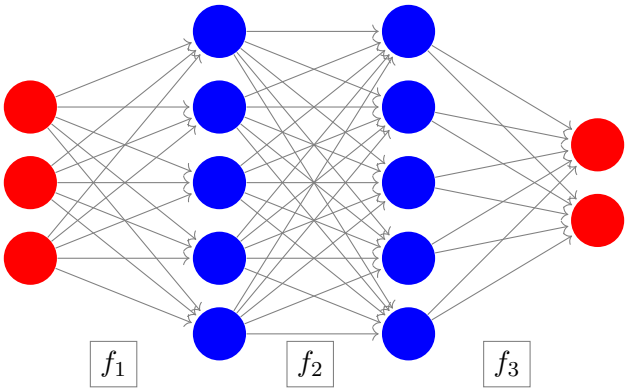


Figure 4.4: *Schematic representation of a neural network* $F : \mathbb{R}^3 \to \mathbb{R}^2$ *consisting of three fully-connected layers:* $F = f_3 \circ f_2 \circ f_1$ *with layers* $f_1 : \mathbb{R}^3 \to \mathbb{R}^5$, $f_2 : \mathbb{R}^5 \to \mathbb{R}^5$, $f_3 : \mathbb{R}^5 \to \mathbb{R}^2$.

Residual Neural Networks in Diffeomorphic Registration

ResNets [92, 93] have become state-of-the-art deep learning architectures used for a variety of problems, including image registration; see e.g. [28, 41]. ResNets can be regarded as numerical schemes for solving differential equations such as (21); see e.g. [109, 110]. This is the motivation to include ResNets in the registration approach RNR.

A ResNet consists of multiple consecutive residual units. Each unit contains an additive *residual function* $F_n : \mathbb{R}^m \to \mathbb{R}^m$ that is a neural network. More precisely, given an input vector $z_{n-1} \in \mathbb{R}^m$, the output of the n-th residual unit is given by

$$z_n := z_{n-1} + F_n(z_{n-1}). \tag{22}$$

Note that due to $z_{n-1} + F_n(z_{n-1}) = (\mathrm{Id} + F_n)(z_{n-1})$, the input of each unit is connected with its output via an identity mapping, which is referred to as *residual connection*. He et al. [93] stated that the residual mappings F_n are easier to learn than the parameters of deep neural networks without residual connections as introduced above. To understand this, one can imagine the extreme case: If an identity mapping were optimal, it would be easier to push the residual functions to zero than to parameterize an identity mapping by a stack of nonlinear layers.

The main motivation to utilize ResNets for diffeomorphic image registration is their connection to differential equations, which has been studied widely; see e.g. [109, 110]. In order to comprehend how to relate the ResNet relation (22) with a differential equation, we take a look at the flow equation (21) as an example. We approximate the solution ϕ of (21) using a forward Euler discretization [111] in time with $N \in \mathbb{N}$ time steps and step size $1/N$. Thus, we define transformations $\phi_0, ..., \phi_N : \mathbb{R}^d \to \mathbb{R}^d$ by

$$\begin{aligned} \phi_n(x) &:= \phi_{n-1}(x) + 1/N \cdot v(t_{n-1}, \phi_{n-1}(x)) && \forall x \in \mathbb{R}^d, n = 1, ..., N, \\ \phi_0(x) &:= x && \forall x \in \mathbb{R}^d, \end{aligned} \tag{23}$$

where $t_n := \frac{n}{N}$ for $n = 0, ..., N$. This means that ϕ_n is an approximation of $\phi(t_n, \cdot)$. This numerical scheme for solving the differential equation (21) relies upon the same recursive relationship as the ResNet relation (22). Setting $z_n = \phi_n(x)$ for $x \in \mathbb{R}^d$ and $F_n(z) = 1/N \cdot v(t_{n-1}, z)$ the connection of (22) and (23) is obvious. Note that using this relationship the residual functions F_n of a ResNet can be interpreted as an approximation to the velocity field v at time t_{n-1}.

Recently, several image registration procedures using ResNets have been developed. One procedure that uses an LDDMM-motivated ResNet architecture for the representation of the deformation was introduced by Zhang et al. [112]. The main difference to our new registration method presented in Sec. 4.2 is that in [112] only one residual function is used, which takes the time variable as an additional input. Huang et al. [113] use the idea of ResNets as discretizations of the LDDMM approach to align two or multiple time series. There are also approaches in which not the deformation but the velocity field on which the deformation is based is represented by a ResNet [28, 114]. For example, in [28] first an initial velocity field v_0 is computed using a CNN, then a sequence of velocity fields is calculated using the ResNet relation (22) and finally the deformation is yielded by a sum of the velocity fields. In this procedure the residual functions are no approximation of the velocity field, but of the derivative of the velocity field. Another recent approach that builds on the presented connection between the ResNet relation (22) and the flow equation (21) arising in LDDMM, was proposed by Ben Amor et al. [41] in

2022. Since our new registration method RNR is based on [41], we take a closer look on this approach in the following section.

4.1.3 The Registration Approach of Ben Amor et al.

In this section, we review the ResNet-based registration approach proposed by Ben Amor et al. [41], which is the basis for our new registration approach RNR introduced in Sec. 4.2.

In [41], a point cloud registration method in order to align 3D shapes is proposed. This means that the method takes two point clouds $q_R, q_T \subset \mathbb{R}^3$ as input and aims to find a 3D deformation y that maps these point clouds to each other. Here, it is assumed that the point clouds q_R, q_T have the same number of elements, but it is not known which points in R and T belong together. In [41], the deformation is parameterized by an INR based on (23).

More precisely, transformations $\phi_0^A, ..., \phi_N^A : \mathbb{R}^3 \to \mathbb{R}^3$ are calculated by

$$\begin{aligned} \phi_n^A(x) &:= \phi_{n-1}^A(x) + F_n^A(\phi_{n-1}^A(x)) && \forall x \in \mathbb{R}^3, n = 1, ..., N, \\ \phi_0^A(x) &:= x, && \forall x \in \mathbb{R}^3 \end{aligned} \tag{24}$$

and the final transformation $y : \mathbb{R}^3 \to \mathbb{R}^3$ is defined by the last point in the time-discretization $y = \phi_N$. For the residual functions F_n^A a deep neural network is utilized, which consists of three fully connected layers, where in the first layer a ReLU activation function is applied. We denote the network width by q, which is set to a few hundred by Ben Amor et al. [41]. The first weight layer extends the dimension of the input vector from 3 to $q \gg 3$, while the second keeps the dimensionality and the third layer again reduces the dimensionality to get a velocity vector in $\mathbb{R}^3$. In more detail, each residual function F_n^A is given by

$$F_n^A(x) := W_3^n(W_2^n(\text{ReLU}(W_1^n x + b_1^n)) + b_2^n) \quad \forall x \in \mathbb{R}^3 \tag{25}$$

with weight matrices $W_1^n \in \mathbb{R}^{q\times 3}, W_2^n \in \mathbb{R}^{q\times q}, W_3^n \in \mathbb{R}^{3\times q}$ and bias vectors $b_1^n, b_2^n \in \mathbb{R}^q$. The ReLU activation function is applied point-wise, i.e. $\text{ReLU}(x) = (\max\{0, x_j\})_{j=1,...,q}$ for $x \in \mathbb{R}^q$. Note that in the last linear layer no bias vector is added. A schematic visualization of the resulting network architecture representing the deformation is given in the upper part of Fig. 4.5.

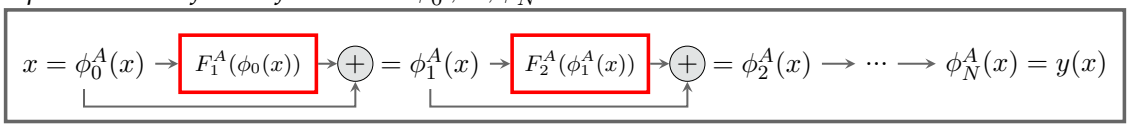
To find a deformation that maps the point clouds q_R and q_T onto each other, the network parameters in the residual functions are trained by minimizing an energy functional of the form (1). Therein, for the data fitting term a point cloud similarity measure is used, which pushes the set of deformed template points $y(q_T)$ closer to the reference set q_R. In addition, the deformation is kept close to the identity by penalizing the sum over all kinetic energies of the system at all time points, i.e.

$$S^A[y] := \sum_{x\in q_T} \sum_{n=1}^{N} \|F_n^A(x)\|_2^2,$$

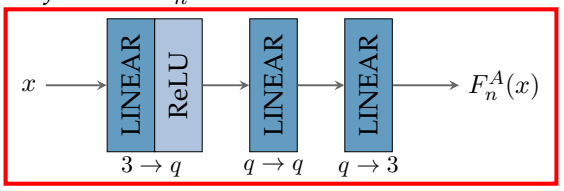
where $F_1^A, ..., F_N^A$ are the residual functions corresponding to the final transformation y. For further details we refer to [41].

Network Architecture in Ben Amor et al. [41]

Representation of the deformations $\phi_0^A, ..., \phi_N^A : \mathbb{R}^3 \to \mathbb{R}^3$*:*

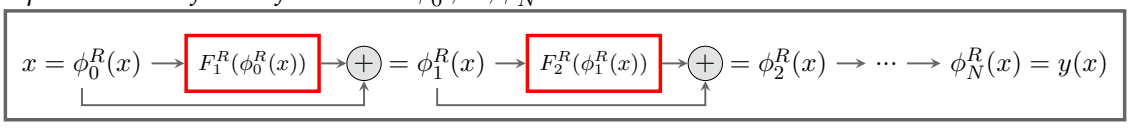


Representation of a residual function $F_n^A : \mathbb{R}^3 \to \mathbb{R}^3$*:*



Network Architecture in RNR

Representation of the deformations $\phi_0^R, ..., \phi_N^R : \mathbb{R}^d \to \mathbb{R}^d$*:*



Representation of a residual function $F_n^R : \mathbb{R}^d \to \mathbb{R}^d$*:*

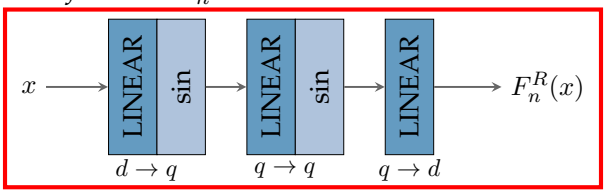


Figure 4.5: *Network architectures in Ben Amor et al. [41] and RNR. For each:* ***Top:*** *INR used to parameterize the deformations.* ***Bottom:*** *representation of the residual functions, which are the main component of the parameterizations for the deformations. See Sec. 4.1.3 and Sec. 4.2 for details.*

4.2 RNR: A New Residual Neural Network-Based Registration Approach

In this section, we present a new registration method called residual neural network-based registration, RNR for short, to solve Problem 2. RNR allows the inclusion of data corresponding to time points between the initial and final state as additional reference images. In addition, RNR yields diffeomorphic deformations under certain conditions, which we present later in Sec. 4.4.

Our new method builds on the point cloud registration approach of Ben Amor et al. [41] introduced in Sec. 4.1.2. The motivation for this is twofold: First, the approach in [41] yields not only a final deformation but also deformations corresponding to intermediate time points, which we will use to include the intermediate data. Second, the approach in [41] already provides diffeomorphic deformations under certain conditions.

As in [41], the central idea of our method is to parameterize the deformations using an INR, which is based on the forward Euler discretization (23) of the LDDMM-related flow equation (21).

We will now introduce the RNR method in detail. The differences between RNR and the approach in [41] are discussed systematically at the end of this section.

In RNR the deformations $\phi_0^R, ..., \phi_N^R : \mathbb{R}^d \to \mathbb{R}^d$, with $d \in \{2, 3\}$, are calculated sequentially, analogous to (24), i.e. we set

$$\begin{aligned} \phi_n^R(x) &:= \phi_{n-1}^R(x) + F_n^R(\phi_{n-1}^R(x)) && \forall x \in \mathbb{R}^d, n = 1, ..., N, \\ \phi_0^R(x) &:= x, && \forall x \in \mathbb{R}^d. \end{aligned} \tag{26}$$

The final deformation is again defined by the last time point, i.e. $y = \phi_N^R$. Note that the deformations defined by (26) can be evaluated directly at any position x without the need for interpolation, i.e. RNR is a grid-free registration approach.
For the residual functions F_n^R occurring in (26), we utilize deep neural networks, which differ from the networks proposed by Ben Amor et al. [41]: The networks for the residual functions in RNR consist of three fully connected layers, where in the first and second layer the sine is applied point-wise as activation function. We denote the network width again by q, which we will chose such that $q \gg d$ in the experiments presented later, for example $q = 100$. Then, the first weight layer of each network extends the dimension of the input vector from d to q, while the second keeps the dimensionality and the third layer reduces the dimensionality again to get an output vector in $\mathbb{R}^d$. Thus, each residual function F_n^R is of the form

$$F_n^R(x) := W_3^n \sin(W_2^n \sin(W_1^n x + b_1^n) + b_2^n) + b_3^n \quad \forall x \in \mathbb{R}^d \tag{27}$$

with weight matrices $W_1^n \in \mathbb{R}^{q \times d}, W_2^n \in \mathbb{R}^{q \times q}, W_3^n \in \mathbb{R}^{d \times q}$ and bias vectors $b_1^n, b_2^n \in \mathbb{R}^q$, $b_3^n \in \mathbb{R}^d$. Note that with (27) translations are the only affine transformations that can be parameterized. However, in our experiments we will show that RNR is nevertheless capable of generating appropriate solutions for several registration problems that arise from affine deformations; see Sec. 4.6.3. A schematic visualization of the network architecture representing the deformations resulting from RNR is given in the lower part of Fig. 4.5.

The total number M of parameters in the network is given by the numbers of elements in the weight matrices and bias vectors, i.e. the total number is

$$M := N \cdot (qd + qq + dq + q + q + d) = N \cdot (q^2 + 2(d + 1)q + d). \tag{28}$$

In order to solve Problem 2, the network parameters in the parameterization of the residual functions are trained by minimizing a loss function of the form (1). To reiterate, we minimize

$$J[y] = D[y] + \alpha_S S[y] + \alpha_P P[y], \tag{29}$$

where D is a data fitting term, S a regularizer, P a penalty term and $\alpha_S > 0$, $\alpha_P \geq 0$. Since each of the components D, S and P is discussed in detail in the following sections, only a brief explanation of how the components are selected is given here.
The data fitting term D ensures that for each time point the deformed template image matches the reference image corresponding to the respective time point. This term therefore takes care of the first condition in Problem 2. Appropriate choices of the data fitting term depend on the application; see Sec. 4.5 for the choices used in the numerical results later. The regularizer S guarantees existence of solutions for the registration problem;

see Sec. 4.3. The penalty term P controls the smoothness of the deformations resulting from RNR and ensures that these are diffeomorphic; see Sec. 4.4. Thus, the penalty takes care of the third condition in Problem 2.

The RNR method is strongly connected to the registration approach of Ben Amor et al. [41] that was presented in Sec. 4.1.2. Basically, in both approaches the same LDDMM-motivated ResNet architecture is utilized to parameterize the deformations. The following three aspects mainly distinguish the approach presented here from [41]:

1. Instead of point cloud registration, we perform an intensity-based registration. Thus, the network parameters are optimized not only for a point cloud but for the entire image domain. This leads to a much larger number of points used during training compared to [41].
2. We incorporate information about the motion by integrating intermediate data into our approach.
3. We utilize a different network architecture for the residual functions F_n. In both approaches such a network consists of three linear layers, but instead of one ReLU activation function applied in the first layer, we use sine as activation function and include it twice.

While the first two modifications are clearly motivated by the application on hand, which was briefly introduced in the beginning of this chapter and will be further discussed in Sec. 4.7, the advantage of the last modification may not be obvious. The reason for adapting the network architecture for the residual functions is that we want to allow large nonlinear deformations. First of all, we would argue that in the approach of Ben Amor et al. [41] the three layers are actually only two, since two linear operations are applied consecutively and thus they can be summarized as one linear operation. By applying the activation function twice in fact three linear layers exist and consequently the degree of freedom in our approach is higher than in [41] when using the otherwise same setting, i.e. the same network width q and so forth. The replacement of the ReLU activation function by the sine activation functions is motivated by the work of Sitzmann et al. [115], which showed that INRs with sine activation functions outperform ReLU-based networks in multiple imaging related experiments. In Sec. 4.6 we will compare our adapted parameterization of the residual functions with the original parameterization of Ben Amor et al. [41] numerically. Our experiments show that with RNR indeed larger deformations are generated.

4.3 Ensuring Existence of Solutions

In this and the following section, we demonstrate that RNR has a sound theoretical basis. We first show that RNR requires explicit regularization in order to guarantee solutions of Problem 2. Furthermore, we proof that penalizing the network weights within RNR is a sufficient choice for the regularizer.

4.3.1 Necessity of Explicit Regularization

Ensuring existence of solutions is a fundamental property of image registration methods; see e.g. [15]. In this section, we show that for RNR this is not fulfilled without using an explicit regularization term. More precisely, we prove that the restriction of the solution

space, which is given by the parameterization of the deformations resulting from the network architecture, is not sufficient to guarantee existence of solutions.

With (26) and (27) the natural choice for the set of admissible transformations for RNR is given by

$$\mathcal{A} := \{y \in L^2(\Omega, \mathbb{R}^d) : y = \varphi_N \circ ... \circ \varphi_1 \text{ with } \varphi_1, ..., \varphi_N \in \mathcal{A}_1\},$$

where $\mathcal{A}_1$ is the set of transformations attained using $N = 1$, i.e.

$$\begin{aligned}\mathcal{A}_1 := \{y \in L^2(\Omega, \mathbb{R}^d) : y(x) = x + W_3 \sin(W_2 \sin(W_1 x + b_1) + b_2) + b_3,\\ W_1 \in \mathbb{R}^{q\times d}, W_2 \in \mathbb{R}^{q\times q}, W_3 \in \mathbb{R}^{d\times q}, b_1, b_2 \in \mathbb{R}^q, b_3 \in \mathbb{R}^d\}.\end{aligned}$$

The following theorem shows that existence of solutions is not guaranteed without additional regularization. More precisely, the theorem provides a registration problem that is a counterexample, i.e. that has no solution in $\mathcal{A}$.

Theorem 4.1. Let $T, R \in \text{Img}(\Omega)$ be two images such that $T(2x) = R(x)$ for all $x \in \Omega$, where $\Omega \subset \mathbb{R}^d$ is a non-empty image domain. Then the registration problem

$$\min_{y\in\mathcal{A}} D(T \circ y, R), \tag{30}$$

where D is the MSE similarity measure, see (2), has no solution.

Proof. Without loss of generality, we assume $\Omega = [-1, 1]$, i.e. in particular that the dimension of the image domain is $d = 1$. An extension to $d > 1$ is straightforward.

Obviously $y^* : \Omega \to \mathbb{R}, y(x) = 2x$ minimizes $D(T \circ y, R)$, but $y^* \notin \mathcal{A}$ due to the sinusoidal part in the residual functions of transformations in $\mathcal{A}$. However, there exists a sequence $(y_k)_{k\in\mathbb{N}} \subset \mathcal{A}$ such that $y_k \to y^*$ for $k \to \infty$. More precisely, for $k \in \mathbb{N}$ define $y_k : \Omega \to \mathbb{R}$ by

$$y_k(x) := x + W_3^k \sin(W_2^k \sin(W_1^k x + b_1^k) + b_2^k) + b_3^k$$

with

$$\begin{aligned}&W_1^k := \begin{pmatrix}\frac{1}{k} & 0 & \cdots & 0\end{pmatrix}^\top \in \mathbb{R}^q, \quad W_2^k := I \in \mathbb{R}^{q\times q}, \quad W_3^k := \begin{pmatrix}k & 0 & \cdots & 0\end{pmatrix} \in \mathbb{R}^{1\times q},\\ &b_1^k := 0 \in \mathbb{R}^q, \qquad\qquad b_2^k := 0 \in \mathbb{R}^q, \qquad b_3^k := 0 \in \mathbb{R}.\end{aligned}$$

Then it holds with L'Hôpital's rule

$$\begin{aligned}\lim_{k\to\infty} y_k(x) &= \lim_{k\to\infty} x + k \sin\left(\sin\left(\frac{1}{k}x\right)\right) = x + \lim_{k\to\infty} \frac{\sin\left(\sin\left(\frac{1}{k}x\right)\right)}{\frac{1}{k}}\\ &= x + \lim_{k\to\infty} \frac{\cos\left(\sin\left(\frac{1}{k}x\right)\right)\cos\left(\frac{1}{k}x\right)\left(-\frac{1}{k^2}x\right)}{-\frac{1}{k^2}} = x + x = 2x = y^*(x).\end{aligned}$$

Thus, $(y_k)_{k\in\mathbb{N}} \subset \mathcal{A}$ is a minimizing sequence, but its limit y^* is not in the set of admissible transformations $\mathcal{A}$. This means that the registration problem (30) has no solution. □

4.3.2 Existence of Solutions with Appropriate Regularization

Since without explicit regularization the RNR method does not guarantee existence of solutions, we introduce an appropriate regularization term in this section. The idea behind the regularizer presented in the following is to ensure boundedness of the deformation by ensuring that all network parameters within RNR used to parameterize the deformation remain bounded.

The boundedness of the bias vectors is already implicit in the registration problem: First, note that since the vectors b_1^n, b_2^n are in the argument of a sine function, it is no restriction for the represented deformation to assume that each component of the vectors lies in the interval $[-\pi, \pi]$. To argue that the vectors b_3^n can also be assumed to be bounded, recall that images are compactly supported in a domain Ω and thus vanish outside Ω. Therefore, for reasonable deformations each component of the vectors b_3^n is bounded by the diameter of the domain, which is given by $\operatorname{diam}(\Omega) = \sup_{x_1,x_2\in\Omega} \|x_1 - x_2\|_2$. For larger deformations the similarity measure is constant and thus no further reduction of the distance functional can be achieved. Summarizing, without loss of generality, we restrict the set of admissible transformations to

$$\mathcal{A}_b := \{y \in \mathcal{A} : \|b_1^n\|_\infty, \|b_2^n\|_\infty, \|b_3^n\|_\infty \leq C_b \text{ for } n = 1, ..., N\},$$

where $C_b \in \mathbb{R}$ is a constant just depending on the domain Ω.

In order to ensure that the weight matrices in the parameterization of the deformation remain bounded, we define the regularizer $S : \mathcal{A}_b \to \mathbb{R}$ by

$$\begin{aligned} S[y] &:= \frac{1}{N} \sum_{n=1}^{N} S_n[\phi_n], \quad \text{where} \\ S_n[\phi_n] &:= \frac{1}{L} \left(\|W_1^n\|_F^2 + \|W_2^n\|_F^2 + \|W_3^n\|_F^2 \right). \end{aligned} \tag{31}$$

Here, the vector fields $\phi_1, ..., \phi_N$ are the successive deformations corresponding to the final transformation y, the matrices W_1^n, W_2^n, W_3^n are the weight matrices in the parameterization of ϕ_n and L is the number of parameters within the matrices for one time step, i.e. $L = q^2 + 2qd$.

Remark 4.2. Using (31), we regularize the deformation by penalizing the corresponding network weights. Note, however, that there is no one-to-one correspondence between the deformation y and the vector fields ϕ_n as well as between the vector fields ϕ_n and the weight matrices W_1^n, W_2^n, W_3^n. The weight matrices uniquely define the vector fields ϕ_n, and the vector fields ϕ_n uniquely define the final transformation y. But the reverse directions do not hold.
In the implementation of RNR this inconsistency is no problem, since the optimization is directly performed on the weights of the network. Thus, the formulation of S as a function of the weight matrices would be correct. Nevertheless we decided to use the notation as in (31) in order to emphasize that we actually regularize the final deformation y by including S in the minimization problem.

The following theorem yields that S is indeed a regularizer, i.e. S ensures existence of solutions in $\mathcal{A}_b$. Note that the assumptions we make for the distance functional are not a restriction in practical registration problems.

