Newton Methods for Image Registration

Dissertationsschrift zur Erlangung
des naturwissenschaftlichen Doktorgrades
der Bayerischen Julius-Maximilian-Universität Würzburg

vorgelegt von

Martin Schröter

aus
Würzburg

18. Januar 2012

Eingereicht am: 18.01.2012

bei der Fakultät für Mathematik und Informatik

“Doch, Papier könnten wir leicht entbehren, wenn wir nur Wachstafeln, oder Baumrinden, oder Häute, oder Palmblätter hätten! und in Ermanglung deren möchten es weisses Blech, Marmor, Elfenbein, oder gar Backsteine thun; denn auf alle diese Dinge pflegte man ehmals zu schreiben, als es noch mehr darum zu thun war dauerhaft als viel zu schreiben [. . .] Nicht schreiben wäre wohl das kürzeste Mittel; aber schreiben will ich nun, das ist beschlossen.”

Die Dialogen des Diogenes von Sinope

Contents

1 Introduction **7**
1.1 The Problem . . . 7
1.2 Previous Work . . . 10
1.3 An Overview . . . 12

2 Optimization Methods on Matrix Lie Groups **17**
2.1 Preliminaries on Matrix Lie Groups . . . 18
2.1.1 Affine Volume-Preserving Transformations . . . 18
2.1.2 Rigid-Body Motions . . . 21
2.2 The (μ, ν)-Newton Algorithm . . . 22
2.2.1 Cost Functions with Group Actions . . . 23
2.3 Stochastic Optimization Methods . . . 33
2.3.1 Preliminaries on Asymptotic Pseudotrajectories . . . 33
2.3.2 Flow-Tracking on Lie Groups . . . 35
2.3.3 Tracking Methods on Manifolds . . . 45
2.3.4 The Robbins-Monro Algorithm . . . 47

3 A Geometric Model for Image Representation **49**
3.1 From 3D Pose-Estimation to Image Registration . . . 49
3.2 Representation of an Image . . . 51
3.3 Image Distance Measures . . . 54
3.3.1 Sum of Squared Differences . . . 55
3.3.2 Mutual Information . . . 55
3.3.3 δ-Distance . . . 56
3.4 The Optimization Problem . . . 62

4 Topics in Exact Image Registration **65**
4.1 Preliminaries on Transversality Theory . . . 66
4.2 Image Appearance Manifolds with Finite Evaluations . . . 71
4.2.1 Properties of Special Group Actions . . . 72
4.2.2 Generic Properties for Fixed Evaluation Points . . . 74
4.2.3 Generic Properties for Fixed Images . . . 87

5 Image Reg. using an Approximate-Newton Alg. 93
5.1 Registration on Smooth Function Spaces 94
5.1.1 The Quasi-Monte Carlo Newton Algorithm 94
5.1.2 A Generic Property of the Cost Function 98
5.2 Registration on Spline Function Spaces 106
5.2.1 Monomodal Registration 107
5.2.2 Multimodal Registration 110
5.3 Experimental Results 116
5.3.1 Verification of the Convergence Rate 116
5.3.2 Comparison with Optimization Techniques on Vector Spaces 120
5.3.3 Comparison with Previous Registration Algorithms 124

6 Image Registration using Stochastic Optimization 131
6.1 Variants of the Robbins-Monro Algorithm 131
6.1.1 Affine Monomodal Registration 132
6.1.2 Affine Multimodal Registration 135
6.1.3 Non-Rigid Registration 137
6.2 Experimental Results 140
6.2.1 Rigid Registration methods in Comparison 141
6.2.2 Non-Rigid Registration Methods in Comparison 148

A The Quasi-Monte Carlo Method 151

Chapter 1

Introduction

1.1 The Problem

Consider the situation where two or more images are taken from the same object. After taking the first image, the object is moved or rotated so that the second recording depicts it in a different manner. Additionally, take heed of the possibility that the imaging techniques may have also been changed. One of the main problems in image processing is to determine the spatial relation between such images. The corresponding process of finding the spatial alignment is called "registration". For instance, in the toy example of Fig. 1.1 we can move and rotate the left image in such a way that it becomes identical to the right image. Here, the registration task is to find the correct translation and rotation.