Theorem 4.3. Let $D : \mathcal{A}_b \to \mathbb{R}_{\geq 0}$ be a distance functional. We assume that D is continuous and $D[\mathrm{Id}] < \infty$. Then the registration problem

$$\min_{y \in \mathcal{A}_b} D[y] + S[y] \tag{32}$$

has at least one solution.

Proof. To simplify the notation, we combine all network parameters in the parameterization of a deformation $y \in \mathcal{A}_b$ in one vector $W \in \mathbb{R}^M$, where M is the number of network parameters given by (28). Let $\mathbb{R}_b^M \subset \mathbb{R}^M$ be the space of vectors corresponding to deformations in $\mathcal{A}_b$ and identify the distance and regularization functional $D, S : \mathcal{A}_b \to \mathbb{R}$ with functions $D, S : \mathbb{R}_b^M \to \mathbb{R}$ by identifying a deformation with its parameter vector. Instead of the registration problem (32) we consider the equivalent problem

$$\min_{W \in \mathbb{R}_b^M} J[W], \quad \text{where } J[W] = D[W] + S[W]. \tag{33}$$

In order to proof existence of a solution of (33), we construct a minimizing sequence $(W_k)_{k \in \mathbb{N}} \subset \mathbb{R}_b^M$, which has a convergent subsequence, show that its limit W^* is a minimizer of J and that W^* actually belongs to $\mathbb{R}_b^M$.

For this, let $(W_k)_{k \in \mathbb{N}} \subset \mathbb{R}_b^M$ be a minimizing sequence of J, i.e.

$$\inf_{W \in \mathbb{R}_b^M} J[W] = \lim_{k \to \infty} J[W_k].$$

We denote by $(y_k)_{k \in \mathbb{N}}$ the corresponding sequence in $\mathcal{A}_b$. Without loss of generality, we assume $J[0] > J[W_k]$ for all $k \in \mathbb{N}$. It follows

$$J[0] > J[W_k] \geq S[W_k]. \tag{34}$$

Note that we used in the second inequality that the data fitting term reaches only non-negative values; see Sec. 2.2.1. The left-hand side of (34) is bounded due to the boundedness assumption on the distance functional and since the regularization part becomes zero if all parameters in the network are zero. Thus, also the right-hand side of (34) remains bounded and, since S is given by (31), all parameters within the weight matrices in the parameterization of y_k are bounded. Since $(y_k)_{k \in \mathbb{N}} \subset \mathcal{A}_b$ also the bias vectors are bounded and together $(W_k)_{k \in \mathbb{N}}$ is bounded.
With the Bolzano-Weierstrass theorem there exists a convergent subsequence. We denote this subsequence again by $(W_k)_{k \in \mathbb{N}}$ (with corresponding sequence of deformations $(y_k)_{k \in \mathbb{N}} \subset \mathcal{A}_b$) and its limit by $W^* \in \mathbb{R}^M$ (with corresponding deformation $y^* \in \mathcal{A}$).

Note that in a first step we only yield $W^* \in \mathbb{R}^M$, i.e. $y^* \in \mathcal{A}$. However, since the definition of $\mathcal{A}_b$ limits the bias vectors in the parameterization of y_k by a constant C_b, this is also the case for the limit value y^*. Thus, we get $y^* \in \mathcal{A}_b$ or equivalently $W^* \in \mathbb{R}_b^M$.

Furthermore, from the continuity assumption of the distance functional and because $S[W_k] \to S[W^*]$ for $W_k \to W^*$, it follows

$$\inf_{W \in \mathbb{R}_b^M} J[W] = \lim_{k \to \infty} J[W_k] = J[W^*]$$

and hence W^* is a minimizer of J. □

Although S is a sufficient regularizer for deformations computed with RNR, a disadvantage of this regularizer is that a physical interpretation of S is missing. Of course, keeping the network weights close to zero ensures that the final deformation is close to the identity. But, in contrast to other physical inspired regularizers such as elastic [65] or hyperelastic regularization [84], there is no physical foundation for the regularizer S yet. For this reason, we will utilize an additional interpretable penalty term in the numerical examples presented in Sec. 4.6 and Sec. 4.7. This penalty term is based on the results of the following section.

4.4 Ensuring Diffeomorphic Solutions

In this section, we investigate whether the deformations determined with RNR are diffeomorphic, i.e. continuously differentiable and bijective with differentiable inverse. Thus, we verify whether the third condition of Problem 2 is satisfied. To this end, in Sec. 4.4.1 we first consider the analytical time-continuous solution of the differential equation underlying RNR. We show that the analytical solution is diffeomorphic for each fixed time point. However, discretization is required to compute the solution numerically. Therefore, in Sec. 4.4.2 we present conditions, which guarantee that also the time-discrete solution calculated with RNR is diffeomorphic for each time point. These conditions will also motivate the choice of the penalty term used in the experiments in Sec. 4.6 and Sec. 4.7.

4.4.1 Time-Continuous Setting

We consider a differential equation whose Euler discretization yields the deformations that result from RNR. We show that this differential equation has a unique analytical solution which is diffeomorphic for each fixed point in time.

For this purpose, we define a velocity function $v^R : [0,1] \times \mathbb{R}^d \to \mathbb{R}^d$ based on the residual functions F_n^R introduced in Sec. 4.2 and proof that the corresponding flow equation (21) is uniquely solvable and yields a diffeomorphism for each fixed time point. Arguillere et al. [97] have shown that this is the case for velocity fields that fulfill certain conditions; see Sec. 4.1.1. However, in order to give insight into the proof in [97], we will not only verify these conditions for the specific velocity field v^R. Instead, we present the proof for the velocity field v^R in detail, based on the same main components as in [97]: the Picard-Lindelöf theorem and the Cauchy-Lipschitz theorem.

For this reason, we now state modified versions of both theorems. While the Picard-Lindelöf theorem guarantees existence and uniqueness of a solution to the flow equation (21), the Cauchy-Lipschitz theorem contains a statement about the differentiability of the solution. The latter is a consequence of the Picard-Lindelöf theorem and the Lemma of Gronwall [116, Lemma 2.7].

Theorem 4.4 (Picard-Lindelöf theorem, adapted from [116, Th. 2.2])**.** Assume that $v : [0,1] \times \mathbb{R}^d \to \mathbb{R}^d$ is continuous in the first argument and Lipschitz continuous in the second with a Lipschitz constant that is independent from the first argument. Then, the flow equation (21) has a unique solution $\phi : [0,1] \times \mathbb{R}^d \to \mathbb{R}^d$.

Theorem 4.5 (Cauchy-Lipschitz theorem, adapted from [116, Th. 2.8, Th. 2.10])**.** Assume $v \in C^k([0,1] \times \mathbb{R}^d, \mathbb{R}^d)$ for an integer $k \geq 1$ and let v be Lipschitz continuous in the second argument with a Lipschitz constant independent from the first argument. Furthermore, let $\phi : [0,1] \times \mathbb{R}^d \to \mathbb{R}^d$ be the unique solution of (21), which exists due to Th. 4.4. Then, for each $t \in [0,1]$ the mapping $\phi(t,\cdot)$ is a C^k-diffeomorphism, i.e. $\phi(t,\cdot)$ is bijective, and $\phi(t,\cdot)$ and its inverse are k-times continuously differentiable.

Now we have the tools together to define the vector field v^R and proof that the corresponding differential equation (21) has a unique analytical solution which is diffeomorphic for each fixed point in time.

Theorem 4.6. Let $F_n^R : \mathbb{R}^d \to \mathbb{R}^d$ be the residual functions given by (27) with fixed weight matrices W_j^n and bias vectors b_j^n. Let the vector field $v^R : [0,1] \times \mathbb{R}^d \to \mathbb{R}^d$ be defined through piecewise linear interpolation of the functions F_n^R in the time variable, i.e.

$$v^R(t,x) = N\left(F_n^R(x) + (t - t_{n-1})\frac{F_{n+1}^R(x) - F_n^R(x)}{t_n - t_{n-1}}\right) \quad \text{for } t \in [t_{n-1}, t_n], x \in \mathbb{R}^d$$

with $t_n = \frac{n}{N}$ for $n = 0, ..., N$.
With this choice for the velocity field, the flow equation (21) has a unique solution $\phi : [0,1] \times \mathbb{R}^d \to \mathbb{R}^d$ and $\phi(t,\cdot)$ is diffeomorphic for each $t \in [0,1]$.

Proof. First, we use the Picard-Lindelöf theorem (Th. 4.4) to prove existence and uniqueness of the solution ϕ of (21). For this, we have to show that v^R is continuous in the first argument and Lipschitz continuous in the second with a Lipschitz constant, which is independent from the first argument. The former applies by the definition of v^R as a piecewise linear interpolation. In order to prove the latter, we fix $t \in [t_{n-1}, t_n]$. Since $v^R(t,\cdot)$ is given by a linear combination of F_n and F_{n+1}, it is sufficient to show that F_n is Lipschitz continuous for each $n = 1, .., N+1$. For $x_1, x_2 \in \mathbb{R}^d$ it holds

$$\begin{aligned}
&\|F_n(x_1) - F_n(x_2)\|_2 \\
&= \|W_3^n \sin(W_2^n \sin(W_1^n x_1 + b_1^n) + b_2^n) + b_3^n - (W_3^n \sin(W_2^n \sin(W_1^n x_2 + b_1^n) + b_2^n) + b_3^n)\|_2 \\
&\leq \|W_3^n\|_F \| \sin(W_2^n \sin(W_1^n x_1 + b_1^n) + b_2^n) - \sin(W_2^n \sin(W_1^n x_2 + b_1^n) + b_2^n)\|_2 \\
&\leq \|W_3^n\|_F \|W_2^n \sin(W_1^n x_1 + b_1^n) + b_2^n - (W_2^n \sin(W_1^n x_2 + b_1^n) + b_2^n)\|_2 \\
&\leq \|W_3^n\|_F \|W_2^n\|_F \|W_1^n\|_F \|x_1 - x_2\|_2 \leq c\|x_1 - x_2\|_2
\end{aligned}$$

for a constant $c > 0$. Here, we have used that $\|W_j^n\|_F$, $j = 1, 2, 3$, is bounded and that sine is 1-Lipschitz. The latter holds because the angle addition theorems imply

$$|\sin(a) - \sin(b)| = 2\left|\cos\left(\frac{a+b}{2}\right)\sin\left(\frac{a-b}{2}\right)\right| \leq 2\left|\sin\left(\frac{a-b}{2}\right)\right| \leq |a-b| \quad \forall a, b \in \mathbb{R}.$$

Thus, F_n is Lipschitz continuous and consequently the Picard-Lindelöf theorem (Th. 4.4) yields that (21) has a unique solution $\phi : [0,1] \times \mathbb{R}^d \to \mathbb{R}^d$.

In order to prove that $\phi(t,\cdot)$ is a diffeomorphism for each $t \in [0,1]$, we apply the Cauchy-Lipschitz theorem (Th. 4.5). For this, instead of (21), we consider the equivalent system

of differential equations

$$\begin{aligned} \partial_t \varphi(t,x) &= v^R(t,\varphi(t,x)) && \forall t \in [t_{n-1},t_n], x \in \mathbb{R}^d, n = 1,...,N \\ \varphi(t_{n-1},x) &= \phi(t_{n-1},x) && \forall x \in \mathbb{R}^d, n = 1,...,N, \end{aligned}$$

which also has the unique solution $\varphi = \phi$. Due to $v^R|_{[t_{n-1},t_n]\times\mathbb{R}^d} \in C^1([t_{n-1},t_n]\times\mathbb{R}^d,\mathbb{R}^d)$, Th. 4.5 yields that $\phi|_{[t_{n-1},t_n]\times\mathbb{R}^d}(t,\cdot)$ is diffeomorphic for each $t \in [t_{n-1},t_n]$. This is synonymous with the fact that $\phi(t,\cdot)$ is diffeomorphic for each $t \in [0,1]$. □

Note that forward Euler discretization with step size $\frac{1}{N}$ of the differential equation discussed in Th. 4.6 yields exactly the deformations $\phi_0^R,...,\phi_N^R$ resulting from RNR. Hence, Th. 4.6 provides a differential equation on which RNR is based and which has a unique analytical solution that is diffeomorphic for each fixed time point.

4.4.2 Time-Discrete Solutions Resulting from RNR

From the previous section, we know that the analytical time-continuous solution of the differential equation underlying RNR is diffeomorphic for each time point. However, this does not mean that the time-discrete solution computed with RNR is also diffeomorphic at each time point. Therefore, we consider the time-discrete setting next.

The following theorem provides a condition which guarantees that the deformations resulting from RNR are diffeomorphic.

Theorem 4.7. Let $F_1^R,...,F_N^R : \mathbb{R}^d \to \mathbb{R}^d$ be the residual functions given by (27) with fixed weight matrices W_j^n and bias vectors b_j^n. Further, let $\phi_1^R,...,\phi_N^R : \mathbb{R}^d \to \mathbb{R}^d$ be the corresponding deformations given by (26) resulting from RNR. If $\|\nabla F_n^R(x)\|_2 < 1$ for all $x \in \mathbb{R}^d$ and $n = 1,...,N$, then each deformation ϕ_n^R is diffeomorphic.

Proof. Since (26) yields

$$\phi_n^R = \left(\mathrm{Id} + F_n^R\right) \circ \phi_{n-1}^R = \left(\mathrm{Id} + F_n^R\right) \circ \left(\mathrm{Id} + F_{n-1}^R\right) \circ ... \circ \left(\mathrm{Id} + F_1^R\right),$$

it is sufficient to proof that $\left(\mathrm{Id} + F_n^R\right)$ is diffeomorphic for each $n = 1,...,N$.

According to the inverse function theorem [117, Sec. 8.6.1, Th. 1], we only need to show that the Jacobian matrix $f_n(x) := \nabla(\mathrm{Id} + F_n^R)(x)$ is invertible for all $x \in \mathbb{R}^d$. Therefore, we fix $x \in \mathbb{R}^d$ and denote with $I \in \mathbb{R}^{d\times d}$ the identity matrix in $\mathbb{R}^{d\times d}$. Due to $\|\nabla F_n^R(x)\|_2 < 1$, for all $z \in \mathbb{R}^d$ holds

$$\begin{aligned} &z^\top f_n(x) z = z^\top (I + \nabla F_n^R(x)) z = \|z\|_2^2 + z^\top (\nabla F_n^R(x)) z, \quad \text{and} \\ &|z^\top (\nabla F_n^R(x)) z| \le \|\nabla F_n^R(x)\|_2 \|z\|_2^2 < \|z\|_2^2. \end{aligned}$$

Thus, $f_n(x)$ is positive-definite and therefore invertible in particular. □

Remark 4.8. Note that in the proof of Th. 4.7 we did not use the exact form of the residual functions given by (27), but only that these are continuously differentiable. Consequently, Th. 4.7 also applies to other continuously differentiable residual functions.

Th. 4.7 yields a sufficient condition to guarantee that RNR yields diffeomorphic deformations. We include this condition in the RNR approach by penalizing the gradients of the residual functions using a corresponding penalty term; see Sec. 4.5.1. This way, RNR provides preferably deformations that are diffeomorphic. As we only include the condition as a penalty and not as a constraint, no guarantee is given for diffeomorphisms in general; see also Sec. 2.2.1. Nevertheless, in Sec. 4.6 we show numerically that this approach is sufficient to obtain diffeomorphisms for the applications considered there.

Even though including the sufficient condition of Th. 4.7 in a penalty term is enough for the applications in Sec. 4.6, in the following we take a look at a common alternative for guaranteeing diffeomorphic deformations. This alternative approach is justified by the subsequent corollary, which is stated e.g. in [118] without the proof presented below.

Corollary 4.9 (adapted from [118, Th. 1])**.** Let $F_1^R, ..., F_N^R : \mathbb{R}^d \to \mathbb{R}^d$ be given by (27) with fixed weight matrices W_j^n and bias vectors b_j^n, and let $\phi_1^R, ..., \phi_N^R : \mathbb{R}^d \to \mathbb{R}^d$ be the corresponding deformations given by (26). If F_n^R is Lipschitz continuous with Lipschitz constant $M < 1$ for all $n = 1, ..., N$, then each deformation ϕ_n^R is a diffeomorphism.

Proof. This is a direct conclusion of Th. 4.7, since the condition on the Lipschitz constant of F_n^R implies for each $x \in \mathbb{R}^d$

$$\begin{aligned}\|\nabla F_n^R(x)\|_2 &= \sup_{\substack{z\in\mathbb{R}^d,\\ \|z\|_2=1}} \left\|(\nabla F_n^R(x))z\right\|_2 = \sup_{\substack{z\in\mathbb{R}^d,\\ \|z\|_2=1}} \left\|\lim_{h\to 0}\frac{1}{h}\left(F_n^R(x+hz) - F_n^R(x)\right)\right\|_2 \\ &= \sup_{\substack{z\in\mathbb{R}^d,\\ \|z\|_2=1}} \lim_{h\to 0}\frac{1}{h}\left\|F_n^R(x+hz) - F_n^R(x)\right\|_2 \leq \sup_{\substack{z\in\mathbb{R}^d,\\ \|z\|_2=1}} \lim_{h\to 0}\frac{M}{h}\|hz\|_2 = M < 1.\end{aligned}$$

□

Remark 4.10. Note that the exact form of the residual functions is also insignificant for the proof of Cor. 4.9.

Recently, several works on ResNets guaranteed diffeomorphisms by ensuring that the Lipschitz constants of the residual functions are less than one; see e.g. [28, 118]. In order to guarantee the Lipschitz condition different normalization methods are used; see e.g. [119, 120, 121]. The advantage of this approach over including the sufficient condition of Th. 4.7 in the penalty term is that diffeomorphic deformations are guaranteed. The disadvantage, however, is the additional computational effort required to normalize the residual functions. This motivates us to use the simpler penalty-based approach for the experiments in Sec. 4.6 and Sec. 4.7.

4.5 Implementation Details

In the previous sections, we have provided a theoretical analysis of RNR. In the remainder of this chapter, we will examine the method from an experimental perspective. Therefore, we summarize implementation details of the RNR method in this section. The experiments presented in Sec. 4.6 and Sec. 4.7 are based on this implementation.

While RNR was introduced for 2D and 3D registration problems, the experiments in the following sections focus only on the more complex case of 3D registration problems. Thus, from now on, by an image we mean a discrete image composed of voxels; see Sec. 2.1. For computational reasons we assume a cuboid image domain that is uniformly scaled to fit into the cube $\Omega \subset [-1,1]^3$.

All experiments presented in the following are implemented in Python 3.10.9 [122] using PyTorch 2.0.0 [36]. The results in Sec. 4.6 are generated on a 12-core Intel(R) Xeon(R) Gold 6128 CPU system with 95GB of RAM, NVIDIA GeForce RTX 2080 and CUDA 12.6. The experiments in Sec. 4.7 are performed on a 24-core AMD EPYC 74F3 system with 256GB of RAM, NVIDIA A100 and CUDA 12.6.
Our implementation builds on the code basis of Wolterink et al. [38], which contains a method for intensity-based image registration based on INRs.

4.5.1 Registration Settings

As described in Sec. 4.2, in the RNR method an energy functional of the form (29) is minimized, which consists of a data fitting term D, a regularizer S and a penalty term P. More precisely, in (29) we set

$$D[y] := \frac{1}{N}\sum_{n=1}^{N} D_n[\phi_n], \quad S[y] := \frac{1}{N}\sum_{n=1}^{N} S_n[\phi_n] \quad \text{and} \quad P[y] := \frac{1}{N}\sum_{n=1}^{N} P_n[\phi_n]$$

with the vector fields $\phi_1, ..., \phi_N$ being the successive deformations corresponding to the final transformation y defined by (26). Note that the above notations are not well defined in general since vector fields ϕ_n uniquely define a final transformation y but the reverse does not hold. Nevertheless, we decided to use these notations in order to keep the formulation short. Numerically this inconsistency is no problem, since the optimization is directly performed on the weights of the network; see Rem. 4.2 for a more detailed discussion.

Next, we present concrete choices for the functionals D_n, S_n and P_n.
Appropriate choices of the data fitting term depend on the application. In the experiments presented in Sec. 4.6.2–4.6.4, which consider academic registration problems, we use the MSE as similarity measure for simplicity; see (2). For the registration problem with lung data in Sec. 4.6.5, the distance measure NCC, see (3), is utilized in order to account for varying intensity changes in the data. Whereas for the calculation of the breathing model in Sec. 4.7, a combination of MSE and NCC is selected based on the data provided for this purpose.
As discussed in Sec. 4.3, we use the regularizer given by (31) in order to guarantee existence of solutions of the minimization problem.
Further, we penalize the gradients of the residual functions F_n^R given by (27) by setting

$$P_n[\phi_n] := \frac{1}{|B|}\sum_{x\in B} \|\nabla F_n^R(x)\|_F^2 \quad \text{for } n = 1, .., N, \tag{35}$$

where B is the set of voxel positions that are taken into account in the optimization; see Sec. 4.5.2. The choice of this smoothness penalty is motivated by Th. 4.7, which states that the deformations ϕ_n^R calculated with RNR are diffeomorphisms if the condition

$$\|\nabla F_n^R(x)\|_2 < 1 \quad \forall x \in \Omega, n = 1, ..., N$$

Table 4.1: *Number of parameters in the network in RNR for the 3D case ($d = 3$) with $N \in \{1, 5\}$ time steps and network width $q \in \{8, 30, 100\}$ calculated using (28).*

N \ q	8	30	100
1	131	1.143	10.803
5	655	5.715	54.015

holds; see the discussion following Th. 4.7. As is usual with registration, appropriate choices for the regularization parameter α_S and the penalty parameter α_P depend on the specific registration problem; see e.g. [9]. However, since RNR is not very sensitive to the choice of the regularization parameter α_S, see Exp. 4.22, we set $\alpha_S = 10^{-4}$ in all experiments. With this choice the regularization term turned out to have no major influence on the resulting deformation, but still guarantees the existence of solutions. We desire this behavior since there is no physical foundation for the utilized regularizer; see Sec. 4.3.2 for further details. The penalty parameter α_P influences the smoothness of the deformation and must therefore be selected individually for each application. We perform an automatic search to determine α_P. The details are presented in Sec. 4.6 and Sec. 4.7.
In order to evaluate the deformed template images $T \circ \phi_n$, sub-voxel image intensity values are computed through trilinear interpolation. The derivative $\nabla(T \circ \phi_n)$ at sub-voxel positions, which is required during the optimization process, is computed with right derivatives.

For the number of time steps performed in a registration with RNR, we use $N = 1$ and $N = 5$. While the first choice for N gives insight about the deformation produced with one time step, the latter shows how the deformations look after several time steps and allows the usage of intermediate data.

The choice of the network width q within the residual functions varies. A total of three different network widths are investigated in the following experiments: $q = 8, 30, 100$. These values were chosen because the number of parameters in the corresponding networks differs by a factor of ten; see Tab. 4.1. While the small and medium sized network ($q = 8, 30$) proved to be reasonable for the structured academic registration problems considered in Sec. 4.6.2 and Sec. 4.6.3, the large network ($q = 100$) is more flexible and thus appropriate for the registration problems with real data presented in Sec. 4.6.5 and Sec. 4.7. Note that all networks utilized in the next sections contain a relatively small number of parameters compared to other registration methods based on INRs. For example, in [38] the network contains approximately $1.3 \cdot 10^5$ parameters, [39] uses between $6.7 \cdot 10^4$ and $1.3 \cdot 10^5$ parameters and [40] even utilizes approximately $1.3 \cdot 10^7$ parameters. In contrast, the largest network used in RNR in the following sections contains only $5.4 \cdot 10^3$ parameters; see Tab. 4.1. We will show that the relatively small number of parameters in RNR is already sufficient for the registration problems under investigation.

4.5.2 Optimization Settings

For the optimization, we use ADAM [107], which has proven to be a successful stochastic gradient descent-based optimization algorithm in recent years; see [123, 124] for popular applications of ADAM. The learning rate is set to 10^{-3}, which is a commonly used value for the step size, see e.g. [40, 60], and has also proven to be suitable in our experiments.

We initialized the weight matrices and bias vectors for each residual function with values drawn from a normal distribution with zero mean, using a standard deviation of one for the first two linear layers and a standard deviation of 10^{-3} for the last layer. This ensures that the initial deformation is close to the identity transformation. The intuitive strategy of setting the initial deformation to the identity, i.e. initializing all parameters with zero, is not reasonable for learning-based approaches. If doing so, the derivatives of the network output with respect to the weights in one layer would be identical, so that the updated weights in this layer would also remain identical throughout training [125].

In each epoch, one optimization step is performed based on a share of the image voxels, such that over multiple epochs all voxels are considered in random order. Naturally, using a small proportion in each epoch saves computation time, while a large proportion improves the registration quality.
Therefore, in the experiments based on large data in Sec. 4.6.2 and Sec. 4.7, we use 1% of the image voxels in each epoch. In Sec. 4.6.3, the data consists of significantly fewer voxels than in the other examples, and thus all voxels can be used in each epoch. For the data in Sec. 4.6.5, using 3% of the voxels turned out to be a good compromise between computational effort and registration quality.

In RNR, terminating the optimization using classical stopping criteria such as no significant change in the loss or reaching a certain threshold for the loss, see e.g. [11], is not suitable. The reason for this are the large stochastic variations in the loss, due to stochastic sampling and the stochastic gradient descent scheme. Instead, we fix the number of epochs, as is usual in learning-based registration approaches; see e.g. [34, 35]. We run the optimization for 300 epochs in the experiments presented in Sec. 4.6 and Sec. 4.7, which was empirically determined to be a reasonable value for the considered registration problems. A validation of this choice for the stopping criteria is presented in Exp. 4.23.

4.6 Validation of RNR

In this section, we present a comprehensive experimental validation of RNR. The results give an impression of the deformations produced by RNR. We demonstrate that RNR provides deformations that build on each other in a meaningful way. Furthermore, we compare RNR with selected state-of-the-art registration approaches, such as the classical optimization-based registration approach [126] and the CNN-based method [60]. As already mentioned in the previous section, we focus on 3D registration problems in this validation.

Before we dive into details about the validation, in Sec. 4.6.1 we shortly introduce the visualizations that will be used to demonstrate the registration results in the following sections. The subsequent validation is then divided into four parts: In Sec. 4.6.2, an academic registration problem in the form of Problem 2 is investigated, whose structure allows to understand what the deformations resulting from RNR look like. We compare the deformations produced by RNR using different network widths and show that the registration quality for this problem generally benefits from the usage of intermediate data. Additionally, we demonstrate that for the considered example the adaptation of the residual functions proposed in Sec. 4.2 indeed yields larger deformations compared to the original residual functions used by Ben Amor et al. [41], which was the motivation for this adaptation. A comparison with the classical LDDMM method, which was

introduced in Sec. 4.1 and forms the basis for RNR, shows that RNR achieves higher registration quality with faster computation time for the considered example. Afterwards, in Sec. 4.6.3 registration problems with X-ray hand data are investigated, which are based on transformations not covered by the structured example in Sec. 4.6.2. We show that RNR is able to provide basic transformations such as translations, scalings and small rotations, but struggles with large rotations and partially with local minima. In Sec. 4.6.4, a brief discussion on the stability and convergence behavior of the optimization within RNR is presented. In Sec. 4.6.5, we compare RNR with state-of-the-art registration methods regarding the registration quality on a publicly available dataset containing lung CT data. We show that RNR can compete with these commonly used methods. Lastly, in Sec. 4.6.6 we summarize the main results of the presented validation.

4.6.1 Overview of Visualization Types

Visualization of 3D images and deformations on a 2D plane always involves limitations. Therefore, many visualization approaches exist, each with different advantages and disadvantages; see e.g. [127] for an overview of 3D visualization methods. Since the focus of this thesis is not on different visualization approaches, we have decided on four intuitive types of visualizations, which will now be described. An example for each type is shown in Fig. 4.6.

All visualizations used in the following are based on 2D image slices taken from the 3D images, as using 2D slices is one of the simplest ways to visualize 3D images on a 2D plane. While visualizing only one slice taken from the center of the 3D image allows details in this 2D image to be recognized, combining several evenly distributed slices in an image collage gives a rough impression of the entire 3D image; see the first two illustrations in Fig. 4.6. In order to give insight into the 3D deformations resulting from registrations, we show projections of deformed grid slices; see the third illustration in Fig. 4.6. More precisely, for this type of visualization we fix a slice of a 3D grid in the image domain, deform this slice using the given 3D deformation and project the deformed slice to a 2D plane. For orientation purposes, the corresponding slice of the template image is also displayed in the background of the resulting grid. Finally, as shown on the right in Fig. 4.6, we use difference images created by subtracting the template from the reference image. Since difference images are particularly well suited for illustrating subtle differences between images, we utilize theses for the registration problem with lung data; see Sec. 4.6.5.

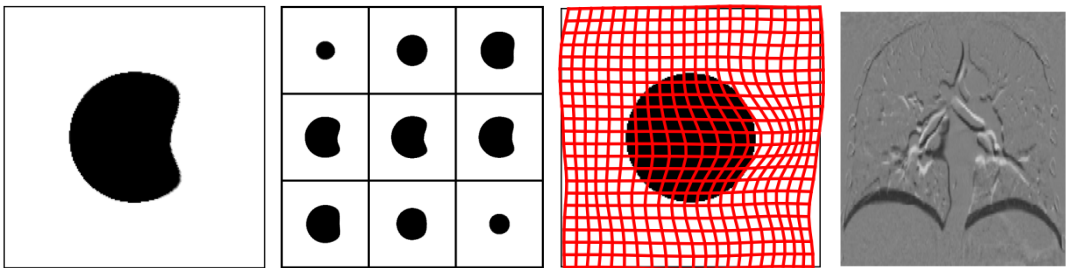


Figure 4.6: *2D visualizations of exemplary 3D images and deformation. From left to right: image slice, image collage containing nine slices of an image, projection of deformed grid slice with template in the background, slice of a difference image between two corresponding images; see Sec. 4.6.1 for details.*

4.6.2 Example 3D-C: Handling Large Deformations

In this section, we consider a registration problem, whose structure allows us to understand what the deformations resulting from RNR look like. The problem under consideration is a 3D adaptation of the example examined by Christensen et al. [128], who performed a registration of a 2D disc to an image of the letter "C"; see Fig. 4.7.
Christensen et al. [128] proposed this example in order to illustrate the limitations of elastic registration methods [65], which are unable to deform the disc into the "C". More precisely, the example is interesting for multiple reasons [128]. Firstly, the template and the reference image differ greatly from each other. Since the right side of the disc has to move a large distance from its undeformed to its deformed position, large deformations are required to solve the registration problem. Secondly, the reference image contains corners while the template image does not. This allows to validate whether calculated deformations are continuously differentiable. Lastly, many points of the disc have to follow nonlinear paths to deform the disc into the "C". Due to all three aspects, elastic models are unable to solve this registration problem.

We show that RNR, on the other hand, is capable of handling such large deformations. Additionally, although deformations computed with RNR are smooth, they can be a good approximation of non-smooth functions that create corners. In order to demonstrate this, we adapt the registration setting of [128] as follows: The template image shows a homogeneous filled ball. Furthermore, we create five successive reference images, which show 3D C-shapes that can be imagined as balls that have been indented from one direction. The cavity is small for the reference image corresponding to the first time point and increases in the subsequent time points. Fig. 4.7 visualizes slices of the images. Each 3D image consists of $200 \times 200 \times 200$ cubic voxels.

The validation in this section is organized in multiple phases. In the first two parts, we neglect the intermediate reference images and perform registrations only based on the final reference image using either one or five time steps. Afterwards the intermediate reference images are taken into account as well, which turns out to have a significant impact on the registration. We compare the results for different network widths. All in all, RNR proves to yield large deformations and is capable of solving the considered registration problem. In the subsequent experiments, RNR is compared with a registra-

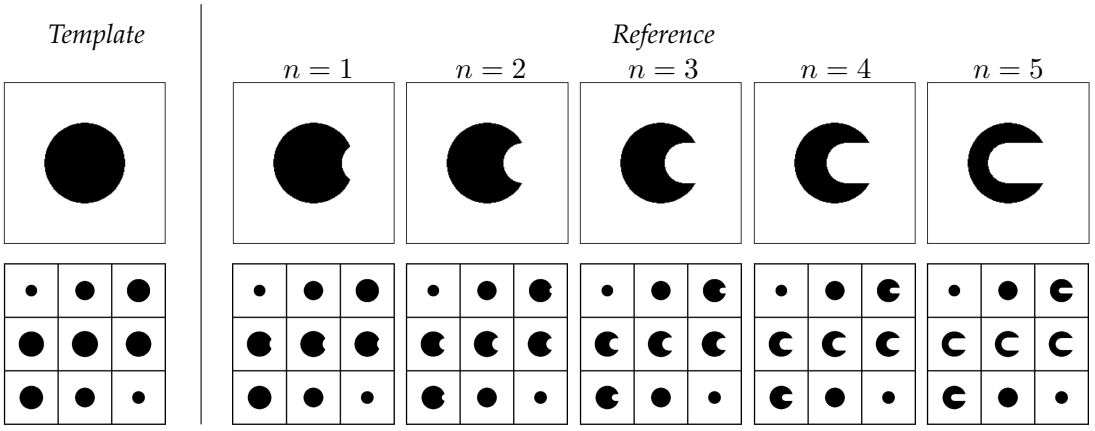


Figure 4.7: *3D template image and reference images of the 3D-C example, which are created by adapting the data [128] to 3D. For details about the visualization see Sec. 4.6.1.*

tion approach that uses the original residual functions [41], as well as with the classical LDDMM method. Lastly a validation on the computation time of all methods used in this section is presented.

Choice of the Network Structure and the Penalty Parameter

For the following experiments, the implementation presented in Sec. 4.5 is utilized. Since the values for the network width q and the penalty parameter α_{P} have to be selected depending on the experiment, we will now describe how these are determined.

In order to evaluate how the deformations resulting from RNR change with the network width, we examine all three different cases for the dimension q mentioned in Sec. 4.5.1, i.e. $q = 8, 30, 100$. While $q = 30$ was chosen as baseline because in preliminary experiments reasonable results were achieved with this network width, $q = 8$ and $q = 100$ were chosen for comparison with a much smaller and a much larger network; see also Sec. 4.5.1 for a discussion of these choices.

The idea for the choice of α_{P} is to set the parameter as small as possible, but large enough such that the final and intermediate deformations resulting from RNR are diffeomorphic, i.e. the third condition of Problem 2 is fulfilled. This allows us to focus on the behavior of the deformations caused by the network structure given by (26) and (27). More precisely, the penalty parameter α_{P} is automatically selected in each experiment as follows: A grid search is executed taking into account values for α_{P} between 0 and 1 with a step size of 0.02. For each value of α_{P} considered, a registration with RNR is performed and checked as described below to determine whether the resulting deformations are diffeomorphic. The smallest value for α_{P} that satisfies the latter condition is chosen.
One sufficient criterion for a deformation to be diffeomorphic is that the Jacobian determinant is greater than zero in $\mathbb{R}^3$. To save computation time, we only check this criterion on a set of 10^5 randomly selected points on a grid within the image domain. This grid has the size of the image data in terms of voxels. Note that with this procedure we cannot guarantee that the obtained deformations are diffeomorphic on the entire space $\mathbb{R}^3$. Nevertheless, this is a commonly used approach to check whether deformations are diffeomorphic; see e.g. [34, 39, 40].
The resulting penalty parameters are listed in Tab. A.1 in the Appendix for all experiments in this section.

Experiment 4.11 (RNR with one time step)**.** In the first experiment, we use RNR with only one time step in order to get insight into how deformations resulting from a single time step look like. Thus, we set $N = 1$ and use only the final reference image in the registration while the intermediate reference images are not taken into account. The deformed template image as well as the deformed grid is visualized at the top of Fig. 4.8. The small network ($q = 8$) generates a deformation that is almost the identity and thus does not produce an image with a C-shape. The image resulting from the medium and large network ($q = 30, 100$) show a C-shape, but the cavity in the C is not nearly as pronounced as in the reference image. In all three cases, the deformation is smooth and too restrictive to deform a ball to the desired C-shape. Additionally note that although both the template image and the reference image have a symmetrical structure, the deformed template image does not have this property. Since the network structure used within RNR allows to represent symmetric deformations, this must be due to the optimization within the method.

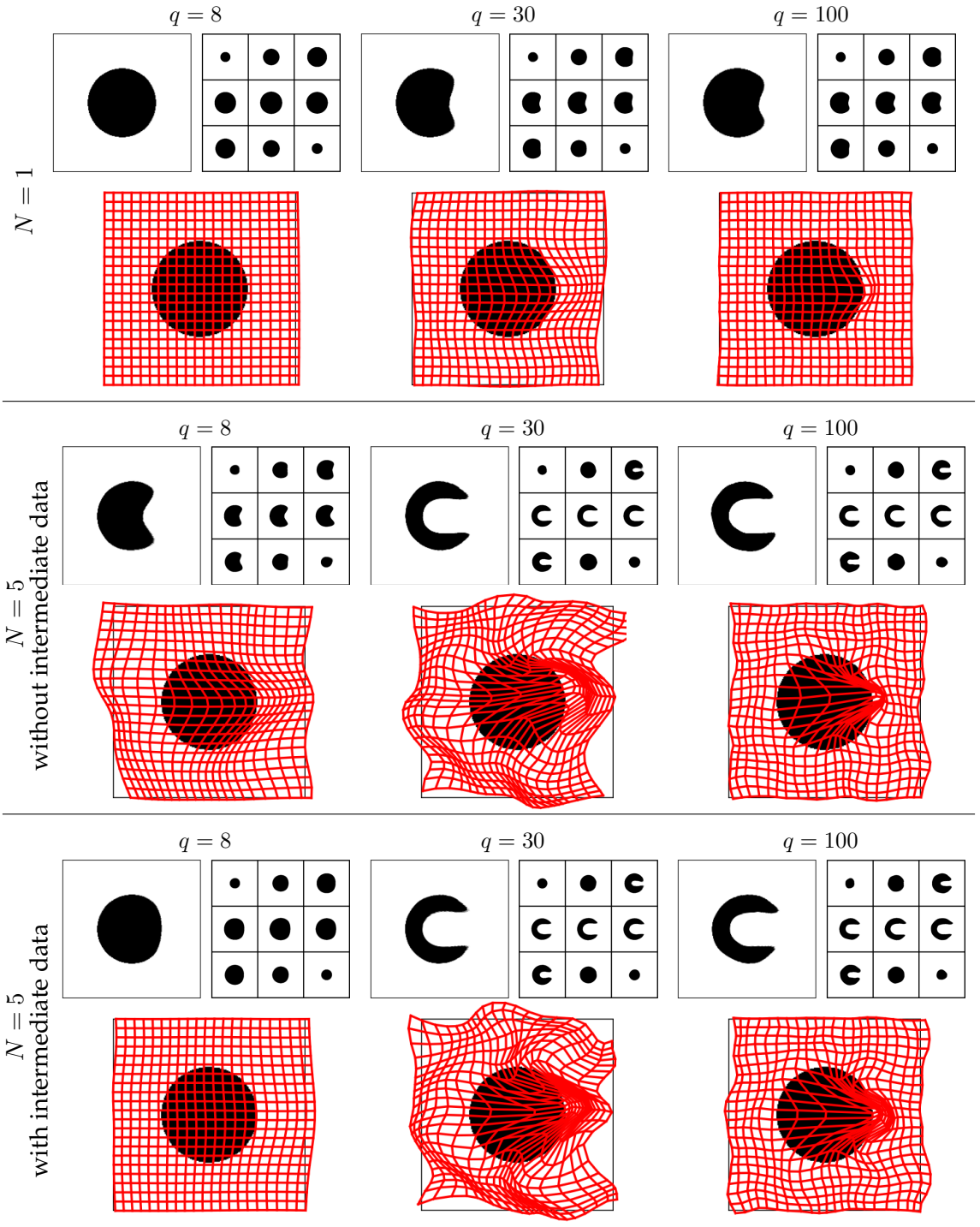


Figure 4.8: *Final deformed template image and deformed grid resulting from RNR using network widths* $q = 8, 30, 100$. ***Top:*** *results using one time step; see Exp. 4.11.* ***Middle:*** *results using five time steps and only the final reference image; see Exp. 4.12.* ***Bottom:*** *results using five time steps and all reference images; see Exp. 4.13. For details about the visualization see Sec. 4.6.1.*

Experiment 4.12 (RNR with multiple time steps omitting intermediate data)**.** Next, we perform a registration using RNR with five time steps, i.e. $N = 5$. In this way, we allow the method to produce intermediate deformations. We still do not include the intermediate data. Instead, as in the previous experiment, the registration is only guided by the final reference image. For the loss function (29) this means that for the data fitting term there is only one summand and thus we omit the weighting by $\frac{1}{N}$ for the data term in order to ensure comparability with the other experiments. The resulting final image and deformed grid is presented in the middle of Fig. 4.8. The result shows that by allowing RNR to produce intermediate deformations, the method is able to deform a ball into the desired C-shape. Thus, RNR can generate large deformations that make it possible to map even very different images onto each other. In this structured registration example, a relatively small network width of $q = 30$ is already sufficient. As we have five times as many parameters when using five instead of one time step, the greater flexibility of RNR in this experiment compared to the previous experiment is not surprising. However, note that the network corresponding to $N = 1, q = 100$ contains twice as many parameters as the network with $N = 5, q = 30$; see Tab. 4.1. Nevertheless, in the latter case the deformation is larger. From this we conclude that the flexibility of RNR depends more on the number of time steps than on the network width.
Note that for $q = 30, 100$ also the corners in the reference images are relatively well represented by the deformations. This means, though deformations generated by RNR are theoretically in $C^\infty(\mathbb{R}^3, \mathbb{R}^3)$, in practice the method can generate deformations that can hardly be distinguished from non-differentiable transformations.
When considering the figures, it should be noted that, due to the choice of the penalty parameter as described in Sec. 4.5, no grid folding occurs. For the cases $q = 30, 100$ the grid seems to fold, but this only an artifact of the visualization.
In order to understand how the template image is deformed over time, Fig. 4.9 visualizes the deformed template image and the corresponding deformed grid after each individual time step. The behavior is shown exemplary for $q = 30$, since for the small and large

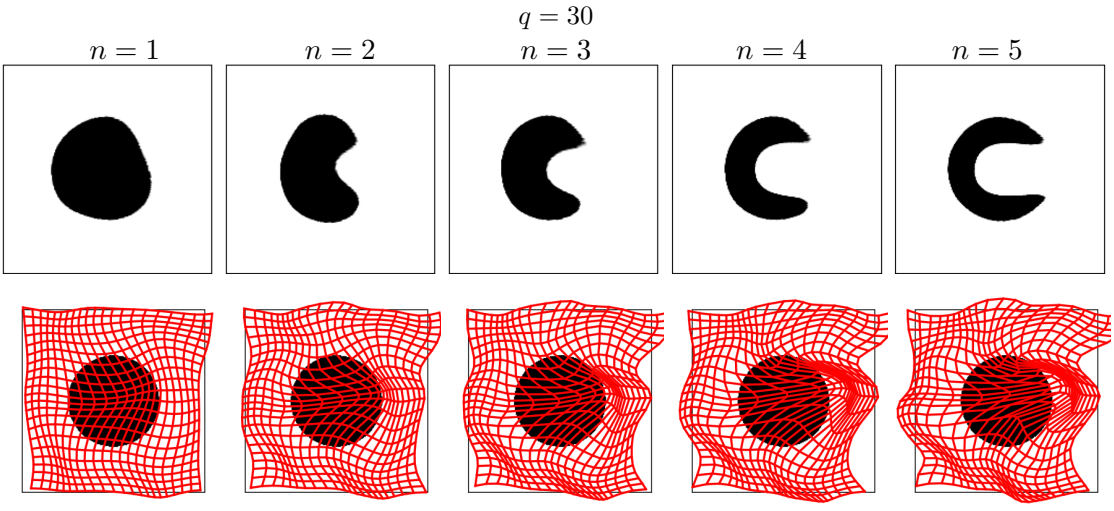


Figure 4.9: *Deformed template image and deformed grid after each time step resulting from RNR using $N = 5$ time steps and network width $q = 30$ without intermediate reference images; see Exp. 4.12. For details about the visualization see Sec. 4.6.1.*

network we obtain analogous results. Although the fully deformed template image is well aligned with the final reference image, the intermediate results differ greatly from the intermediate reference images. Since these were not incorporated in the registration this is not surprising. In the next experiment, we show that including the intermediate reference images solves this issue.

Experiment 4.13 (RNR with multiple time steps including intermediate data)**.** In the next experiment, we retain the network structure in RNR as used in the previous experiment, but include the intermediate reference images in the registration. In this way, we again allow the method to produce four intermediate deformations, but we additionally support the calculation of these deformations by providing data corresponding to the intermediate time points. The final deformed template image and the deformed grid is presented in the bottom of Fig. 4.8. For the medium and large network width ($q = 30, 100$) the final results do not differ significantly from the one of the previous experiment. In these cases, RNR yields deformations that are able to generate the desired C-shape. Since the network architectures used coincide with those of Exp. 4.12, this behavior matches our expectation. Surprisingly, for the small network width ($q = 8$) the ball is not deformed as much as in the previous experiment. One explanation for this is that the data fitting term now takes into account not only the final time point, but also the time points in between, so that the last time point is less important.
The deformed template images and deformed grids corresponding to each individual time point are shown in Fig. 4.10. In contrast to the previous experiment, at least for the medium and large network ($q = 30, 100$) the deformed template image matches the reference images for each time point well.
Summarizing, including the intermediate reference turned out to be both partly advantageous and partly disadvantageous for the registration of the 3D-C example. Using the small network width, the incorporation of the intermediate data had a negative impact on the quality of the final registration result. However, in the other two cases the quality of the final registration result was not visibly affected by the inclusion of the additional data, while the intermediate results have clearly benefited from this.

Experiment 4.14 (Influence of the adaptation of the residual functions in RNR)**.** Next, we validate whether the adaptation of the original residual functions of Ben Amor et al. [41] does indeed result in larger deformations, as described in Sec. 4.2. Since with the adapted residual functions the registration problem is already solved well for the medium network ($q = 30$) using the intermediate reference data, we repeat Exp. 4.13 with the original residual functions (25). The parameters were selected as in the previous experiments. The resulting deformed template images and deformed grids corresponding to each individual time point are shown in Fig. 4.11. There is a large difference to the results attained in Exp. 4.13; see Fig. 4.10. The use of the adjusted residual functions leads to a much larger deformation, which is more suitable to deform the ball to the C-shape.