The process of registration becomes often necessary when the comparison or combination of the information from different images is wished for. The fields of application range from astro-physics, biology, medicine and robotics to computer vision (cf. [1]). It basically appears whenever a question of the form "Where is the object in the new image?", "What has changed?", "Which parts grew?" or "What is a suitable transformation to get the new image?" arises.

In particular, in medical imaging the registration task becomes a necessary tool in diagnostic settings as well as in the planning and assisting of surgical processes (cf. e.g. [2]). In radiation therapy, for instance, images from different imaging techniques, the so-called modalities, form the indispensable base. Treatment volumes and organs at risk are defined using computerized tomography (CT) and magnetic resonance (MR) images [3]. In order to aid this task, often functional imaging, like proton emission tomography (PET) is performed [4]. These images have to be co-registered during the treatment planning process [5]. The aim is to define the target volume based on quantitative measures of the tumor cell density of the tissue in question. During treatment, imaging helps to determine the correct position of the patients. In general, orthogonal X-ray images are matched with the data set from the planning CT. More and more CT imaging is performed directly on the treatment couch and can thus be compared to the planning CT in every treatment session (cf. [6, 7, 8, 9]). The strong dependence on various imaging techniques in radiation therapy necessitates fast, robust and reliable image registration tools.

According to Maintz et al. [2] a (medical) registration process can always be decomposed

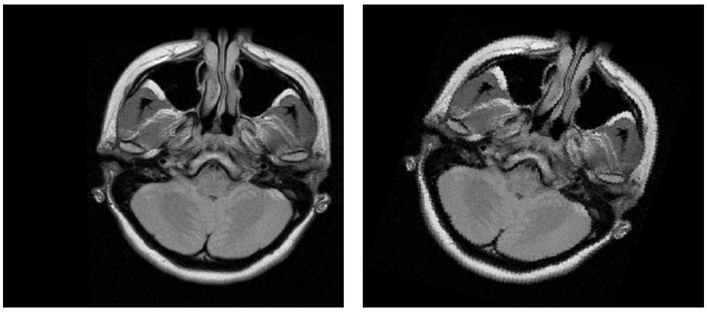


Figure 1.1: Toy example of a registration problem: We seek a rotation and a translation which transfers the image on the left best to the image on the right.

into three major parts:

- The problem statement: Here, the specification of the modalities which are used to create the images and of the morphological parts which are under considerations takes place. We also have to distinguish between patient-to-patient problems (comparing the anatomy of two patients), single-patient problems (for instance putting the physical recovery under surveillance) and atlas problems (comparing a patient to a database).
- The registration paradigm: In each registration process, we have to transform one image and then compare it with another one. Thus, the specification of the admissible set of transformations becomes necessary. Moreover, we have to decide whether we use the pure voxel-data of the image for a comparison, or whether we incorporate segmentation processes and use the information of the contours of the objects. It also should be clarified to what extend a user is involved in the registration process. For instance, does he identify landmarks or does he give an initial transformation?
- The optimization problem: In almost all registration processes, the task is expressed in terms of an optimization problem. This means that an appropriate cost function can be created which is defined on the set of admissible transformations and the optimum of this function is considered as the solution of the registration task. Thus, it is necessary to specify the optimization algorithm.

In this thesis we are not interested in a specific registration problem, although this work is mainly motivated by medical imaging. We mainly focus on optimization problems, which correspond to the registration task. However, the three points mentioned above are not completely decoupled and it seems to be impossible to present one general optimization problem of all various kinds of registration problems. Hence, we will specify our setting, i.e. the problem statement and the registration paradigm, as follows:

We consider the case of two given images, the so-called reference R and the so-called template T. These images are supposed to be grayscale and n-dimensional (where $n = 2$ or $n = 3$ are most common in medicine). We will see in Section 3.2 that R and T can be represented as real-valued functions of the form $\mathbb{R}^n \to \mathbb{R}$. A key point in this work will be our assumption that the set of admissible transformations can be represented by a finite-dimensional Lie group