Experiment 4.15 (Comparison of RNR with LDDMM)**.** In this validation part, we compare RNR with a registration method belonging to the classical LDDMM framework as described in Sec. 4.1.1. More precisely, we use torch-lddmm [129] with the default settings except for minor adaptations. Since the gradient descent step size was too small for the considered example, we set the step size to 10^{-1}, which turned out to be a reasonable choice. The regularization parameter is optimized in the same way as the penalty

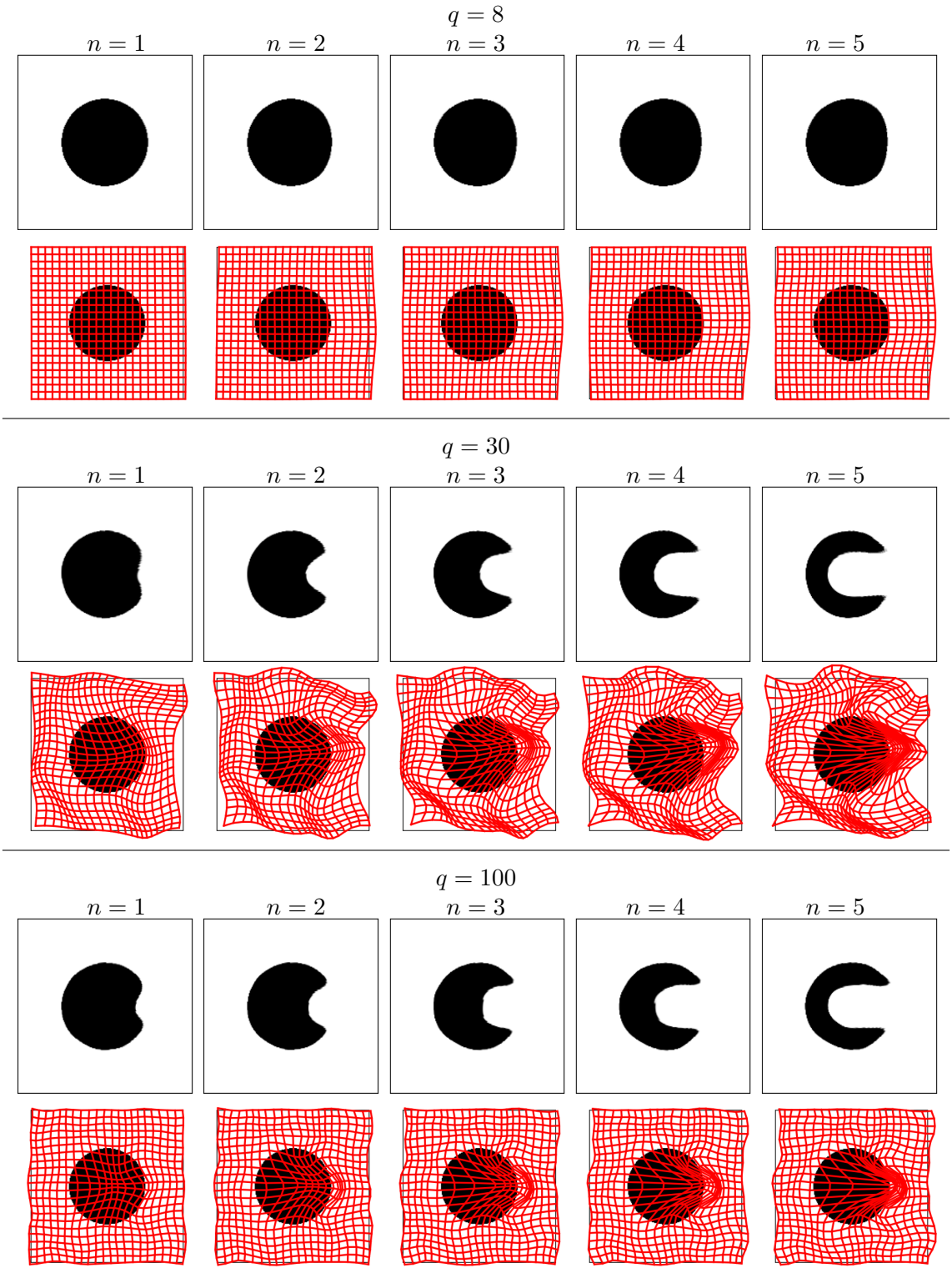


Figure 4.10: *Deformed template image and deformed grid after each time step resulting from RNR using $N = 5$ time steps and network widths $q = 8, 30, 100$ when using intermediate reference images; see Exp. 4.13. For details about the visualization see Sec. 4.6.1.*

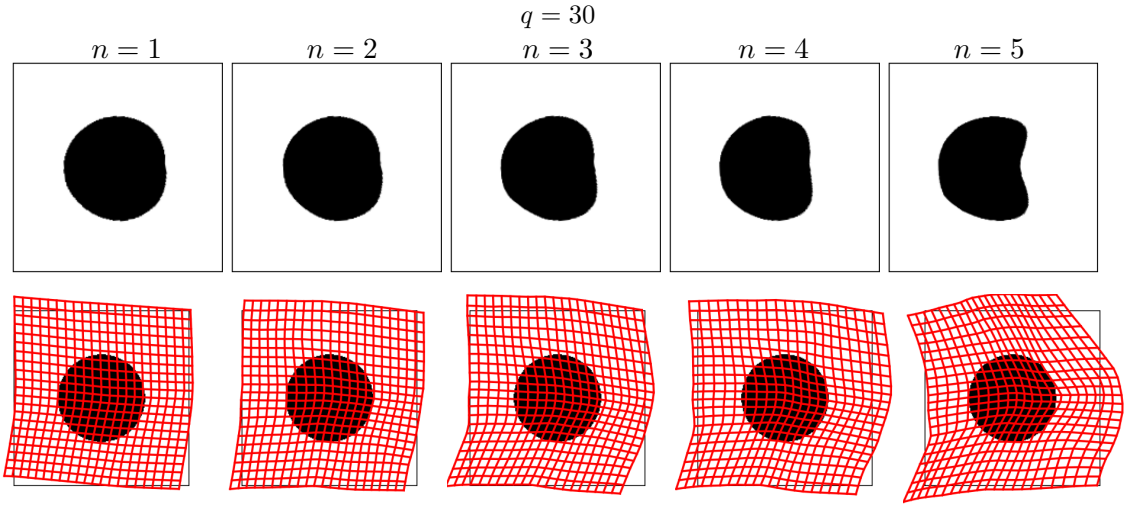


Figure 4.11: *Deformed template image and deformed grid after each time step resulting from RNR with original residual function of [41] using $N = 5$ time steps and network width $q = 30$, including intermediate reference images; see Exp. 4.14. For visualization details see Sec. 4.6.1.*

parameter in the previous experiment, such that the weighting of the regularization is as small as possible without allowing for grid folding. The maximum number of iterations in the optimization is set to 300, but in both registrations presented in the following, the optimization stopped prematurely because the energy change threshold was reached.

In order to compare LDDMM with RNR, we first use LDDMM with $N = 5$ time steps analogous to the previous experiments for RNR. The computed final image and deformed grid is shown on the left of Fig. 4.12. The resulting deformation is quite small compared to those produced in Exp. 4.13 using RNR with network width $q = 30, 100$ and including the intermediate reference data. When enlarging the number of time steps to $N = 25$ LDDMM is also able to generate deformations that map the ball to the desired

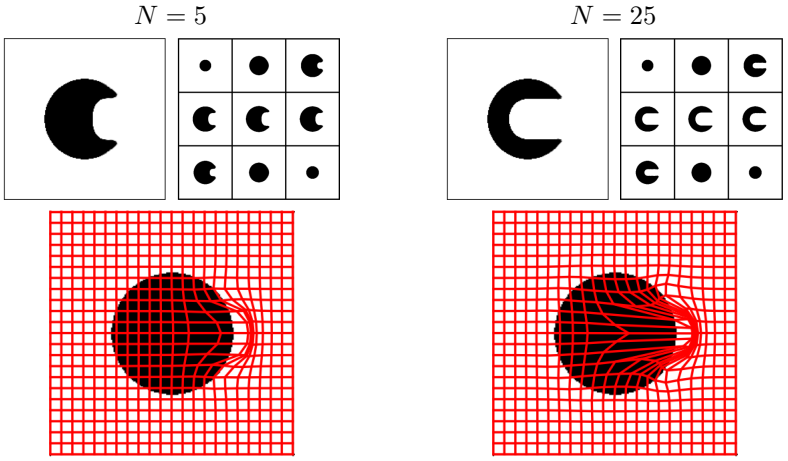


Figure 4.12: *Final deformed template image and deformed grid resulting from LDDMM using $N = 5$ and $N = 25$ time steps; see Exp. 4.15. For details about the visualization see Sec. 4.6.1.*

Table 4.2: *Computation time in seconds for each method utilized for the 3D-C example using different number of time steps N and network widths q: RNR without intermediate data (RNR w/o) with $N = 1, 5$ and $q = 8, 30, 100$; RNR with intermediate data (RNR w) with $N = 5$ and $q = 8, 30, 100$; RNR with the original residual functions [41] and intermediate data (RNR w+[41]) with $N = 5$ and $q = 30$; torch-lddmm [129] with $N = 5, 25$.*

Method	RNR w/o						RNR w			RNR w+[41]	LDDMM	
N	1			5			5			5	5	25
q	8	30	100	8	30	100	8	30	100	30	-	-
Time [s]	6	8	21	39	71	204	53	90	250	55	37	500

C-shape; see Fig. 4.12 right. This comes at the price of computation time as discussed in the next experiment. Note that the used implementation of LDDMM does not take the intermediate reference images into account.
A fundamental difference between RNR and the LDDMM implementation is how the methods deal with the image boundary $\partial\Omega$. While LDDMM fixes the boundary, RNR is not restricted by any boundary conditions. This means on the one hand that RNR is universally applicable and it is not necessary to consider which boundary conditions are appropriate for each registration task. On the other hand, if a registration problem comes with specific boundary conditions, these cannot be guaranteed using RNR. An easy implementable adaptation of RNR that at least comes close to fulfilling certain boundary conditions is to include them as a penalty in the minimized loss function. Adapting RNR to incorporate boundary conditions as constraints is not straightforward.

Experiment 4.16 (Computation time)**.** Tab. 4.2 summarizes the computation times for all methods considered in this section for the 3D-C example. Note that with RNR the entire cavity in the C-shape is generated in $71\,\mathrm{s}$, when using RNR without the intermediate reference data and network width $q = 30$. LDDMM requires $500\,\mathrm{s}$ for this task and we need to set the number of time steps to $N = 25$. Thus, RNR can generate large deformations much faster than the LDDMM approach.

4.6.3 Example Hands: Handling Affine Deformations and Local Minima

Though the 3D-C example, which was examined in Sec. 4.6.2, already gives a comprehensive insight into what deformations resulting from RNR look like, some aspects are not covered by this example. Still open questions are whether simple transformations such as translations, scaling and rotations can be computed with RNR and how the method deals with local minima. In order to answer these questions, we utilize the hand data as visualized in Fig. 4.13. In this section the MSE between the template and reference is specified on the left side of each depicted template image, since the image similarity of the hand data is more difficult to assess visually than for the 3D-C example. The data in Fig. 4.13 result from an adaptation of the original X-ray hand data [130]. The original hand data consist of 2D images of size $20\,\mathrm{cm} \times 25\,\mathrm{cm}$ with 128×128 pixels and can be downloaded from the software toolbox documented in [11]. Note that the implementation of RNR used for the 3D-C example cannot handle 2D data. Although it is possible to adapt the implementation of RNR to process 2D data, we decided to adjust the hand data instead due to time constraints. Thus, we stacked together five identical

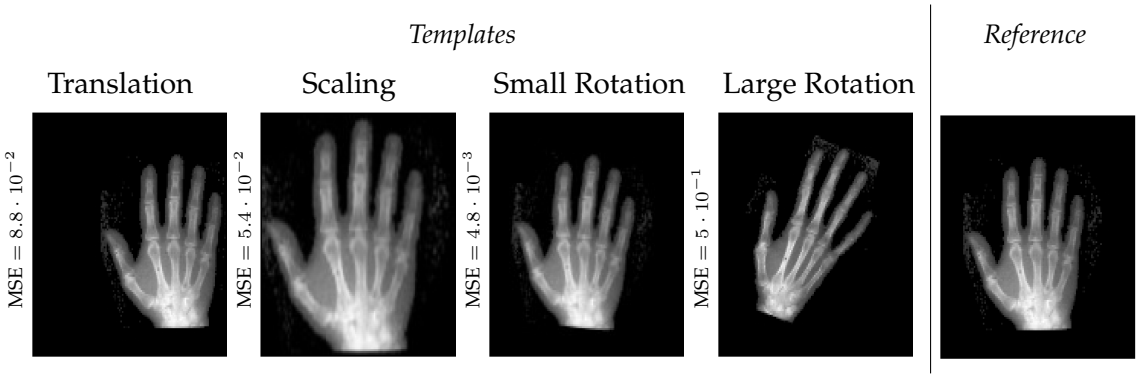


Figure 4.13: *3D template images and reference image of the hand examples. For each experiment the MSE between the corresponding template image and the reference image is specified on the left-hand side. For details about the visualization see Sec. 4.6.1.*

2D images to get 3D data. In this section, we conduct four experiments to investigate different types of deformations: a translation, a scaling, a small rotation by five degrees and a larger rotation by 22.5 degrees with an additional nonlinear deformation. While in the first three experiments the template image results from the reference image by only a translation, scaling or rotation, in the last experiment the ground truth transformation does not only consist of a pure rotation. Instead, in the latter experiment we use for the template image a rotated version of the original template in [130]. This means that in this experiment the template and reference images show different hands and thus the ground truth deformation contains also a nonlinear component. Note that for all hand examples in this section no intermediate reference images are provided.

Choice of the Network Structure and the Penalty Parameter

In order to keep the validation in this section short, we limit the versions of the network structure within RNR under investigation: In each experiment, we utilize RNR with $N = 1$ time step and small or medium network width, i.e. $q = 8, 30$. We use one time step since the data in this section comes without intermediate data. The choice of the utilized network width is justified by the fact that the affine transformations on which the data are based on can be parameterized by only a few parameters. Thus, we assume that even the small network ($q = 8$) is sufficient, but we consider the medium network ($q = 30$) as well to be on the safe side.

The penalty parameter α_{P} is chosen in the same way as in the 3D-C example in Sec. 4.6.2. The resulting values are summarized in Tab. A.2 in the Appendix.

Experiment 4.17 (Translation)**.** First, we consider the example in which the template image results from the reference image using a translation; see Fig. 4.13 left. More precisely, the applied underlying transformation is a shift in x_1-direction of around $4.6\,\mathrm{cm}$, which corresponds to about a quarter of the image width for the image size described above. The difficulty of this example is that a local minimum is reached as soon as any finger in the template matches any finger in the reference image. In the left-most column of Fig. 4.14 the final images and deformed grids are shown. While the network within RNR with small width ($q = 8$) is able to represent the transformation of the hand, the larger network ($q = 30$) gets stuck in a local minimum (overlap of thumb with index finger)

and yields an image of an unnaturally deformed hand. These results are unsatisfactory, since by (26) and (27) the network structure within RNR can represent translations exactly and translations are included in the kernel of both the regularizer (31) and the penalty (35). Thus, RNR should be able to provide the transformation of the hand exactly. If the penalty parameter is increased by a factor of 100, the expected behavior of RNR is obtained, i.e. the network within RNR is now able to represent the translation. The corresponding results are visualized in Fig. 4.15 for the medium sized network.
One might expect the smoother transformation in Fig. 4.15 to yield a smaller value of the original functional than the result shown in Fig. 4.14. In order to verify this, denote for the moment the deformation resulting from the registration with larger penalty parame-

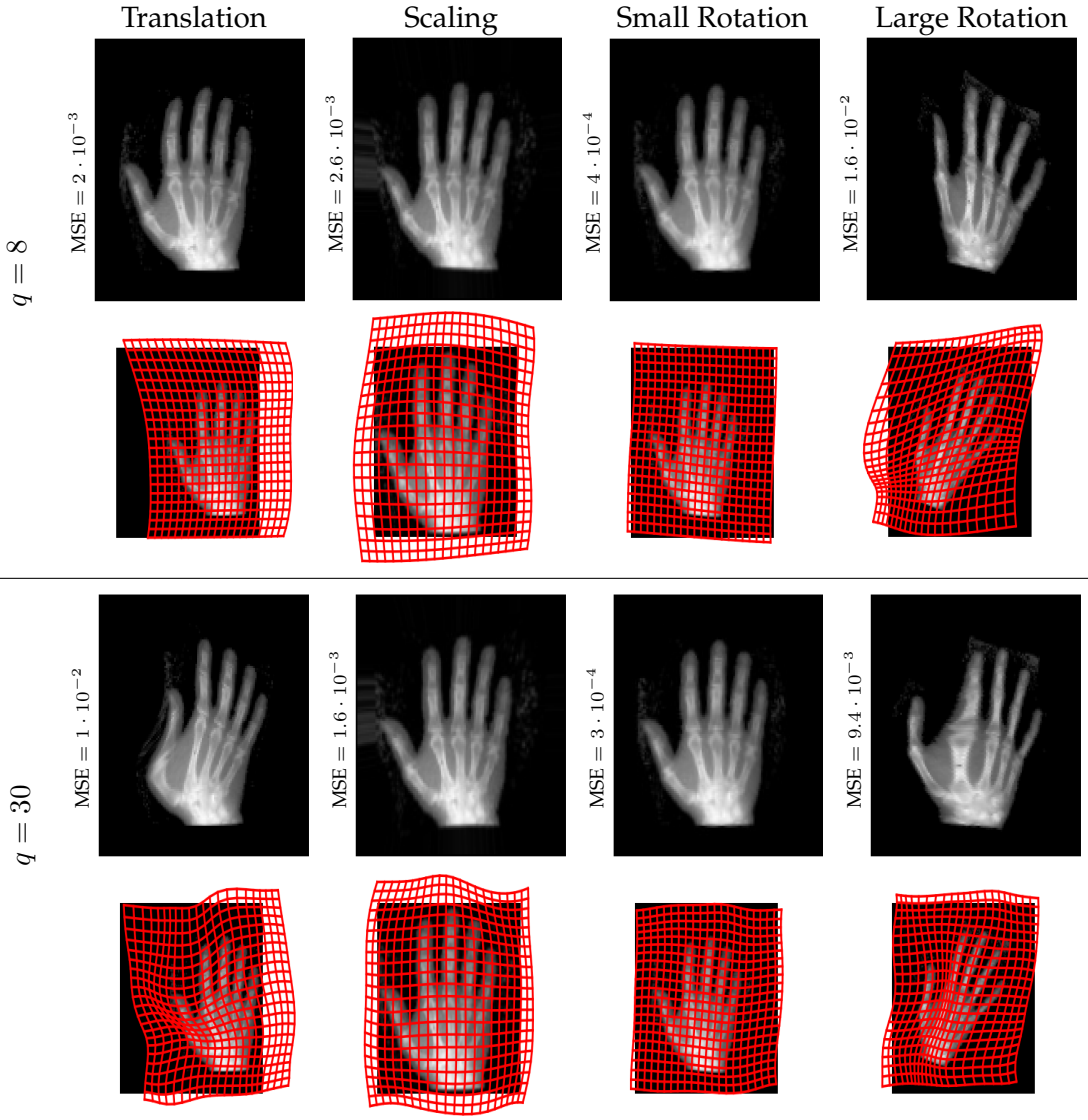


Figure 4.14: *Final deformed template image and deformed grid resulting from RNR using $N = 1$ time step; see Exp. 4.17–4.20. The MSE between the deformed template image and the reference image is specified on the left-hand side.* ***Top:*** *results using network width $q = 8$.* ***Bottom:*** *results using network width $q = 30$. For details about the visualization see Sec. 4.6.1.*

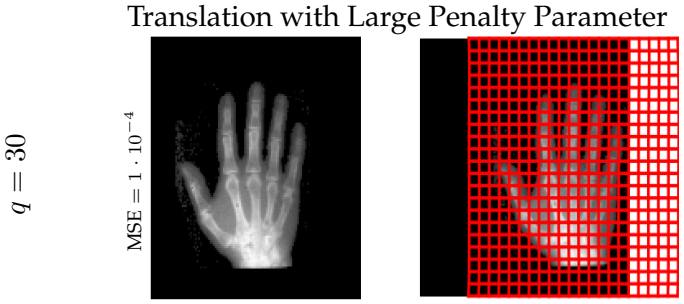


Figure 4.15: *Final deformed template image and deformed grid resulting from RNR using $N = 1$ time step, network width $q = 30$ and a large penalty parameter; see Exp. 4.17. The MSE between the deformed template image and the reference image is specified on the left-hand side. For details about the visualization see Sec. 4.6.1.*

ter by y_{L} and the one attained using the smaller regularization by y_{S}. The corresponding loss functions (29) minimized in RNR are denoted by J_{L} and J_{S}. The deformation y_{L} yields a much smaller MSE between the deformed template and the reference image than y_{S} ($1 \cdot 10^{-4}$ vs. $1 \cdot 10^{-2}$). Since additionally the deformed grid of y_{L} is much smoother, we expect $J_{\mathrm{S}}[y_{\mathrm{L}}] < J_{\mathrm{S}}[y_{\mathrm{S}}]$. This is indeed the case as $J_{\mathrm{S}}[y_{\mathrm{L}}] = 1.8 \cdot 10^{-4}$ and $J_{\mathrm{S}}[y_{\mathrm{S}}] = 1.4 \cdot 10^{-2}$. From this we conclude that y_{S} is at most a local minimum. An adapted optimization procedure could find the deformation in Fig. 4.15 instead of the one in Fig. 4.14. Nevertheless, we have shown that RNR is capable of calculating translations.

Experiment 4.18 (Scaling)**.** In the second experiment regarding the hand data, the template image is created by enlarging the reference image by a factor of 1.3; see second column of Fig. 4.13. The results corresponding to this experiment shown in the second column of Fig. 4.14 suggest that RNR is able to provide scalings numerically. In theory, the network within RNR is not able to represent scalings due to the sine activation function in the residual functions given by (27). It is therefore positively surprising that RNR is able to compute the scaling deformation of the hands numerically.
One potential explanation for this result could be the approximate linearity of the sine close to zero, which together with the outer operation in (27) could come close to a scaling. In the proof of Th. 4.1 a sequence of deformations was presented that converges to a scaling. Although the regularizer (31) prohibits that this sequence is produced within RNR, it may be possible that an element of it is achieved that already comes close.

Experiment 4.19 (Small rotation)**.** Next, the template is the reference image rotated by five degrees; see third column of Fig. 4.13. The resulting final images and deformed grids are shown in the corresponding column in Fig. 4.14. The deformed template image matches the reference well when using both the small ($q = 8$) and the medium network width ($q = 30$). With regard to the deformed grid there is a small difference between the two variants. While the small network yields a deformed grid that visually corresponds to a rotation, this is not the case for the larger network. The latter is not unexpected since rotations cannot be represented exactly by the network in RNR due to the sine activation function in (27). However, analogous argumentation as for the scaling experiment could explain that the result for $q = 8$ comes very close to a pure rotation. Since for $q = 30$ the network contains more parameters, there is a higher possibility to converge to another local minimum. This in turn could explain the result obtained for $q = 30$.

Nevertheless, for both utilized network widths the deformed template image corresponds with the reference, i.e. the rotation of the hand is captured.

Experiment 4.20 (Large rotation with nonlinear component)**.** In the last experiment in this section, a larger rotation by 22.5 degrees is considered. This example comes again with the difficulty of local minima. Also note that the template image differs from the reference image due to a further transformation in addition to the pure rotation, since the images show different hands; see fourth column of Fig. 4.13. The registration results are shown on the right of Fig. 4.14. With both the small and the medium network width ($q = 8, 30$) the optimization process within RNR gets stuck in a local minimum (overlap of index finger with middle finger). Therefore in this experiment RNR is not able to calculate the large rotation. This means alternative approaches are required, such as affine preregistration [11]. We can make no statement about the nonlinear component in this experiment because not even the dominating rotation is achieved.

4.6.4 Plausibility Checks

In this section, three checks on the plausibility of RNR are presented. First, we verify that the optimization within RNR is stable with respect to the initialization of the network parameters. Second, we examine the impact of the regularization and penalty parameter. Finally, we show that the choice for the number of epochs proposed in Sec. 4.5 is a reasonable termination condition for the optimization within RNR for the previous experiments. While we use exemplary the 3D-C example for the first two checks, the termination condition is validated for both the 3D-C example and the hand examples.

Experiment 4.21 (Stability)**.** In order to validate the stability of RNR with respect to the initialization, we will exemplary take another look at one of the registrations of the 3D-C example. More precisely, we base our validation on the registration using RNR with $N = 5$ time steps and network width $q = 30$ including all reference data; see Exp. 4.13. This setting yields a deformation that is able to map a ball to the desired C-shape.
For the experiments presented in the previous sections the network parameters were initialized with values drawn from a normal distribution with zero mean, using a standard deviation of one for the first two linear layers in each residual function and a standard deviation of 10^{-3} for each last layer; see Sec. 4.5.2. With this initialization the initial deformation is close to the identity; see Fig. 4.16 top, left. We repeat the registration using larger standard deviations of $5 \cdot 10^{-3}$ and 10^{-2} for the last linear layers in each residual function, which yield initial deformed grids differing strongly from the identity transformation; see Fig. 4.16 top. The final deformed template images and deformed grids depicted in the bottom of Fig. 4.16 show a small change in the final deformed grid compared to the change in the initialization. From this we conclude that the optimization within RNR is stable with respect to parameter initialization.

Experiment 4.22 (Impact of regularization and penalty parameter)**.** Next, we validate the impact of the regularization parameter α_S and the penalty parameter α_P on the registration quality. This analysis provides information about how sensitive the optimization within RNR is to the choice of these parameters. For this, we consider again the 3D-C example with $N = 5$ time steps including all reference data, but use the network width $q = 100$ now. We repeat the registration using the the values $\alpha_S = 10^{-6}, 10^{-4}, 10^{-2}, 1$

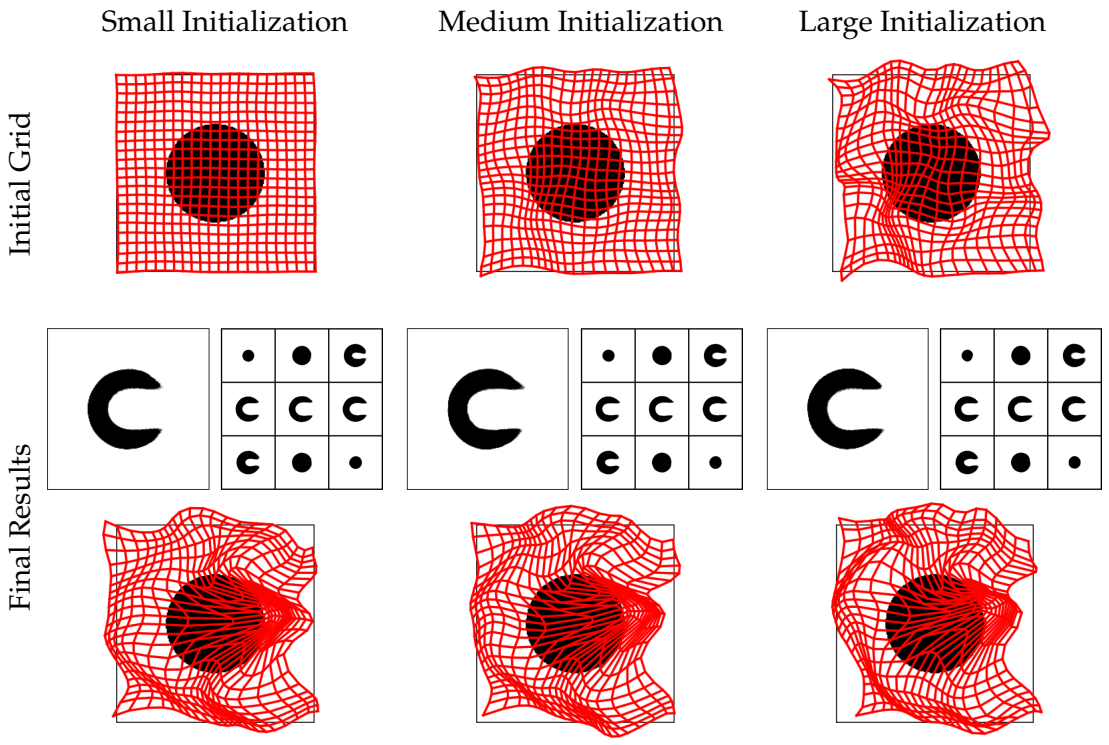


Figure 4.16: *Final deformed template image and deformed grid resulting from RNR for different initializations of the network parameters; see Exp. 4.21. Exemplary the 3D-C example with $N = 5$ time steps and network width q = 30 including all intermediate data is considered; see also Exp. 4.13. The initializations differ by the standard deviation of the parameters in the last layer of the residual functions (small: deviation $= 10^{-3}$, medium: deviation $= 5 \cdot 10^{-3}$, large: deviation $= 10^{-2}$). **Top:** initial grids. **Bottom:** final deformed templates image and deformed grids. For details about the visualization see Sec. 4.6.1.*

for the regularization parameter. With this choice we cover a wide range of values, including $\alpha_S = 10^{-2}$, which was utilized in the previous experiments in this section. For each choice of α_S the penalty parameter α_P is varied from zero to one with a step size of 0.02. Fig. 4.17 shows the resulting values for the data fitting term MSE and the smoothness penalty P after registration. In order to assess the results, note that the corresponding registration results presented in Exp. 4.13 were obtained using $\alpha_S = 10^{-2}$ and $\alpha_P = 2 \cdot 10^{-2}$ and the initial MSE without registration is $8 \cdot 10^{-3}$. The graphs of the MSE for $\alpha_S = 10^{-6}, 10^{-4}, 10^{-2}$ are almost identical, while for $\alpha_S = 1$ the MSE increases faster with α_P. The graph of the smoothness term hardly differs for all values of α_S. This implies that RNR is not sensitive to α_S in the large interval $[10^{-6}, 10^{-2}]$. The impact of the penalty parameter α_P is almost linear for the MSE, but exponential for the penalty P. Thus, for α_P close to zero, a slight increase of α_P has not much influence on the MSE, but significantly improves the smoothness of the deformation.

Experiment 4.23 (Convergence)**.** As discussed in Sec. 4.5.2, classical stopping criteria such as no significant change in the loss or reaching a certain threshold for the loss, are not suitable for the optimization within RNR. This is due to the large stochastic variations in the loss; see Sec. 4.5.2 for more details. Therefore, we simply fixed the total number of epochs to terminate the optimization. The disadvantage of this stopping cri-

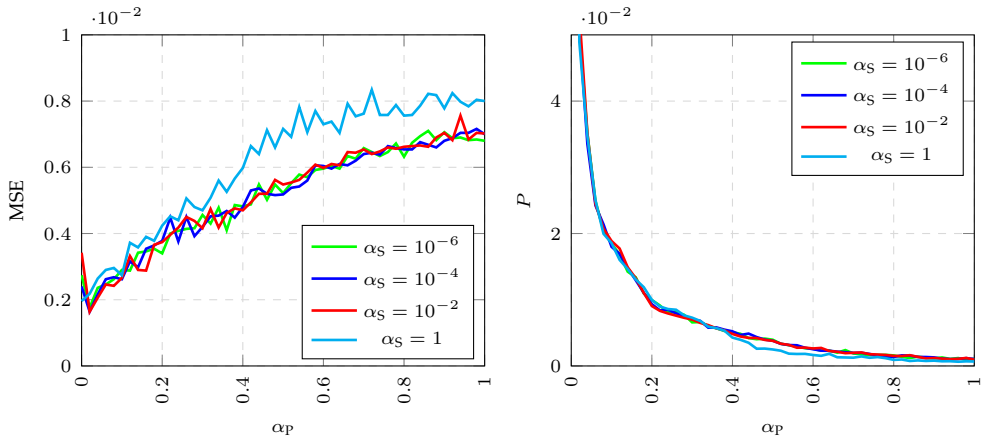


Figure 4.17: *Data fitting term MSE and penalty term P in RNR depending on the regularization parameter α_S and the penalty parameter α_P; see Exp. 4.22. Exemplary the 3D-C example with $N = 5$ time steps and network width $q = 100$ including all intermediate data is considered; see also Exp. 4.13.* ***Left:*** *The MSE increases almost linearly in α_P and is very similar for $\alpha_S = 10^{-6}, 10^{-4}, 10^{-2}$. The initial MSE without registration is $8 \cdot 10^{-3}$.* ***Right:*** *The smoothness penalty P decreases exponentially with α_P and is almost identical for all utilized values of α_S.*

terion is that the iteration number in an optimization procedure is not meaningful for the progression of the optimization. A reasonable choice for the total number of epochs depends on the considered problem. If the total number of epochs is too small, this can significantly reduce the quality of the final result, while choosing the number of epochs to large leads to a waste of computation time. Nonetheless terminating the optimization after a fixed number of epochs is a common approach in learning-based registration methods, since this stopping condition is not influenced by stochastic fluctuations of the loss; see e.g. [34, 35]. In each of the experiments presented up to this point, we decided to perform the optimization for 300 epochs. This number was empirically determined to be reasonable. The plots of the loss functions of exemplary experiments depicted in Fig. 4.18 confirm this choice.

On the left is the loss function for two experiments within the 3D-C example discussed in Sec. 4.6.2, one with the smallest network and one with the largest network. The oscillations in the graphs are due to the fact that only a small set of image data was used in each optimization step. In both cases shown, the loss function appears to have converged. Note that the red graph is close to constant except for the noise. This is because the graph belongs to the experiment with the smallest network in which the resulting deformation is almost the identity; see top left of Fig. 4.8.

On the right of Fig. 4.18 a loss is shown for each experiment type with the hand data from Sec. 4.6.3, where only the loss functions using the medium sized network ($q = 30$) are depicted for clarity. In contrast to the 3D-C example, in the hand examples all image data was utilized in each step, which results in smooth loss functions. The depicted graphs indicate that the loss functions have converged.

The behavior of the loss functions in the experiments discussed in the previous sections that are not shown in Fig. 4.18, is very similar to those depicted here.

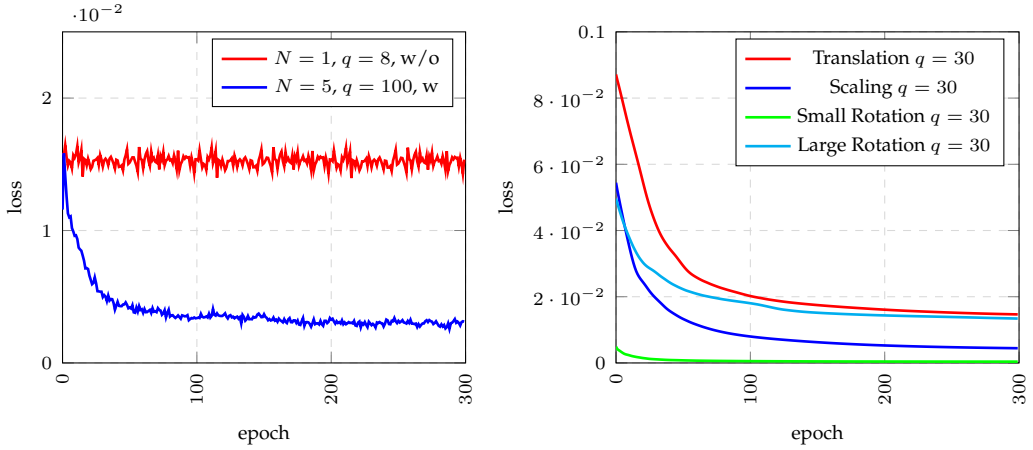


Figure 4.18: *Loss during the optimization epochs performed in RNR; see Exp. 4.23.* ***Left:*** *loss for two exemplary experiments within the 3D-C example, one with the smallest used network and one with the largest; see Exp. 4.11 and Exp. 4.13.* ***Right:*** *loss for four experiments using the hand data; see Exp. 4.17–4.20. For each type of the experiments the loss for the medium sized network ($q = 30$) is shown.*

4.6.5 Example Lung CT: Comparison to State-of-the-Art Methods

In the previous sections, we presented experiments on academic registration problems. These structured examples have enabled us to examine individual properties of RNR. In this section, we apply RNR on the publicly available DIR-Lab 4D CT dataset [131]. This dataset has been widely used to evaluate image registration methods. We show that RNR can compete with current state-of-the-art registration approaches regarding the registration quality on this dataset.

The DIR-Lab 4D CT dataset [131] provides ten thoracic 4D CT images acquired as part of a radiotherapy planning process for the treatment of thoracic malignancies. In order to compare the results generated by RNR with other registration methods such as [38], we model the inhalation movement. Therefore, we only use the images from the DIR-Lab 4D CT dataset that were acquired between the end-expiratory and the end-inspiratory phases, resulting in five 3D images per 4D scan. In each of the ten cases, we use the expiration image as the template and the inspiration image as the final reference. Thus, the goal is to map the expiration to the inspiration image; for a visualization of an exemplary case see Fig. 4.19.
The images vary in size and resolution. They contain 256×256 or 512×512 voxels in the axial plane, with a resolution between $0.97\,\mathrm{mm}$ and $1.16\,\mathrm{mm}$. The voxel thickness is $2.5\,\mathrm{mm}$. On average each 3D scan consists of 100 of these slices.
As it is common practice [38, 132], we optimize the deformation only for points within a lung mask of the inspiration image. This mask is obtained using the deep learning-based lung segmentation algorithm of Hoffmanninger et al. [133].
Our quantitative evaluation is based on the anatomical landmarks included for the extreme phases in the DIR-Lab 4D CT dataset. Each expiration-inspiration image pair contains 300 manually annotated corresponding landmarks in the lung. We will calculate the TRE, see (9), resulting from registration with RNR, as is standard practice for this dataset; see e.g. [38, 60]. This allows the results to be compared with state-of-the-art reg-

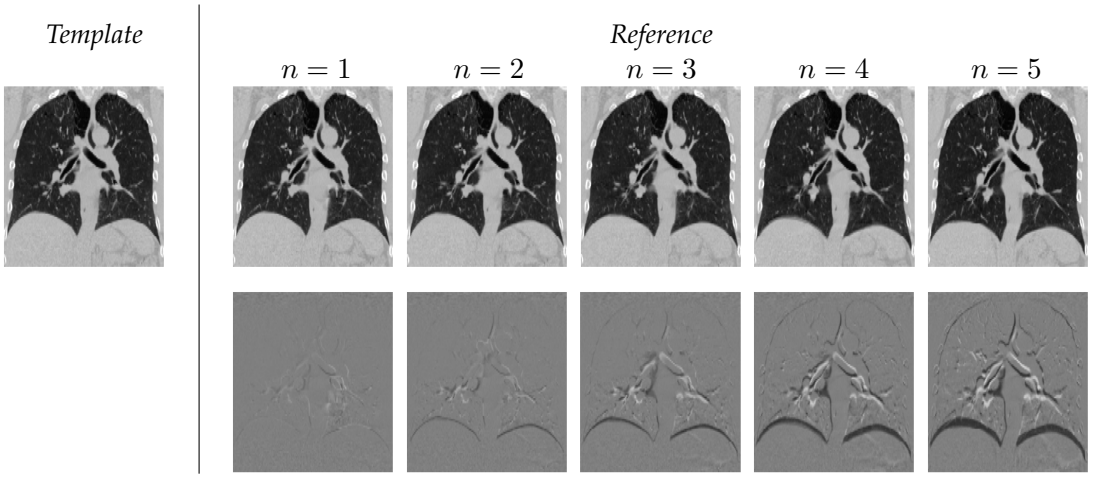


Figure 4.19: *Initial setting for case two from the DIR-Lab 4D CT dataset [131]. Coronal slices of a 3D template image, 3D reference images and their difference images are shown. For details about the visualization see Sec. 4.6.1.*

istration methods. The deformations corresponding to intermediate time points cannot be evaluated in this way, as no landmarks are available for these. Therefore, we will additionally present qualitative results; see Exp. 4.24 for details.

Choice of the Network Structure and the Penalty Parameter

In order to represent the thorax deformation between the end-expiration and the end-inspiration data, we choose the network structure that turned out to be the most flexible in the previous sections: RNR with network width $q = 100$. We use $N = 5$ time steps as this matches the number of images contained in the DIR-Lab 4D CT dataset.

Since choosing a learning rate of 10^{-3}, as described in Sec. 4.5.2, turned out to be too large in some of the ten cases of the dataset, we automatically select an appropriate learning rate for each case individually. Likewise the parameter for the smoothing penalty is chosen automatically for each case. More precisely, we perform a grid search for the learning rate and the penalty parameter. For this, we only use a coarse grid that covers a large range, as the aim of this section is to show that RNR provides comparable results to other registration methods and the focus is not on finding the best possible results using RNR. Based on experiments, the considered ranges are $[10^{-4}, 10^{-3}]$ for the learning rate and $[10^{-3}, 10]$ for the penalty parameter. Logarithmic sampling is performed for both, with only three values used for the learning rate and nine for the penalty parameter. The parameters selected are those that provide the best TRE on the landmarks from the dataset; see Tab. A.3 in the Appendix for a summary of the resulting values.

Experiment 4.24 (Comparison to state-of-the-art methods on the DIR-Lab 4D CT data)**.** For each case of the DIR-Lab 4D CT dataset, we perform two registrations using RNR, one without the intermediate reference data and one including all five reference images. Tab. 4.3 lists the TRE, see (9), obtained with both registrations using RNR as well as selected state-of-the-art registration methods. The selection of the methods used for comparison is based on [38], so we compare RNR with: a classical method [126], an INR-based approach [38], a CNN-based one-shot method [134] and a CNN-based approach [60] that, to the best of our knowledge, is the most accurate published approach

Table 4.3: *Mean and standard deviation of the TRE in* mm *for each case in the DIR-Lab 4D CT dataset [131] using two variants of RNR and state-of-the-art registration methods: RNR without intermediate data (RNR w/o) with* $N = 5$ *and* $q = 100$*; RNR with intermediate data (RNR w) with* $N = 5$ *and* $q = 100$*; a classical optimization-based registration approach [126]; an INR-based registration method [38]; a CNN-based one-shot method [134] and a CNN-based registration approach [60]; see Exp. 4.24.*

Case	Initial	RNR w/o	RNR w	Classical [126]	INR [38]	Shot [134]	CNN [60]
01	3.89±2.78	0.84±0.93	0.83±0.95	0.76±0.90	0.76±0.94	1.21±0.88	0.99±0.47
02	4.34±3.90	0.76±0.92	0.81±0.95	0.77±0.89	0.76±0.94	1.13±0.65	0.98±0.46
03	6.94±4.05	0.92±1.06	1.02±1.08	0.90±1.05	0.94±1.02	1.32±0.82	1.11±0.61
04	9.83±4.85	1.32±1.38	1.39±1.23	1.24±1.29	1.32±1.27	1.84±1.76	1.37±1.03
05	7.48±5.50	1.44±1.48	1.57±1.53	1.32±1.36	1.23±1.47	1.80±1.60	1.32±1.36
06	10.89±6.96	1.20±1.07	1.24±1.07	0.85±0.89	1.09±1.03	2.30±3.78	1.15±1.12
07	11.03±7.42	1.30±1.07	1.36±1.12	0.80±1.28	1.12±1.00	1.91±1.65	1.05±0.81
08	14.99±9.00	1.44±1.30	1.55±1.46	1.34±1.93	1.21±1.29	3.47±5.00	1.22±1.44
09	7.92±3.97	1.24±1.00	1.31±1.04	0.92±0.94	1.22±0.95	1.47±0.85	1.11±0.66
10	7.30±6.34	1.27±1.06	1.31±1.17	0.82±0.89	1.01±1.05	1.79±2.24	1.05±0.72
Mean	8.46	1.17	1.24	0.95	1.07	1.83	1.14

on the DIR-Lab data using deep learning to predict the deformation. The idea of one-shot learning in registration is to start with an untrained model and optimize the network parameters until convergence based only on the images to be registered. The results for the state-of-the art methods in Tab. 4.3 are taken from the original papers.
On average the TRE of RNR is $1.17\,\mathrm{mm}$ when omitting the intermediate reference data and $1.24\,\mathrm{mm}$ when using these. Thus, RNR falls slightly behind the conventional [126], the INR-based [38] and the CNN-based method [60] and outperforms the CNN-based one-shot method [134]. For the test cases one to four of the DIR-Lab dataset, RNR yields some of the best results. In the other cases, RNR performs slightly worse and cannot compete with the best results. Overall, however, from Tab. 4.3 we conclude that the results of RNR are comparable to these of the considered state-of-the-art methods. Moreover, we expect that the results of RNR can be further improved if more effort is put into the choice of the learning rate and the penalty parameter, instead of only performing the coarse parameter search as in this experiment.
It is noticeable in Tab. 4.3 that the TRE obtained from RNR with intermediate data is almost invariably higher than the one obtained from RNR without the additional data. On the one hand, this is surprising, as one might expect that the integration of the intermediate data would provide more information about the thoracic movement and thus better guide the registration. On the other hand, including more reference data means that during registration the similarity measure is minimized not only for the images belonging to the last time point, but also for the images in-between. Thus, the focus is less on the final time point, which is used to calculate the TRE. This could explain the worse TRE when using the intermediate reference data.
However, calculating the final TRE does not provide any information about how plausible the determined deformation is. Therefore, Fig. 4.20 visualizes the difference image between the reference image and the corresponding deformed template image for each time point using RNR without and with the intermediate reference data. More precisely, coronal slices of the difference images are depicted there for the exemplary case that was already used for the visualization of the data in Fig. 4.19. Since in the state-of-the-art methods above no intermediate results are computed, analogous visualization to evalu-

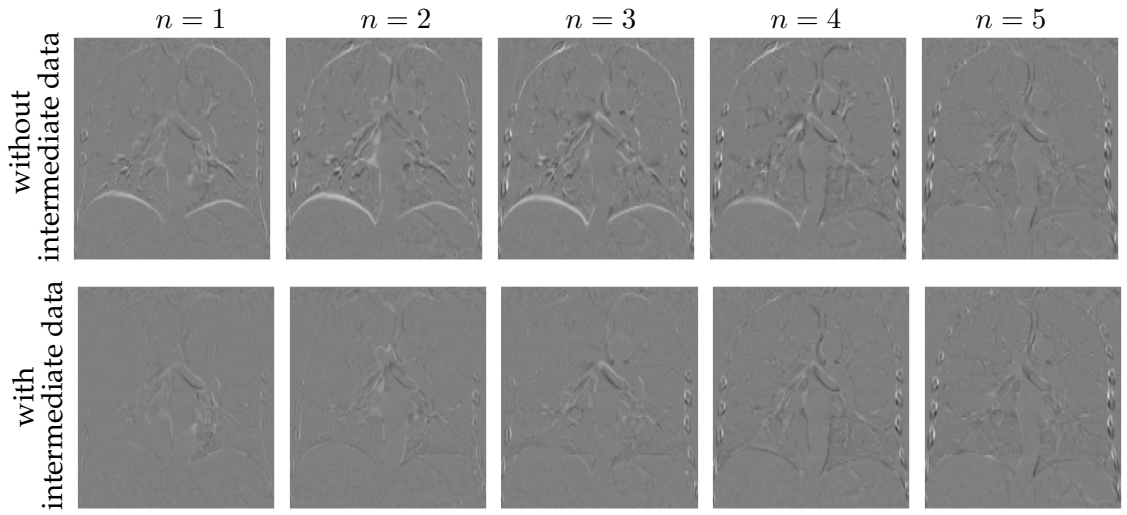


Figure 4.20: *Coronal slices of difference images between reference and deformed template image after each time step for an exemplary case of the DIR-Lab 4D CT dataset [131]. For registration RNR with $N = 5$ time steps and network width $q = 100$ is used. The difference images show that including the intermediate data significantly improves the ability of the deformation to represent lung movement; see Exp. 4.24. For details about the visualization see Sec. 4.6.1.*

ate the reasonability of the deformations is not possible. Fig. 4.20 shows that the deformation resulting from RNR is more realistic when the intermediate data is used as the lung movement is better captured in this case. This is relevant for applications where realistic deformations are more important than a good match of the images at the last time point as for example in the application presented in Sec. 4.7.

4.6.6 Summary of the Validation

We summarize the main results obtained during the validation of RNR.