G which acts on the space of images. This might be too restrictive for some applications since it implies that each admissible transformation has a unique inverse. We will oftentimes focus on the following two examples of G: the affine volume-preserving transformations $SA(n)$ and the Euclidean transformations $SE(n)$. Thus, the optimization problem which corresponds to the registration task has the form

$$\min_{g \in G} \mathfrak{D}\left(R \circ g, T\right). \tag{1.1}$$

Here, $\mathfrak{D}$ denotes a function which measures the distance between two images. The choice of $\mathfrak{D}$ is part of the problem statement. In addition, it has to be taken into account that the images might be taken by different modalities. We give examples of $\mathfrak{D}$ in Section 3.3. A regularization term might be necessary in (1.1) as well if the dimension of G is too high (cf. e.g. [10]).

We want to point out that there might be a gap between a perfect alignment of two given images, which is intuitively considered to be the solution of the registration task, and the solution of (1.1), caused by the huge differences between our approach and the processes of human vision. In human cognition an image contains several additional pieces of information, like shapes and the position of objects. These objects are recognized using edge detection and segmentation processes (see e.g. [11] and the reference therein). In human vision, the registration task is accomplished by a comparison of these objects, and not by a comparison the raw pixel data, as is the case in (1.1). In medical imaging, it usually is the other way round: in order to segment the organs in a real medical image, the image data are registered with a human body atlas, where the exact positions of the organs and their shapes are known, see e.g. [12, 13, 14]. Nevertheless, each registration algorithm has to show that its results agree with the human intuition, but this is more a question of the choice of the similarity measure $\mathfrak{D}$ than of the optimization algorithm.

In applications, the distinction is to be made between monomodal (R and T are made by the same modality) and multimodal (R and T are made by different modalities) registration. The monomodal case is, for instance, used for treatment verification by comparing the pre- and post-intervention images for growth monitoring (cf. [2]). The multimodal registration is necessary for combining the information from different imaging techniques. In computerized tomography (CT), for instance, an object is hit by X-rays approaching from one side. The denser the material is along the path of the rays, the smaller the amount of radiation is on the other side which is detectable with a photographic plate. Bones for instance absorb a large amount of the rays which is the reason why they appear black. In Fig. 1.1 the negative image of a CT can be seen. Other imaging techniques can make use of completely different properties of the matter. The magnetic resonance imaging (MR), for instance, does not measure the density distribution, but the concentration of H-protons. (We refer to [15] for more details in the underlying physics.) This provides a higher contrast between different, soft tissues, since one can find particularly H-protons in water and fat tissue; cf. Fig. 1.2. In the Proton Emission Tomography (PET) the patient is administered a radioactive preparation, most often intravenously. Depending on its chemical properties, this substance accumulates to specific organs or to the blood circulation. The widely used $^{18}F-$Fluordesoxyglucose, for instance, behaves similar to sugar. Hence, it accumulates in regions of the human body which have a high energy requirement, like the brain or tumors. Through the detection of

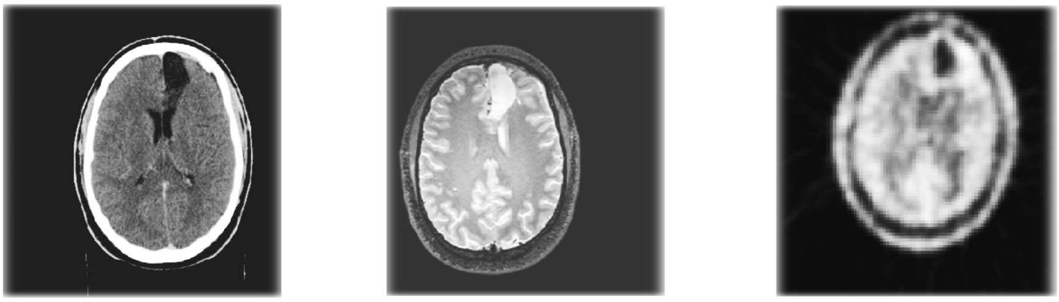


Figure 1.2: Each of the three images show the same cross-section of a patient's skull. The first is made by computerized tomography (CT), the second by magnetic resonance imaging (MR) and the third by proton emission tomography (PET).

its radiation, it is possible to deduce the 3D distribution of this substance. These and other imaging processes create 3D data sets, which indicate certain properties of the respective tissue wherefore they are primarily used in medical diagnostics.