First in Sec. 4.6.2, we validated RNR on the academic 3D-C example, whose structure allows to understand how the deformations resulting from RNR look like. We found that RNR can generate large deformations, which results in high registration quality. These deformations are capable of mapping the significantly different images of the ball and C-shape onto one another. The corners in the C-shape can also be represented well. This is surprising since RNR provides deformations in $C^\infty(\mathbb{R}^3, \mathbb{R}^3)$ that cannot converge to non-differentiable deformations either; see Sec. 4.3. Depending on the application, this property can be considered advantageous or disadvantageous.
A comparison between different network widths showed that a relative small network width of $q = 30$ is already reasonable for the considered structured example. Of course, the flexibility of RNR increases with the number of parameters in the underlying network. We have found that increasing the number of parameters by increasing the number of time steps has a greater effect than increasing the network width. However, even the largest network using $q = 100$ leads to relatively few network parameters compared to other INR-based registration methods.
For the 3D-C example, including the intermediate reference data had a great impact on the deformed template images at intermediate time points. At the same time in most

cases the registration result corresponding to the last time point was not visibly distinguishable from the result without the intermediate data.
Furthermore, we verified that the adaptation of the residual functions compared to the original residual functions used by Ben Amor et al. [41] indeed yields larger deformations. A comparison of RNR with an implementation of the classical LDDMM approach using the same number of time steps as RNR showed that RNR yields larger deformations than the LDDMM method. By increasing the number of time steps in the LDDMM approach, the method can also achieve the flexibility of RNR, but this involves significantly higher computational effort.

Second in Sec. 4.6.3, we considered different hand examples, which allowed to validate whether RNR is able to generate simple transformations such as translations, scalings and rotations. We showed that RNR can compute translations. The registrations of the scaled and the slightly rotated images were also performed well in our experiments, although these transformations cannot be represented exactly by deformations resulting from RNR. However, in the registration problem where the template and reference differ mainly by a large rotation, RNR fails. Here, RNR got stuck in local minima during the optimization, which is a common problem of registration approaches; see e.g. [22, 135].

Our plausibility checks in Sec. 4.6.4 showed that the optimization within RNR is stable with respect to the initialization of the network parameters. Furthermore, the impact of the regularization parameter on the registration proved to be small. In contrast, the penalty parameter, which mainly controls the smoothness of the deformation, had a large influence. Appropriate choices for the latter depend on the application.

Lastly in Sec. 4.6.5, the examination of the DIR-Lab 4D CT data allowed us to compare RNR with state-of-the-art registration methods on thoracic images. Here, RNR turned out to yield comparable results to the other methods with respect to the TRE for this data. Additionally, we found that including intermediate data slightly deteriorates the result of RNR corresponding to the last time point, but significantly improves the results corresponding to the intermediate time points. This is particularly advantageous for applications where calculation of realistic deformations is more important than a good match of the images at the last time point.

All in all, RNR proved to be a useful registration approach in order to compute multiple large and reasonable deformations that describe the motion of an object. In other words, RNR can be used to solve Problem 2.

4.7 Breathing Model Based on RNR

To highlight the power of RNR, we now apply the method to improve the modeling of sEMG signals. Respiratory sEMG modeling is a highly relevant field that contributes, for example, to improving mechanical ventilation for patients [5]. In order to model respiratory sEMG signals, the breathing movement itself must first be modeled. We will use RNR to create such a breathing model based on 3D and 2D MRI data; see Fig. 4.21.

First, in Sec. 4.7.1, we motivate respiratory sEMG modeling. We further explain how we use RNR to generate a dynamic breathing model that improves respiratory sEMG modeling. Afterwards, in Sec. 4.7.2 we introduce the data that is provided to compute the dynamic breathing model. The implementation details for the calculation of this model are presented in Sec. 4.7.3 and the resulting model itself is discussed in Sec. 4.7.4.

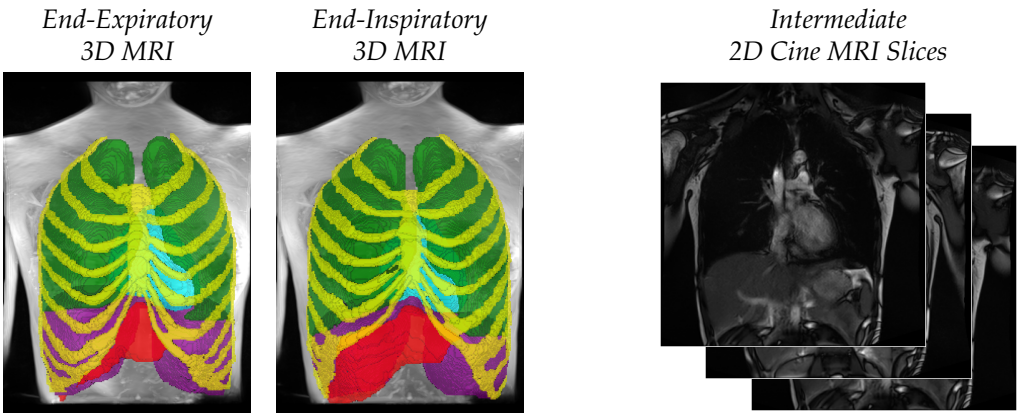


Figure 4.21: *Part of the data used for generation of a breathing model that forms the basis for respiratory sEMG modeling.* ***Left:*** *end-expiratory and end-inspiratory 3D MRI images with segmented torso parts that are relevant for respiratory sEMG modeling (diaphragm (purple), lungs (green), heart (cyan), liver (red), rib cage (yellow)).* ***Right:*** *2D Cine MRI slice images taken during breathing. See Sec. 4.7.2 for further details about the data.*

Finally, in Sec. 4.7.5 we summarize the advantages and disadvantages of using RNR to calculate a dynamic breathing model. Moreover, we provide ideas on how the breathing model and thus respiratory sEMG modeling can be further improved.

4.7.1 Respiratory sEMG Modeling

Respiratory sEMG modeling is a valuable tool to better understand the underlying physiology of breathing [5]. Respiration is caused by muscle contraction that generates a flow of air into and out of the lungs. As sEMG measures the electrical muscle activity resulting from muscle contraction non-invasively on the skin, it is a useful method for quantifying the respiratory effort [136]. Modeling these signals allows, for example, to test novel signal processing algorithms and to gain more insight into processes that take place during breathing. One particular application of respiratory sEMG modeling is the improvement of mechanical ventilation [137].
In the past, sEMG models for static, isometric muscle contractions were developed [138]. However, such models are not suited for representing respiratory sEMG signals, as breathing is a dynamic process. Thus, dynamic respiratory sEMG modeling is required.

A promising approach for developing such a model was proposed by Oltmann et al. [5]. This method is based on knowledge of the torso deformations at a finite number of time points during the breathing cycle. The acquisition of 3D torso MRI data during breathing is an error-prone task; see e.g. [56, 139]. Therefore, we use RNR to generate a dynamic breathing model that contains the needed torso deformations, without relying on intermediate 3D data. RNR is well suited for this purpose, as the method allows to compute multiple realistic deformations corresponding to successive time points. In the respiratory sEMG model approach [5] the deformations of specific torso parts between end-expiration and end-inspiration are required. The parts relevant for this are the diaphragm, the lungs, the heart, the liver, the rib cage and the entire thorax itself. Thus, we use RNR to calculate the deformation of these torso parts at a finite number of time points between end-expiration and end-inspiration.

4.7.2 Dataset

The data used to create the dynamic breathing model originates from an ongoing project together with the Fraunhofer IMTE on sEMG modeling and is currently not publicly available. The dataset consists of 3D and 2D MRI scans of the torso; see Fig. 4.21. The 3D images were recorded in the end-expiratory and end-inspiratory phase while the patient was holding the breath. The 2D images were taken during the breathing process. The holding of the breath was necessary due to the long scan time of 3D MRI; see e.g. [139]. Since the acquisition of 2D images is much faster than the acquisition of 3D images, the 2D images could be taken during breathing.
More precisely, the 3D data consist of T1-weighted MRI sequences recorded using the Dixon technique [140]. Thus, for both the end-expiratory and the end-inspiratory phase, four differently weighted 3D images (fat, water, in-phase, out-phase) are given. The 3D data consist of $250 \times 320 \times 128$ cubic voxels with a resolution of $1.3\,\mathrm{mm}$ each. The 2D data are T2-weighted TRUFI MRI sequences [141] corresponding to eight coronal and three sagittal slices. For each slice, 24 images were recorded with a repetition time of $422.91\,\mathrm{ms}$ for the coronal slices and $236.43\,\mathrm{ms}$ for the sagittal slices. The different perspectives and slice position were measured consecutively such that the obtained 2D slices correspond to multiple breathing cycles. The coronal slices consist of 512×512 square pixels with a resolution of $0.74\,\mathrm{mm}$, while the sagittal slices consist of 256×204 square pixels with a resolution of $1.48\,\mathrm{mm}$. The given 2D MRI scans are not flat but have a specific thickness of $10\,\mathrm{mm}$.
As the respiratory sEMG model is based on knowing the deformation of the specific torso parts described in Sec. 4.7.1, a semi-automatic data segmentation of these parts was generated in the 3D images with the software Materialise Mimics 24.0 [142]. Fig. 4.21 shows the segmented torso parts in an end-expiratory and an end-inspiratory MRI scan.

To build the dynamic breathing model we perform a registration with the RNR method. In the registration, we use the 3D expiration scans as template images, while the 3D inspiration images are the final reference images. In a preprocessing step, the 2D data are assigned to a finite number of time points between the end-expiratory and end-inspiratory phase. These are used as intermediate reference data. The segmentation provided is not utilized for the registration, but serves as a tool for the evaluation of the resulting dynamic breathing model.

4.7.3 Implementation Adjustments

Creating a breathing model based on the data presented in the previous section poses two challenges that are not addressed by the implementation of RNR from Sec. 4.5. Firstly, the data contains both 3D and 2D images, not just 3D images as was previously the case. Secondly, breathing involves opposite movements of different parts of the torso. While the rib cage moves upwards and outwards, the lungs, the diaphragm and the liver mainly move downwards [143, Ch. 3]. In this section, we show that only minor adjustments are required to adapt the implementation of RNR so that the method can overcome these challenges.

For this, we replace the energy functional (29) minimized in RNR with the functional

$$J[y] = D[y] + \alpha_{\mathrm{S}} S[y] + \alpha_{\mathrm{P}} P[y] + \alpha_{\mathrm{Q}} Q[y], \tag{36}$$

where the regularizer S as well as the smoothness penalty P are used as in Sec. 4.5. The data fitting term D as well as the penalty term Q, which is weighted by the parameter $\alpha_{\mathrm{Q}} > 0$, are chosen so that they can overcome the challenges mentioned above. We describe these terms in the following.
Based on the given data, the data fitting term consists of two parts, one to handle the 3D reference images and one for the 2D reference images, i.e. D is of the form

$$D[y] = D^{\mathrm{3D}}[y] + \alpha_{\mathrm{2D}} D^{\mathrm{2D}}[y]. \tag{37}$$

The first term ensures that the fully deformed template images match the 3D reference images. More precisely, in D^{3D} the similarity of the four differently weighted 3D image pairs is averaged. The second term measures the similarity between the given 2D images and the corresponding slices of the deformed template images at the corresponding time point. The parameter $\alpha_{\mathrm{2D}} \geq 0$ determines the weighting between the 3D and the 2D part in the data fitting term (37). For both the 3D and the 2D part, we use NCC as similarity measure in order to account for varying intensity changes in the data.
The penalty term Q addresses the difficulty that arises from the fact that breathing involves opposing movements in the torso. Without Q, we observed that RNR generates deformations that pull down parts of the ribs, as the ribs are located locally close to the lungs, which mainly move downward. Since this is an unrealistic movement of the rib cage, it should be prevented. Therefore, we penalize scalings of the rib cage in the vertical axis using for each time step

$$Q[\phi_n] := \frac{1}{|R \cap B|} \sum_{x \in R \cap B} \left\| \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} (\nabla \phi_n(x) - I_3) \right\|_2^2, \tag{38}$$

where $R \cap B$ is the set of voxels positions that are located within the segmented rib cage and are considered in the optimization step, analogous to (35). In (38), we assume that the vertical axis is oriented in the x_1-direction.
We choose the parameters in (36) and (37) as follows: We fix $\alpha_{\mathrm{2D}} = 0.5$ since with this choice the 3D and the 2D similarity terms before registration are of the same order of magnitude. The regularization parameter is set to $\alpha_{\mathrm{S}} = 10^{-4}$ as in the previous section; see Sec. 4.5 for a motivation of this choice. Reasonable values for the remaining parameters α_{P} and α_{Q} are determined by performing a logarithmic grid search. More precisely, for each parameter seven values are considered, which are in the range $[10^{-2}, 10]$ for α_{P} and in $[10^{-1}, 10^{2}]$ for α_{Q}. For the selection of the values both the quality of the final registration result and the intermediate deformations are taken into account, primarily by visual inspection. The resulting choices are $\alpha_{\mathrm{P}} = 3.16 \cdot 10^{-2}$ and $\alpha_{\mathrm{Q}} = 3.16$.

Besides the adaptation of the minimized loss term, we mainly retain the settings described in the general implementation of RNR; see Sec. 4.5. The only other difference is that now we restrict the validation to RNR with a large network width of $q = 100$ and $N = 5$ time steps. As shown in Sec. 4.6, this network width allows RNR to generate large deformations, which are reasonable for registration problems with real data. With the choice $N = 5$ we compute four deformations corresponding to intermediate time points between the end-expiratory and end-inspiratory phase, which are required for the respiratory sEMG model.
Additionally, since the similarity measure (37) is based not only on the 3D image voxels, but also on the pixels in the 2D data, the procedure described in Sec. 4.5 of taking

1% of the image voxels in each epoch is not directly applicable here. So, it remains to clarify what proportion of voxels and pixels is used in each optimization step: The 3D part in the data fitting term is evaluated on 10^5 voxels in each epoch, which corresponds to approximately 1% of the image voxels, as suggested in Sec. 4.5. The 2D similarity is averaged over 15 slices in each epoch, i.e. 5% of the 2D data. Hereby in each slice 10^3 pixels are considered, which corresponds to 0.5% of the pixels for a coronal slice and 2% for a sagittal slice.

4.7.4 The Breathing Model

We apply the RNR method, as described in Sec. 4.7.3, on the data presented in Sec. 4.7.2 to calculate a dynamic breathing model. More precisely, we compute the torso deformations at a finite number of time points between the end-expiratory and end-inspiratory phase. Using these deformations, we also obtain the deformed torso parts that are the basis for modeling respiratory sEMG signals.
Next, we evaluate the quality of the calculated breathing model. Note that, as usual with image registration, we do not have access to a ground truth deformation that can be used for this purpose.

First, we consider the quality of the registration result at the last time point. For this, we check how well the deformed torso parts important for the respiratory sEMG model in the end-expiratory image are mapped to the corresponding parts in the end-inspiratory image. As the segmentation of these torso parts is provided in both extreme phases, see Sec. 4.7.2, we calculate the Dice scores, see (10), between the corresponding segmentations. The resulting values for all relevant torso parts before and after registration are summarized in Tab. 4.4. As a Dice score of zero indicates no overlap and a value of one indicates perfect overlap, the performed registration achieves a significant improvement in the Dice value for each torso part. For the lungs, the heart, the liver and the thorax, the resulting Dice scores are around 0.9 after registration, which indicates a good alignment of these parts. In contrast, though much improved by the registration, the Dice scores of the diaphragm and rib cage fall significantly behind. However, as these are rather thin structures, already a small misalignment impacts the Dice score considerably. In addition, medical image segmentation is still a very challenging topic [74]. In general one can therefore not expect segmentations to be perfect. An examination of the lower region of the rib cage in Fig. 4.21 confirms this suspicion. Here, the segmentation of the rib cage in the end-expiratory and end-inspiratory image differ by an unrealistic deformation, which indicates an error in the segmentation. Thus, we do not aim for perfect Dice scores in Tab. 4.4 at all.

More important than a perfect overlap of the segmentations at the last time point is that the obtained deformations represent a physiologically realistic movement. Unfortunately, it is not possible to fully assess whether the generated breathing model is realistic.

Table 4.4: *Dice scores of different torso parts relevant for respiratory sEMG modeling before and after registration using RNR. A significant improvement in the Dice score for all torso parts is obtained by the registration; see Sec. 4.7.4 for more details.*

Dice	Diaphragm	Lung	Heart	Liver	Rib cage	Thorax
Before registration	0.176	0.838	0.849	0.677	0.246	0.965
After registration	0.546	0.960	0.926	0.882	0.529	0.986

This is because the breathing motion depends on many factors that are not included in the data, such as the pressure across the airway wall [143]. Nevertheless, we provide in the following at least a rough assessment of the plausibility of the determined breathing model. As there is no quantitative measure for the meaningfulness of a deformation, we assess this quantity on the basis of qualitative results presented in Fig. 4.22.
The first column in Fig. 4.22 shows coronal slices of the 3D difference images between the reference image corresponding to the last time point and the deformed template image corresponding to the respective time point. This visualization gives an overall impression of the computed torso movement. In addition, the difference image belonging to the last time point confirms the statement in Tab. 4.4 that the registration achieves a good overall alignment of the deformed expiratory and inspiratory images.
In order to assess the movement of the torso parts relevant for the sEMG model in more detail, the other three columns each show one deformed torso part over time. Exemplary we have chosen to show the lungs, the diaphragm and the rib cage as these are especially important for the respiratory movement [143]. Another reason for considering these parts is that, due to the opposite movements of the rib cage in contrast to the other torso parts, the registration is especially difficult at the interface between the rib cage and the lungs as well as between the rib cage and the diaphragm.
The second column in Fig. 4.22 shows coronal slices of the deformed template image with the deformed lung segmentation highlighted in red. For orientation also the reference lung segmentation is drawn in blue in the background. The images show that, as expected, the lungs move downwards and slightly outwards over time. Only at the lowest outer edge does the lung not move as far down as desired.
The third column of Fig. 4.22 shows an analogous visualization for the diaphragm. Since the diaphragm moves downward while maintaining its shape, we deem the movement realistic.
Depicting image slices is, in contrast to the lungs and the diaphragm, not useful for visualizing the rib cage. A realistic movement of the rib cage is characterized not only by the upwards and outwards movement. Important is also that the movement is uniform so that the individual ribs are rigidly deformed and no unnatural bending of the ribs occurs. These conditions cannot be verified in a 2D slice visualization analogous to the previous illustrations, which is why we show a 3D visualization of the rib cage in the last column of Fig. 4.22. Since 3D visualizations quickly become confusing when a lot of information is displayed, we concentrate on the deformed segmentation in the template image and do not show the segmentation of rib cage in the reference image. To provide orientation between the successive time points, white dashed lines are depicted in each image at the same level. From the visualization of the rib cage we conclude that the individual ribs are indeed deformed rigidly. A close look at the rib cage with the help of the white dashed lines shows the expected upward movement. While in the initial setting the third rib touches the white dashed line, at the last time point it is above the line. Since the outward movement of the rib cage is very small, it is not visible in the illustration. However, while writing this thesis we conducted further investigations, which are not presented here for the sake of brevity, but which showed that the outward movement is reasonably represented by the calculated deformation.
In summary, Fig. 4.22 shows that the deformations resulting from RNR represent a physiologically realistic movement of the torso parts that are important for respiratory sEMG modeling.

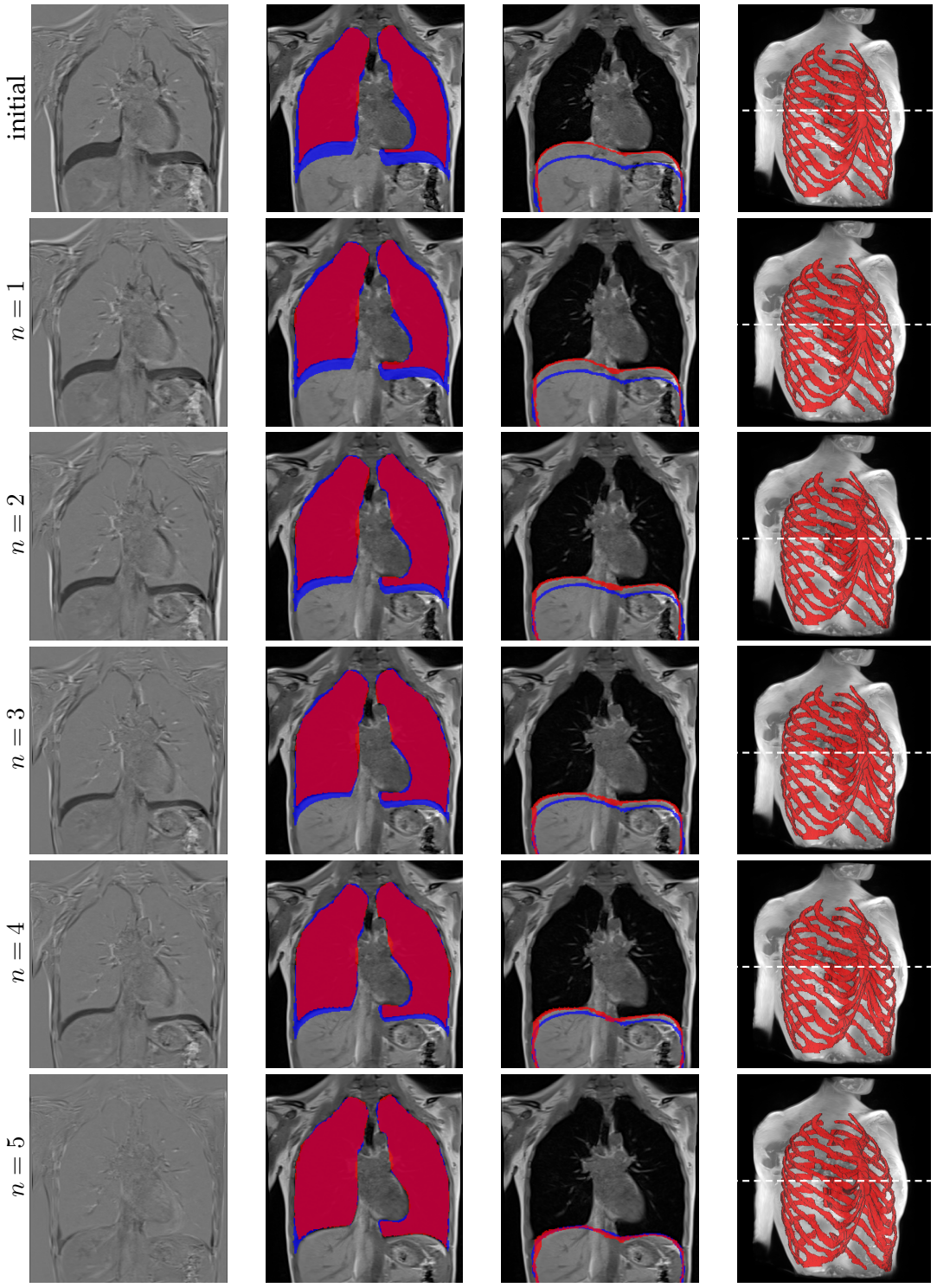


Figure 4.22: *Exemplary visualizations showing that the registration with RNR yields a realistic torso deformation over time; see Sec. 4.7.4. From left to right: coronal slices of the difference images between the final reference image and the deformed template image corresponding to the respective time step; coronal slices of deformed template image with lung segmentation (red: deformed template, blue: reference); coronal slices of deformed template image with diaphragm segmentation (red: deformed template, blue: reference); deformed 3D template with deformed rib cage segmentation (red).*

4.7.5 Summary of the RNR-Based Breathing Model

In the previous sections, we have used RNR to create a breathing model based on 3D and 2D MRI data. The motivation for this is the modeling of respiratory sEMG signals, an important field that contributes, for example, to improving mechanical ventilation for patients. We have shown that only minor adjustments are required to adapt the implementation of RNR to build the desired breathing model. It turned out that the deformations computed with RNR provide realistic movements of the torso parts that are important for sEMG modeling.
However, the match of some torso parts in the deformed expiration image at the last time point and the corresponding parts in the inspiration image could be further improved; see e.g. the lowest outer edge of the lungs in Fig. 4.22 or the Dice scores of the diaphragm and the rib cage in Tab. 4.4.
An approach that can potentially achieve this is to split the deformations into two different parts, one to represent the upward and outward movement of the rib cage and the other for the mainly downward movement of the other torso parts. This is a common procedure for the registration of torso images [1, 3, 144] that allows to better represent the disparate movements occurring during breathing. However, it should be noted that with this approach it is necessary to define which region is affected by each deformation. This task is generally carried out manually, which is time-consuming and error-prone; see e.g. [144]. Therefore, it remains for future work to verify whether including the approach of generating two different deformations into RNR actually improves the dynamic breathing model and thus respiratory sEMG modeling.

4.8 Conclusion

In this chapter, we introduced RNR, a new registration approach based on ResNets. We demonstrated that RNR is built on a solid theoretical foundation, validated the method in various experiments and presented an application of the method. The most important findings and ideas for future work are summarized in the following.

The motivation for RNR is the calculation of several deformations that describe the movement of an object at successive time points in a realistic way; see Problem 2. In order to solve this registration task, RNR uses an INR, i.e. the deformations are represented by a neural network. More specifically, a ResNet is utilized here, as this network architecture is connected to LDDMM, a popular approach for diffeomorphic image registration. Moreover, this network architecture enables the calculation of reasonable deformations by including data corresponding to intermediate time points into the registration.

In this chapter, we showed that RNR is capable of solving Problem 2. We proved that solutions to the minimization problem within RNR are guaranteed. In addition, we presented conditions under which the method provides diffeomorphic deformations.
In the validation, we found that RNR can generate large deformations, which can even map the significantly different images of a ball and a C-shape onto each other. Our experiments on a publicly available lung dataset showed that RNR can compete with current state-of-the-art registration approaches.
As a special highlight of this chapter, we used RNR to improve the modeling of sEMG signals. More precisely, we created a dynamic breathing model that forms the basis for modeling respiratory sEMG signals.

The advantages and disadvantages of our new registration method can be summarized as follows: The most important benefit of RNR is the inclusion of intermediate data in

the registration, which allows for realistic deformations to be calculated. Another advantage is the ability to compute large deformations, which are reasonable in registration problems where very different images are mapped onto each other. Furthermore, since RNR uses a neural network to parameterize the deformations, these deformations can be evaluated directly at any position. This means that RNR is a grid-free registration approach. Finally, we would like to emphasize that the implementation of RNR can easily be adapted to various applications. An example of a relevant medical application for which we used RNR is calculating a breathing model for respiratory sEMG modeling.
A disadvantage of RNR is that the inclusion of intermediate data in the registration can have a negative impact on the quality of the registration result at the last time point. Thus, the method is particularly suitable for applications in which the calculation of realistic time-dependent deformations is more important than a good match of the images at the last time point. A further drawback of RNR is that, apart from translations, no affine transformation can be represented by the utilized network structure. In addition, in some of the experiments we conducted, RNR got stuck in local minima during optimization. A disadvantage of the implementation of RNR is that it depends on many parameters, such as the learning rate and network width, which must be set appropriately to achieve reasonable registration results. However, this drawback is shared with other state-of-the-art registration approaches; see e.g. [38, 39, 60].

To conclude, we present ideas for future work, including approaches to remedy the latter disadvantages. Affine transformation could be made possible by adjusting the network structure, for example, by adding a linear part in the residual functions (27), such that

$$F_n^R(x) = W_3^n \sin(W_2^n \sin(W_1^n x + b_1^n) + b_2^n) + b_3^n + A^n x \quad \text{for } x \in \mathbb{R}^d$$

with $W_1^n, W_2^n, W_3^n, b_1^n, b_2^n, b_3^n$ as in (27) and $A^n \in \mathbb{R}^{d\times d}$.
In order to prevent the method from getting stuck in local minima, a multilevel strategy can be incorporated. The idea of multilevel approaches is to solve a series of discrete problems on a coarse-to-fine hierarchy of levels; see e.g. [68, 69]. One option to incorporate a multilevel strategy into RNR is to use a down-sampled version of the images at the beginning of the optimization and to gradually refine the image resolution during the optimization process. Another possibility is to realize a coarse-to-fine strategy for the representation of the deformation. This can be implemented by gradually increasing the number of network parameters used in the optimization, for which the unused parameters from the network are simply set to zero. To our knowledge, such multilevel strategies have not yet been incorporated into INR-based registration approaches.
Further ideas for future work include a more extensive validation of the impact of the individual network components. For instance other options for the activation function can be examined; see e.g. [106] for an overview of popular activation functions. In addition, RNR could be converted into a SVF approach, a popular and efficient adaptation of LDDMM, which is based on velocity fields that are constant in time; see e.g. [101]. The only adaptation necessary for this is to use shared weights between the layers corresponding to different time steps in the underlying network. This adaptation results in less network parameters, which goes along with less computational effort but also reduced flexibility. Future studies could validate the influence of this adjustment on computing time and registration quality.

Summarizing, RNR is a powerful new registration method with a wide range of possibilities for further investigation.

CHAPTER 5

Conclusion

In this thesis, we contributed to the field of image registration. We focused on two areas of research: the analysis of regularization in 3D/2D registration and the development of new registration approaches for the alignment of multiple consecutive images. For each of these topics, we briefly summarize our main results as well as ideas for further work. We refer to Sec. 3.6 and Sec. 4.8 for a more detailed discussion about our contributions and remaining questions.

First, we analyzed which type of regularization is appropriate for 3D/2D registration. As image registration is an inherently ill-posed problem, specifying suitable regularizers is one of its core questions [15]. A major difficulty with 3D/2D registration is that a higher dimensional deformation is sought from a lower dimensional image. However, in many applications, it is reasonable to assume that this lower dimensional data is representative for a higher dimensional domain in the surrounding area; see e.g. [50, 55]. Therefore, a meaningful class of deformations for such registration problems are continuously differentiable deformations, since these transfer information to a higher dimensional region.
For this reason, we analyzed which type of regularization ensures continuously differentiable deformations in 3D/2D registration. To the best of our knowledge, there had been no thorough investigation of this question prior to our work. Our main result regarding this topic is that specific second-order regularizers guarantee continuous differentiability (Th. 3.13), while first-order regularizers generally do not (Th. 3.15). In addition, we showed that the assumptions on which our proofs are based hold for commonly used data fitting and regularization terms (Sec. 3.4), and we presented extensions of the proofs to registration problems of other dimensionalities (Sec. 3.5).
Open questions on this topic include how the results can be extended if other properties are required for the deformation. For example, a higher regularity of the deformation could be assumed. In volume-to-slice registration, a special case of 3D/2D registration, it may also be useful to increase the space of suitable deformations by assuming that the deformations are only continuously differentiable in the direction perpendicular to the 2D slice.

In the second part of this thesis, we presented the new registration approach RNR that is based on residual neural networks (ResNets). This method enables the computation of multiple deformations that correspond to successive time points. Furthermore, RNR yields diffeomorphic deformations under certain conditions. This is a desirable property for many applications, as diffeomorphisms are invertible and preserve certain image features [29]. RNR can be used, for example, to build a breathing model, which in turn forms the basis for the modeling of respiratory surface electromyography (sEMG) signals. One of the applications of the latter is the improvement of mechanical ventilation for patients [5].
Our new method is based on large deformation diffeomorphic metric mapping (LDDMM), a popular approach for diffeomorphic registration; see e.g. [29]. In contrast, to classical LDDMM methods, we use ResNets for the representation of the deformations. This results in reduced computational effort compared to classical LDDMM methods. More precisely, our approach is based on the method of Ben Amor et al. [41], which also utilizes ResNets. However, our RNR method differs from it in terms of the underlying network architecture and the areas of applications for which it is suitable. Due to the adapted network structure, RNR yields larger deformations than the method in [41], which is beneficial as large deformations enable the registration of even very different images.
We showed that RNR has a sound theoretical basis and provided a comprehensive experimental validation. In particular, we proved that the chosen regularizer guarantees the existence of solutions (Th. 4.3). In addition, we showed that the analytical solution approximated by RNR is diffeomorphic (Th. 4.6), and we presented conditions that also guarantee diffeomorphisms for the discretized solution (Th. 4.7). We conducted various experiments to examine RNR (Sec. 4.6) and finally applied the method to build a breathing model, which can be used as a basis for respiratory sEMG modeling (Sec. 4.7).
Promising directions for future research include the adaptation of the network structure within RNR to exactly parameterize affine deformations, as these represent an important class of deformations [9], as well as the incorporation of a multilevel strategy to overcome the problem of local minima; see e.g. [69].

Summarizing, although image registration is still far from being fully solved, in this thesis we achieved a step towards a more comprehensive understanding of image registration problems. Beyond proving fundamental results to ensure meaningful solutions, we developed a new registration approach that is competitive with other state-of-the-art methods.

APPENDIX A

Parameters for RNR Validation

In this chapter, we list the parameter values chosen in the validation of RNR in Sec. 4.6.

Tab. A.1–A.2 summarize the values for the penalty parameter α_P used in the experiments within the 3D-C example and the hand examples; see Sec. 4.6.2 and Sec. 4.6.3. As discussed in Sec. 4.6.2, the idea is to choose α_P as small as possible, but large enough such that the resulting deformations are diffeomorphic. Therefore, in each experiment a grid search is performed, using values between 0 and 1 with 0.02 as step size. For each value for α_P, we consider the Jacobian determinants of the resulting deformations. More precisely, we selected the smallest value, such that the Jacobian determinants are greater than zero on a certain set of points; see Sec. 4.6.2 for more details.

Table A.1: *Values for α_P used in the experiments within the 3D-C example in Sec. 4.6.2: RNR without intermediate data (RNR w/o) with $N = 1, 5$ and $q = 8, 30, 100$; RNR with intermediate data (RNR w) with $N = 5$ and $q = 8, 30, 100$; RNR with the original residual functions [41] (RNR w+[41]), intermediate data, $N = 5$ and $q = 30$; torch-lddmm [129] with $N = 5, 25$.*

Exp.	RNR w/o						RNR w			RNR w+[41]	LDDMM	
N	1			5			5			5	5	25
q	8	30	100	8	30	100	8	30	100	30	-	-
α_P	0	0.02	0.1	0	0	0.4	0	0	0.02	0	0.56	0.02

Table A.2: *Values for α_P used in the experiments within the hand examples in Sec. 4.6.3. The registrations are performed using RNR with $N = 1$ time step, network width $q = 8, 30$ and without intermediate data.*

Exp.	Translation		Scaling		Small Rotation		Large Rotation	
q	8	30	8	30	8	30	8	30
α_P	0	0.02	0	0.02	0	0	0	0.02

Tab. A.3 shows the values for the learning rate and penalty parameter α_P that were used for the experiments with the DIR-Lab 4D CT dataset [131] in Sec. 4.6.5. Since the experiments considered there do not focus on choosing optimal parameter values, the grid search performed to select the values is based only on a coarse grid covering a large range of values. More precisely, the ranges are $[10^{-4}, 10^{-3}]$ for the learning rate and $[10^{-3}, 10]$ for the penalty parameter. Logarithmic sampling is performed with three values used for the learning rate and nine for the penalty parameter. The parameters selected are those that provide the best TRE on the landmarks from the considered dataset; see Sec. 4.6.5 for more details.

Table A.3: *Values for the learning rate and the penalty parameter α_P for all ten cases of the DIR-Lab 4D CT dataset [131] used for registration with RNR without and with intermediate data (RNR w/o, RNR w), $N = 5$ and $q = 100$; see Sec. 4.6.5.*

Exp.	Param.	01	02	03	04	05	06	07	08	09	10
RNR w/o	lr $[\cdot 10^{-4}]$	3	1	1	1	3	1	3	1	1	1
	α_P	0.32	10	3.16	10	3.16	3.16	10	3.16	0.01	3.16
RNR w	lr $[\cdot 10^{-4}]$	1	1	1	1	3	1	1	3	3	1
	α_P	10	1	0.32	0.32	3.16	3.16	3.16	1	0.003	0.32

Notation

General Notations

$\partial\Omega$	Boundary of Ω
$\bar{\Omega}$	Closure of Ω
$\lvert\Omega\rvert$	Cardinality of a finite set Ω
I_n	Identity matrix of size n
$\mathbb{N}$	Natural numbers (integers): $\{1, 2, 3, ...\}$
$\mathbb{R}^d$	d-dimensional Euclidean space

Function Spaces

$C(\Omega)$	Same as $C(\Omega, \mathbb{R})$
$C_c(\Omega)$	Same as $C_c(\Omega, \mathbb{R})$
$C^k(\Omega)$	Same as $C^k(\Omega, \mathbb{R})$
$C^\infty(\Omega)$	Same as $C^\infty(\Omega, \mathbb{R})$
$C^{k,\alpha}(\Omega)$	Same as $C^{k,\alpha}(\Omega, \mathbb{R})$
$C(\Omega, \mathbb{R}^n)$	Continuous func. from Ω to $\mathbb{R}^n$
$C_c(\Omega, \mathbb{R}^n)$	Compactly supported func. in $C(\Omega, \mathbb{R}^n)$
$C^k(\Omega, \mathbb{R}^n)$	k-times continuously differentiable func. from Ω to $\mathbb{R}^n$
$C^\infty(\Omega, \mathbb{R}^n)$	$\bigcap_k C^k(\Omega, \mathbb{R}^n)$
$C^{k,\alpha}(\Omega, \mathbb{R}^n)$	Hölder space: $C^k(\Omega, \mathbb{R}^n)$-func. with Hölder continuous k-th derivatives
$C_c^k(\Omega, \mathbb{R}^n)$	$C^k(\Omega, \mathbb{R}^n) \cap C_c(\Omega, \mathbb{R}^n)$
$C_c^\infty(\Omega, \mathbb{R}^n)$	$C^\infty(\Omega, \mathbb{R}^n) \cap C_c(\Omega, \mathbb{R}^n)$
$H^k(\Omega)$	Same as $W^{k,2}(\Omega)$
$H^k(\Omega, \mathbb{R}^n)$	Same as $W^{k,2}(\Omega, \mathbb{R}^n)$
$H_0^k(\Omega, \mathbb{R}^n)$	Same as $W_0^{k,2}(\Omega, \mathbb{R}^n)$
$H_{\text{loc}}^k(\Omega, \mathbb{R}^n)$	Same as $W_{\text{loc}}^{k,2}(\Omega, \mathbb{R}^n)$
$L^p(\Omega)$	Same as $L^p(\Omega, \mathbb{R})$
$L^p(\Omega, \mathbb{R}^n)$	Lebesgue p-integrable func. from Ω to $\mathbb{R}^n$
$W^{k,p}(\Omega)$	Same as $W^{k,p}(\Omega, \mathbb{R})$
$W^{k,p}(\Omega, \mathbb{R}^n)$	Sobolev space: func. with weak derivatives up to order k in $L^p(\Omega, \mathbb{R}^n)$
$W_0^{k,p}(\Omega, \mathbb{R}^n)$	Sobolev space with zero boundary: closure of $C_c^\infty(\Omega, \mathbb{R}^n)$ in $W^{k,p}(\Omega, \mathbb{R}^n)$
$W_{\text{loc}}^{k,p}(\Omega, \mathbb{R}^n)$	Local Sobolev func.: $\{f : f\lvert_A \in W^{k,p}(\Omega, \mathbb{R}^n)\ \forall A$ open, $\bar{A} \subset \Omega$ compact$\}$

X^{-1}	Topological dual space to X: bounded linear functionals on X

Operators and Functions

$\otimes$	Kronecker product
$d\mathcal{H}^n$	n-dimensional Hausdorff measure
$\partial_{x_j} f$	Partial derivative of f with respect to the variable x_j
$\frac{\partial f}{\partial v}, d_v f$	Directional derivative of f in direction v $\left(\frac{\partial f}{\partial v}(x) = \lim_{t\to 0} \frac{f(x+tx)-f(v)}{t}\right)$
$\operatorname{div} f$	Divergence of $f : \Omega \subset \mathbb{R}^n \to \mathbb{R}^n$ $\left(\operatorname{div} f = \sum_{j=1}^n \partial_{x_j} f_j\right)$
$D^\alpha f$	Partial derivative of f for multi-index α $\left(D^\alpha f = \frac{\partial^{\alpha_1+\cdots+\alpha_d} f}{\partial x_1^{\alpha_1} \ldots \partial x_d^{\alpha_d}}\right)$
∇f	Gradient/Jacobian of f
$\nabla^2 f$	Hessian of f
Δf	Laplacian of f $(\Delta f = \operatorname{div}(\nabla f))$
Id	Identity func.
$f\|_\Omega$	Restriction of f to Ω
$\operatorname{supp}(f)$	Support of f: closure of $\{x : f(x) \neq 0\}$
$\|x\|_2, \|x\|$	Euclidean norm of vector x
$\|x\|_\infty$	Maximum norm of vector x
$\|x\|_X$	Norm of x with respect to given normed space X
$\|A\|_2$	Spectral norm of matrix A
$\|A\|_F$	Frobenius norm of matrix A
$\|f\|_{L^p(\Omega)}$	Same as $\|f\|_{L^p(\Omega,\mathbb{R})}$
$\|f\|_{L^p(\Omega,\mathbb{R}^n)}$	L^p-norm of f $\left(\|f\|_{L^p(\Omega,\mathbb{R}^n)} = \left(\int_\Omega \|f(x)\|_{\mathbb{R}^n}^p \, dx\right)^{1/p}\right)$
$\|f\|_{H^k(\Omega)}$	Same as $\|f\|_{H^k(\Omega,\mathbb{R})}$
$\|f\|_{H^k(\Omega,\mathbb{R}^n)}$	Sobolev norm of f $\left(\|f\|_{H^k(\Omega,\mathbb{R}^n)} = \left(\sum_{\|\alpha\|\leq k} \|D^\alpha f\|_{L^2(\Omega,\mathbb{R}^n)}^2\right)^{1/2}\right)$
$\langle x, y\rangle_{\mathbb{R}^n}$	Euclidean scalar product
$\langle f, g\rangle_{L^2(\Omega)}$	Same as $\langle f, g\rangle_{L^2(\Omega,\mathbb{R})}$
$\langle f, g\rangle_{L^2(\Omega,\mathbb{R}^n)}$	Scalar product on $L^p(\Omega,\mathbb{R}^n)$ $\left(\langle f, g\rangle_{L^2(\Omega,\mathbb{R}^n)} = \int_\Omega \langle f(x), g(x)\rangle_{\mathbb{R}^n} \, dx\right)$
$\langle f, g\rangle_{H^k(\Omega,\mathbb{R}^n)}$	Scalar product on $H^k(\Omega,\mathbb{R}^n)$ $\left(\langle f, g\rangle_{H^k(\Omega,\mathbb{R}^n)} = \sum_{\|\alpha\|\leq k} \langle D^\alpha f, D^\alpha g\rangle_{L^2(\Omega,\mathbb{R}^n)}\right)$

Bibliography

[1] S. Neuber, P. F. Schulz, S. Kuckertz, and J. Modersitzki, "Segmentation-inspired image registration," in *BVM Workshop*, pp. 205–210, Springer, 2024.

[2] P. F. Schulz, F. Mannel, and J. Modersitzki, "Appropriate order of regularization in 3D/2D image registration," in *Proc. SSVM 2025*, pp. 377–390, Springer, 2025.

[3] O. Gildemeister, J. Bostelmann, P. F. Schulz, A. Oltmann, P. Rostalski, J. Modersitzki, and J. Lellmann, "Time-continuous sliding motion image registration using stationary velocity fields for respiratory motion interpolation," in *Proc. SSVM 2025*, pp. 404–416, Springer, 2025.

[4] P. F. Schulz, A. Oltmann, J. Bostelmann, O. Gildemeister, F. Wegner, J. Lellmann, P. Rostalski, and J. Modersitzki, "Image registration for a dynamic breathing model," in *BVM Workshop*, pp. 5–11, Springer, 2025.

[5] A. Oltmann, O. Gildemeister, B. Johannes, P. F. Schulz, J. Graßhoff, F. Wegner, A. Frydrychowicz, J. Lellmann, J. Modersitzki, T. Knopp, and P. Rostalski, "Simulation of diaphragmatic motor unit action potentials throughout the respiratory cycle using a dynamic breathing model," in *Proc. EMBC 2025*, accepted.

[6] D. Rueckert and J. A. Schnabel, *Medical image registration*, pp. 131–154. Berlin, Heidelberg: Springer, 2011.

[7] A. Sotiras, C. Davatzikos, and N. Paragios, "Deformable medical image registration: A survey," *IEEE Trans. Med. Imaging*, vol. 32, no. 7, pp. 1153–1190, 2013.

[8] J. Chen, Y. Liu, S. Wei, Z. Bian, S. Subramanian, A. Carass, J. L. Prince, and Y. Du, "A survey on deep learning in medical image registration: New technologies, uncertainty, evaluation metrics, and beyond," *Med. Image Anal.*, vol. 100, p. 103385, 2025.

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

[10] A. A. Goshtasby, *2-D and 3-D image registration: for medical, remote sensing, and industrial applications*. US: Wiley, 2005.

[11] J. Modersitzki, *FAIR. Flexible algorithms for image registration.*, vol. 6 of *Fundam. Algorithms*. Philadelphia, PA: SIAM, 2009.

[12] R. T. Chen, J. Behrmann, D. K. Duvenaud, and J.-H. Jacobsen, "Residual flows for invertible generative modeling," in *Proc. NeurIPS 2019*, vol. 32, Curran Associates, Inc., 2019.

[13] J.-M. Guyader, W. Huizinga, D. H. J. Poot, M. van Kranenburg, A. Uitterdijk, W. J. Niessen, and S. Klein, "Groupwise image registration based on a total correlation dissimilarity measure for quantitative mri and dynamic imaging data," *Sci. Rep.*, vol. 8, no. 1, p. 13112, 2018.

[14] Kai Brehmer, *SqN: A Novel Distance Measure for Multiple Images and Applications*. PhD thesis, University of Lübeck, 2021.

[15] B. Fischer and J. Modersitzki, "Ill-posed medicine - an introduction to image registration," *Inverse Prob.*, vol. 24, no. 3, p. 034008, 2008.

[16] J. B. Antoine Maintz and M. A. Viergever, "A survey of medical image registration," *Med. Image Anal.*, vol. 2, no. 1, pp. 1–36, 1998.

[17] B. Zitova and J. Flusser, "Image registration methods: a survey," *Image Vis. Comput.*, vol. 21, no. 11, pp. 977–1000, 2003.

[18] M. A. Viergever, J. B. Antoine Maintz, S. Klein, K. Murphy, M. Staring, and J. P. Pluim, "A survey of medical image registration–under review," *Med. Image Anal.*, vol. 33, pp. 140–144, 2016.

[19] K. Chen, L. M. Lui, and J. Modersitzki, "Image and surface registration," in *Processing, analyzing and learning of images, shapes, and forms. Part 2*, vol. 20, pp. 579–611, Amsterdam: Elsevier/North Holland, 2019.

[20] G. Haskins, U. Kruger, and P. Yan, "Deep learning in medical image registration: a survey," *Mach. Vis. Appl.*, vol. 31, no. 1, p. 8, 2020.

[21] Y. Fu, Y. Lei, T. Wang, W. J. Curran, T. Liu, and X. Yang, "Deep learning in medical image registration: a review," *Phys. Med. Biol.*, vol. 65, no. 20, p. 20TR01, 2020.

[22] B. Deng, Y. Yao, R. M. Dyke, and J. Zhang, "A survey of non-rigid 3D registration," in *Computer Graphics Forum*, vol. 41, pp. 559–589, Wiley Online Library, 2022.

[23] T. Duan, W. Chen, M. Ruan, X. Zhang, S. Shen, and W. Gu, "Unsupervised deep learning-based medical image registration: a survey," *Phys. Med. Biol.*, vol. 70, no. 2, p. 02TR01, 2025.

[24] J. V. Hajnal and D. L. Hill, *Medical image registration*. Boca Raton: CRC press, 2001.

[25] D. Fortun, P. Bouthemy, and C. Kervrann, "Optical flow modeling and computation: A survey," *Comput. Vis. Image Und.*, vol. 134, pp. 1–21, 2015.

[26] B. K. P. Horn and B. G. Schunck, "Determining optical flow," *Artif. Intell.*, vol. 17, no. 1-3, pp. 185–203, 1981.

[27] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. IJCAI 1981*, vol. 2, pp. 674–679, Morgan Kaufmann Publishers Inc., 1981.

[28] A. Joshi and Y. Hong, "R2net: Efficient and flexible diffeomorphic image registration using lipschitz continuous residual networks," *Med. Image Anal.*, vol. 89, p. 102917, 2023.

[29] M. Hernandez and U. R. Julvez, "Insights into traditional large deformation diffeomorphic metric mapping and unsupervised deep-learning for diffeomorphic registration and their evaluation," *Comput. Biol. Med.*, vol. 178, p. 108761, 2024.

[30] R. Jena, D. Sethi, P. Chaudhari, and J. Gee, "Deep learning in medical image registration: Magic or mirage?," in *Proc. NeurIPS 2024*, vol. 37, pp. 108331–108353, 2025.

[31] K. Han, S. Sun, X. Yan, C. You, H. Tang, J. Naushad, H. Ma, D. Kong, and X. Xie, "Diffeomorphic image registration with neural velocity field," in *Proc. WACV 2023*, pp. 1869–1879, 2023.

[32] X. Yang, R. Kwitt, M. Styner, and M. Niethammer, "Quicksilver: Fast predictive image registration–a deep learning approach," *NeuroImage*, vol. 158, pp. 378–396, 2017.

[33] M.-M. Rohé, M. Datar, T. Heimann, M. Sermesant, and X. Pennec, "Svf-net: learning deformable image registration using shape matching," in *Proc. MICCAI 2017*, pp. 266–274, Springer, 2017.

[34] G. Balakrishnan, A. Zhao, M. R. Sabuncu, J. Guttag, and A. V. Dalca, "Voxelmorph: a learning framework for deformable medical image registration," *IEEE Trans. Med. Imaging*, vol. 38, no. 8, pp. 1788–1800, 2019.

[35] J. Chen, E. C. Frey, Y. He, W. P. Segars, Y. Li, and Y. Du, "Transmorph: Transformer for unsupervised medical image registration," *Med. Image Anal.*, vol. 82, p. 102615, 2022.

[36] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," in *Proc. NIPS 2017*, 2017.

[37] T. C. Mok and A. Chung, "Fast symmetric diffeomorphic image registration with convolutional neural networks," in *Proc. CVPR 2020*, pp. 4643–4652, 2020.

[38] J. M. Wolterink, J. C. Zwienenberg, and C. Brune, "Implicit neural representations for deformable image registration," in *Proc. MIDL 2022*, pp. 1349–1359, PMLR, 2022.

[39] S. Sun, K. Han, C. You, H. Tang, D. Kong, J. Naushad, X. Yan, H. Ma, P. Khosravi, J. S. Duncan, *et al.*, "Medical image registration via neural fields," *Med. Image Anal.*, vol. 97, p. 103249, 2024.

[40] J. Bostelmann, O. Gildemeister, and J. Lellmann, "Stationary velocity fields on matrix groups for deformable image registration," *arXiv preprint arXiv:2410.10997*, 2024.

[41] B. Ben Amor, S. Arguillère, and L. Shao, "Resnet-lddmm: advancing the lddmm framework using deep residual networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 3, pp. 3707–3720, 2023.

[42] A. Molaei, A. Aminimehr, A. Tavakoli, A. Kazerouni, B. Azad, R. Azad, and D. Merhof, "Implicit neural representation in medical imaging: A comparative survey," in *Proc. ICCV 2023*, pp. 2381–2391, 2023.

[43] A. Hering, L. Hansen, T. C. Mok, A. C. Chung, H. Siebert, S. Häger, A. Lange, S. Kuckertz, S. Heldmann, W. Shao, *et al.*, "Learn2reg: comprehensive multi-task medical image registration challenge, dataset and evaluation in the era of deep learning," *IEEE Trans. Med. Imaging*, vol. 42, no. 3, pp. 697–712, 2023.

[44] L. Tian, H. Greer, R. S. José Estépar, R. Sengupta, and M. Niethammer, "Nephi: Neural deformation fields for approximately diffeomorphic medical image registration," in *Proc. ECCV 2024*, pp. 213–237, Springer Nature Switzerland, 2025.

[45] H. Sun, "A review of 3D-2D registration methods and applications based on medical images," *Highl. Sci. Eng. Technol.*, vol. 35, pp. 200–224, 2023.

[46] M. Groher, D. Zikic, and N. Navab, "Deformable 2D-3D registration of vascular structures in a one view scenario," *IEEE Trans. Med. Imaging*, vol. 28, no. 6, pp. 847–860, 2009.

[47] J. H. Hipwell, G. P. Penney, R. A. McLaughlin, K. Rhode, P. Summers, T. C. Cox, J. V. Byrne, J. A. Noble, and D. J. Hawkes, "Intensity-based 2-D-3-D registration of cerebral angiograms," *IEEE Trans. Med. Imaging*, vol. 22, no. 11, pp. 1417–1426, 2003.

[48] S. Benameur, M. Mignotte, S. Parent, H. Labelle, W. Skalli, and J. de Guise, "3D/2D registration and segmentation of scoliotic vertebrae using statistical models," *Comput. Med. Imaging Graph.*, vol. 27, no. 5, pp. 321–337, 2003.

[49] P. Markelj, D. Tomaževič, B. Likar, and F. Pernuš, "A review of 3D/2D registration methods for image-guided interventions," *Med. Image Anal.*, vol. 16, no. 3, pp. 642–661, 2012.

[50] S. Heldmann and N. Papenberg, "A variational approach for volume-to-slice registration," in *Proc. SSVM 2009*, pp. 624–635, Springer, 2009.

[51] M. Unberath, C. Gao, Y. Hu, M. Judish, R. H. Taylor, M. Armand, and R. Grupp, "The impact of machine learning on 2D/3D registration for image-guided interventions: A systematic review and perspective," *Front. Robot. AI*, vol. 8, p. 716007, 2021.

[52] P. Maken and A. Gupta, "2D-to-3D: a review for computational 3D image reconstruction from x-ray images," *Arch. Comput. Methods Eng*, vol. 30, no. 1, pp. 85–114, 2023.

[53] E. Ferrante and N. Paragios, "Slice-to-volume medical image registration: A survey," *Med. Image Anal.*, vol. 39, pp. 101–123, 2017.

[54] J. Weickert and C. Schnörr, "A theoretical framework for convex regularizers in PDE-based computation of image motion," *Int. J. Comput. Vis.*, vol. 45, pp. 245–264, 2001.

[55] G. Song, N. Tustison, B. Avants, and J. C. Gee, "Lung ct image registration using diffeomorphic transformation models," in *Proc. Med. Image Anal. Clin.: Grand Chall.*, pp. 23–32, 2010.

[56] K. S. Nayak, Y. Lim, A. E. Campbell-Washburn, and J. Steeden, "Real-time magnetic resonance imaging," *J. Magn. Reson. Imaging*, vol. 55, no. 1, pp. 81–99, 2022.

[57] L. Ruthotto, *Hyperelastic image registration*. PhD thesis, Westfälische Wilhelms-Universität Münster, 2012.

[58] H. W. Alt, *Linear functional analysis. An application-oriented introduction*. London: Springer, 2016.

[59] A. Reithmeir, V. Spieker, V. Sideri-Lampretsa, D. Rueckert, J. A. Schnabel, and V. A. Zimmer, "From model based to learned regularization in medical image registration: A comprehensive review," *arXiv preprint arXiv:2412.15740*, 2024.

[60] A. Hering, S. Häger, J. Moltz, N. Lessmann, S. Heldmann, and B. Van Ginneken, "Cnn-based lung ct registration with multiple anatomical constraints," *Med. Image Anal.*, vol. 72, p. 102139, 2021.

[61] J. Nocedal and S. J. Wright, *Numerical Optimization*. New York, NY: Springer, 1999.

[62] X. Jiang, J. Ma, G. Xiao, Z. Shao, and X. Guo, "A review of multimodal image matching: Methods and applications," *Inf. Fusion*, vol. 73, pp. 22–71, 2021.

[63] J. Hadamard, "Sur les problèmes aux dérivées partielles et leur signification physique," *Princeton university bulletin*, pp. 49–52, 1902.

[64] M. A. Fischler and R. A. Elschlager, "The representation and matching of pictorial structures," *IEEE Trans. Comput.*, vol. C-22, no. 1, pp. 67–92, 1973.

[65] C. Broit, *Optimal registration of deformed images*. PhD thesis, University of Pennsylvania, 1981.

[66] B. Fischer, J. Modersitzki, M. Nashed, and O. Scherzer, "Fast diffusion registration," *AMS Contemporary Mathematics, Inverse Problems, Image Analysis, and Medical Imaging*, vol. 313, pp. 117–129, 2002.

[67] B. Fischer and J. Modersitzki, "Curvature based image registration," *J. Math. Imaging Vis.*, vol. 18, no. 1, pp. 81–85, 2003.

[68] D. Rueckert, L. I. Sonoda, C. Hayes, D. L. G. Hill, M. O. Leach, and D. J. Hawkes, "Nonrigid registration using free-form deformations: application to breast mr images," *IEEE Trans. Med. Imaging*, vol. 18, no. 8, pp. 712–721, 1999.

[69] E. Haber and J. Modersitzki, "A multilevel method for image registration," *SIAM J. Sci. Comput.*, vol. 27, no. 5, pp. 1594–1607, 2006.

[70] A. Lange and S. Heldmann, "Multilevel 2D-3D intensity-based image registration," in *Proc. WBIR 2020*, pp. 57–66, Springer, 2020.

[71] P. Striewski and B. Wirth, "Elastic 3D-2D image registration," *J. Math. Imaging Vis.*, vol. 64, no. 5, pp. 443–462, 2022.

[72] L. R. Dice, "Measures of the amount of ecologic association between species," *Ecology*, vol. 26, no. 3, pp. 297–302, 1945.

[73] T. Sørensen, "A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on danish commons," *Biol. Skr.*, vol. 5, pp. 1–34, 1948.

[74] R. Wang, T. Lei, R. Cui, B. Zhang, H. Meng, and A. K. Nandi, "Medical image segmentation using deep learning: A survey," *IET Image Process.*, vol. 16, no. 5, pp. 1243–1267, 2022.

[75] P. Chen, Y. Guo, D. Wang, and C. Chen, "Dlung: Unsupervised few-shot diffeomorphic respiratory motion modeling," *J. Shanghai Jiaotong Univ. (Sci.)*, vol. 28, no. 4, pp. 536–545, 2023.

[76] S. Häger, A. Lange, S. Heldmann, J. Modersitzki, A. Petersik, M. Schröder, H. Gottschling, T. Lieth, E. Zähringer, and J. H. Moltz, "Robust intensity-based initialization for 2D-3D pelvis registration (robin)," in *BVM Workshop*, pp. 69–74, Springer, 2022.

[77] Y. Otake, A. S. Wang, J. Webster Stayman, A. Uneri, G. Kleinszig, S. Vogt, A. J. Khanna, Z. L. Gokaslan, and J. H. Siewerdsen, "Robust 3D–2D image registration: application to spine interventions and vertebral labeling in the presence of anatomical deformation," *Phys. Med. Biol.*, vol. 58, no. 23, pp. 8535—8553, 2013.

[78] W. Yu, M. Tannast, and G. Zheng, "Non-rigid free-form 2D–3D registration using a B-spline-based statistical deformation model," *Pattern Recognit.*, vol. 63, pp. 689–699, 2017.

[79] The MathWorks Inc., "Matlab, version 9.13.0 (r2022b)." `https://de.mathworks.com/products/matlab.html`, 2022.

[80] B. Berkels, I. Cabrilo, S. Haller, M. Rumpf, and K. Schaller, "Co-registration of intra-operative brain surface photographs and pre-operative MR images," *Int. J. Comput. Assist. Radiol. Surg.*, vol. 9, pp. 387–400, 2014.

[81] A. Yavariabdi, A. Bartoli, C. Samir, M. Artigues, and M. Canis, "Mapping and characterizing endometrial implants by registering 2D transvaginal ultrasound to 3D pelvic magnetic resonance images," *Comput. Med. Imag. Grap.*, vol. 45, pp. 11–25, 2015.

[82] W. Hinterberger, O. Scherzer, C. Schnörr, and J. Weickert, "Analysis of optical flow models in the framework of the calculus of variations," *Numer. Funct. Anal. Optim.*, vol. 23, no. 1-2, pp. 69–89, 2002.

[83] C. Pöschl, J. Modersitzki, and O. Scherzer, "A variational setting for volume constrained image registration," *Inverse Probl. Imaging*, vol. 4, no. 3, pp. 505–522, 2010.

[84] M. Burger, J. Modersitzki, and L. Ruthotto, "A hyperelastic regularization energy for image registration," *SIAM J. Sci. Comput.*, vol. 35, no. 1, pp. B132–B148, 2013.

[85] L. Ambrosio, A. Carlotto, and A. Massaccesi, *Lectures on elliptic partial differential equations*. Pisa: Edizioni della Normale, 2018.

[86] L. C. Evans, *Partial differential equations*, vol. 19. Providence, RI: AMS, 2022.

[87] R. A. Adams and J. J. F. Fournier, *Sobolev spaces*, vol. 140 of *Pure Appl. Math., Academic Press*. New York, NY: Academic Press, 2nd ed. ed., 2003.

[88] Y. Lei, Y. Fu, T. Wang, Y. Liu, P. Patel, W. J. Curran, T. Liu, and X. Yang, "4D-ct deformable image registration using multiscale unsupervised deep learning," *Phys. Med. Biol.*, vol. 65, no. 8, p. 085003, 2020.

[89] D. Shen, H. Sundar, Z. Xue, Y. Fan, and H. Litt, "Consistent estimation of cardiac motions by 4D image registration," in *Proc. MICCAI 2005*, pp. 902–910, Springer, 2005.

[90] M. F. Beg, M. I. Miller, A. Trouvé, and L. Younes, "Computing large deformation metric mappings via geodesic flows of diffeomorphisms," *Int. J. Comput. Vis.*, vol. 61, no. 2, pp. 139–157, 2005.

[91] L. Younes, *Shapes and diffeomorphisms*, vol. 171. Berlin: Springer, 2010.

[92] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *Proc. ECCV 2016*, pp. 630–645, Springer, 2016.

[93] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR 2016*, pp. 770–778, 2016.

[94] A. Trouvé, "An infinite dimensional group approach for physics based models in pattern recognition," *Technical Report*, 1995.

[95] G. E. Christensen, R. D. Rabbitt, and M. I. Miller, "Deformable templates using large deformation kinematics," *IEEE Trans. Image Process.*, vol. 5, no. 10, pp. 1435–1447, 1996.

[96] P. Dupuis, U. Grenander, and M. I. Miller, "Variational problems on flows of diffeomorphisms for image matching," *Q. Appl. Math.*, vol. 56, no. 3, pp. 587–600, 1998.

[97] S. Arguillère, E. Trélat, A. Trouvé, and L. Younes, "Shape deformation analysis from the optimal control viewpoint," *J. Math. Pures Appl. (9)*, vol. 104, no. 1, pp. 139–178, 2015.

[98] M. I. Miller, A. Trouvé, and L. Younes, "On the metrics and euler-lagrange equations of computational anatomy," *Annu. Rev. Biomed. Eng.*, vol. 4, no. 1, pp. 375–405, 2002.

[99] J. Ashburner and K. J. Friston, "Diffeomorphic registration using geodesic shooting and gauss–newton optimisation," *NeuroImage*, vol. 55, no. 3, pp. 954–967, 2011.

[100] M. Zhang and P. T. Fletcher, "Finite-dimensional lie algebras for fast diffeomorphic image registration," in *Proc. IPMI 2015*, pp. 249–260, Springer, 2015.

[101] V. Arsigny, O. Commowick, X. Pennec, and N. Ayache, "A log-euclidean framework for statistics on diffeomorphisms," in *Proc. MICCAI 2006*, pp. 924–931, Springer, 2006.

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

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

[104] A. Maier, C. Syben, T. Lasser, and C. Riess, "A gentle introduction to deep learning in medical image processing," *Z. Med. Phys.*, vol. 29, no. 2, pp. 86–101, 2019.

[105] C. F. Higham and D. J. Higham, "Deep learning: An introduction for applied mathematicians," *SIAM Rev.*, vol. 61, no. 4, pp. 860–891, 2019.

[106] S. R. Dubey, S. K. Singh, and B. B. Chaudhuri, "Activation functions in deep learning: A comprehensive survey and benchmark," *Neurocomputing*, vol. 503, pp. 92–108, 2022.

[107] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR 2015*, 2015.

[108] R. Mohanasundaram, A. S. Malhotra, R. Arun, and P. Periasamy, "Deep learning and semi-supervised and transfer learning algorithms for medical imaging," in *Deep learning and parallel computing environment for bioengineering systems*, pp. 139–151, Elsevier/Academic Press, 2019.

[109] E. Weinan, "A proposal on machine learning via dynamical systems," *Commun. Math. Stat.*, vol. 5, no. 1, pp. 1–11, 2017.

[110] E. Haber and L. Ruthotto, "Stable architectures for deep neural networks," *Inverse problems*, vol. 34, no. 1, p. 014004, 2017.

[111] L. Euler, *Institutiones calculi integralis*. impensis Academiae imperialis scientiarum, 1768.

[112] C. Zhang and X. Liang, "Inr-lddmm: fluid-based medical image registration integrating implicit neural representation and large deformation diffeomorphic metric mapping," *arXiv preprint arXiv:2308.09473*, 2023.

[113] H. Huang, B. B. Amor, X. Lin, F. Zhu, and Y. Fang, "Residual networks as flows of velocity fields for diffeomorphic time series alignment," *arXiv preprint arXiv:2106.11911*, 2021.

[114] A. Joshi and Y. Hong, "Metaregnet: Metamorphic image registration using flow-driven residual networks," in *Proc. CMMCA 2023*, pp. 160–170, Springer, 2023.

[115] V. Sitzmann, J. Martel, A. Bergman, D. Lindell, and G. Wetzstein, "Implicit neural representations with periodic activation functions," in *Proc. NeurIPS 2020*, vol. 33, pp. 7462–7473, 2020.

[116] G. Teschl, *Ordinary differential equations and dynamical systems*, vol. 140. Providence, RI: AMS, 2012.

[117] V. A. Zorich, *Mathematical analysis II*. Berlin: Springer, 2nd edition ed., 2016.

[118] J. Behrmann, W. Grathwohl, R. T. Chen, D. Duvenaud, and J.-H. Jacobsen, "Invertible residual networks," in *Proc. ICML 2019*, pp. 573–582, PMLR, 2019.

[119] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," in *Proc. ICLR 2018*, 2018.

[120] H. Gouk, E. Frank, B. Pfahringer, and M. J. Cree, "Regularisation of neural networks by enforcing lipschitz continuity," *Mach. Learn.*, vol. 110, no. 2, pp. 393–416, 2021.

[121] H. Sedghi, V. Gupta, and P. M. Long, "The singular values of convolutional layers," in *Proc. ICLR 2019*, 2019.

[122] Python Software Foundation, "Python language reference, version 3.10.9." `http://www.python.org`, 2022.

[123] K. I. Roumeliotis and N. D. Tselikas, "Chatgpt and open-ai models: A preliminary review," *Future Internet*, vol. 15, no. 6, 2023.

[124] Z. Liu, Y. Wang, S. Vaidya, F. Ruehle, J. Halverson, M. Soljacic, T. Y. Hou, and M. Tegmark, "Kan: Kolmogorov-arnold networks," in *Proc. ICLR 2025*, 2025.

[125] M. V. Narkhede, P. P. Bartakke, and M. S. Sutaone, "A review on weight initialization strategies for neural networks," *Artif. Intell. Rev.*, vol. 55, no. 1, pp. 291–322, 2022.

[126] V. Vishnevskiy, T. Gass, G. Szekely, C. Tanner, and O. Goksel, "Isotropic total variation regularization of displacements in parametric image registration," *IEEE Trans. Med. Imaging*, vol. 36, no. 2, pp. 385–395, 2017.

[127] L. Zhou, M. Fan, C. Hansen, C. R. Johnson, and W. Daniel, "A review of three-dimensional medical image visualization," *Health Data Sci.*, p. 9840519, 2022.

[128] G. E. Christensen, R. D. Rabbitt, M. I. Miller, *et al.*, "A deformable neuroanatomy textbook based on viscous fluid mechanics," in *Proc. CISS 1993*, vol. 27, pp. 211–216, 1993.

[129] B. C. Lee, "torch-lddmm." `https://github.com/brianlee324/torch-lddmm`, 2024.

[130] Y. Amit, "A nonlinear variational problem for image matching," *SIAM J. Sci. Comput.*, vol. 15, no. 1, pp. 207–224, 1994.

[131] R. Castillo, E. Castillo, R. Guerra, V. E. Johnson, T. McPhail, A. K. Garg, and T. Guerrero, "A framework for evaluation of deformable image registration spatial accuracy using large landmark point sets," *Phys. Med. Biol.*, vol. 54, no. 7, pp. 1849–1870, 2009.

[132] J. Rühaak, T. Polzin, S. Heldmann, I. J. Simpson, H. Handels, J. Modersitzki, and M. P. Heinrich, "Estimation of large motion in lung ct by integrating regularized keypoint correspondences into dense deformable registration," *IEEE Trans. Med. Imaging*, vol. 36, no. 8, pp. 1746–1757, 2017.

[133] J. Hofmanninger, F. Prayer, J. Pan, S. Röhrich, H. Prosch, and G. Langs, "Automatic lung segmentation in routine imaging is primarily a data diversity problem, not a methodology problem," *Eur. Radiol. Exp.*, vol. 4, pp. 1–13, 2020.

[134] T. Fechter and D. Baltas, "One-shot learning for deformable medical image registration and periodic motion tracking," *IEEE Trans. Med. Imaging*, vol. 39, no. 7, pp. 2506–2517, 2020.

[135] F. Alam, S. U. Rahman, S. Ullah, and K. Gulati, "Medical image registration in image guided surgery: Issues, challenges and research opportunities," *Biocybern. Biomed. Eng.*, vol. 38, no. 1, pp. 71–89, 2018.

[136] J. Sauer, J. Graßhoff, N. M. Carbon, W. M. Koch, S. Weber-Carstens, and P. Rostalski, "Automated characterization of patient–ventilator interaction using surface electromyography," *Ann. Intensive Care*, vol. 14, no. 1, p. 32, 2024.

[137] A. H. Jonkman, R. S. Warnaar, W. Baccinelli, N. M. Carbon, R. F. D'Cruz, J. Doorduin, J. L. van Doorn, J. Elshof, L. Estrada-Petrocelli, J. Graßhoff, *et al.*, "Analysis and applications of respiratory surface emg: report of a round table meeting," *Crit. Care*, vol. 28, no. 1, p. 2, 2024.

[138] D. Pereira Botelho, K. Curran, and M. M. Lowery, "Anatomically accurate model of emg during index finger flexion and abduction derived from diffusion tensor imaging," *PLOS Comput. Biol.*, vol. 15, no. 8, p. e1007267, 2019.

[139] J. H. Yoon, M. D. Nickel, J. M. Peeters, and J. M. Lee, "Rapid imaging: recent advances in abdominal mri for reducing acquisition time and its clinical applications," *Korean J. Radiol.*, vol. 20, no. 12, pp. 1597–1615, 2019.

[140] W. T. Dixon, "Simple proton spectroscopic imaging.," *Radiology*, vol. 153, no. 1, pp. 189–194, 1984.

[141] J. Barkhausen, S. G. Ruehm, M. Goyen, T. Buck, G. Laub, and J. F. Debatin, "Mr evaluation of ventricular function: True fast imaging with steady-state precession versus fast low-angle shot cine mr imaging: Feasibility study," *Radiology*, vol. 219, no. 1, pp. 264–269, 2001.

[142] Materialise NV, Leuven, Belgium, "Materialise mimics 24.0." `https://www.materialise.com/en/industrial/software`, 2021.

[143] S. G. Spiro, G. A. Silvestri, and A. Agustí, *Clinical respiratory medicine*. Philadelphia: Elsevier Saunders, 2012.

[144] Y. Fu, S. Liu, H. Harold Li, H. Li, and D. Yang, "An adaptive motion regularization technique to support sliding motion in deformable image registration," *Med. Phys.*, vol. 45, no. 2, pp. 735–747, 2018.