Another application of the registration task is the so-called segmentation of organs: Segmentation is the process of determining a region of interest of an image. This region is typically given by some homogeneous properties of the gray values. For instance, a segmentation task in the right image of Fig. 1.1 would be to determine the form of the soft tissue (gray matter) of the skull. This question is strongly related to the problem of edge detection and active contours techniques (we refer to [15] for an overview). However, a lot of practical segmentation techniques in medicine work with an atlas of the whole human body or parts of it. In such a case, the segmentation of an organ is already done for a test data set R and another data set T is given. It shows of the same region of the human body but from another patient. Hence, we can solve the registration problem of R and T and transfer the contour of the organ in R to the one in T instead of solving the segmentation task of T.

The task of (medical) image processing is to find fast and accurate algorithms for these kinds of questions. The imaging techniques have undergone such a fast development that nowadays the speed of these algorithms restricts the use of image guided medicine. Therefore, scientific research in this area is of direct, practical benefit.

In this work, we study the optimization problem (1.1). Especially, we exploit the Lie group structure of the set of transformations to construct efficient, intrinsic algorithms. We also apply the algorithms to medical registration tasks. However, the methods developed are not restricted to the field of medical image processing. We also have a closer look at more general forms of optimization problems and show connections to related tasks.

1.2 Previous Work

As mentioned in the last section image registration is a fundamental task in image processing, with applications in various fields, including e.g. robotics [16] and geophysics [17]. For medical image applications, registration is used for image-based treatment planning and image-guided treatment delivery, see e.g. [1, 2, 18] and the references therein. Although an overwhelming number of publications focuses on non-rigid registration, where the task is to find a non-linear diffeomorphism, rigid registration is still of considerable interest and may lead to good starting point for a subsequent non-rigid registration phase; see e.g. [19, 20, 21, 22, 23]. Often, as in e.g. [10, 24], rigid image registration algorithms are designed to employ fixed local coordinates of the Euclidean group via Euler-angles and then to apply standard optimization algorithms on the affine parameter space. This simple local coordinate chart approach, however, has its drawbacks and may lead to ill-conditioned algorithms at the boundary of the parameter space. For example, in [25] it is observed that the singularities inherent in local Euler angle coordinates may reduce the speed of convergence, compared to algorithms acting on the Lie group. Therefore Lee et al. propose a linearly convergent Nelder-Mead algorithm on the Lie group $SE(n)$ for multi-modal image registration in [25]. In order to achieve faster, locally quadratic convergence rates we develop a new type of approximate-Newton methods on $SE(n)$ that avoids the singularities of Euler-angle coordinates.

Of course, the task of applying Newton's method to Lie groups is not new and has been already exploited e.g. to robotics and computer vision problems; see e.g. Park [26], Hüper et al.[27] and Sastry [28] for background material. However, such preceding work suffers from a number of shortcomings that limit the applicability to image registration problems. The Riemannian Gauss-Newton method [29] performs a Newton step by a line search along a geodesic. For the Euclidean rotation group $SO(n)$, such geodesics require the computation of the matrix exponential of a skew-symmetric matrix; which may be a difficult numerical task with regard to high dimensions. Moreover, since the Euclidean group $SE(n)$ carries no natural bi-invariant Riemannian metric, such geodesics are only described by solutions to nonlinear second order differential equations which are difficult to compute.

Even in the case of non-rigid registration, the Lie group structure of the admissible set has a key position in the performance of the algorithms. Here, the corresponding optimization problem is defined on the group of diffeomorphisms. In [30] it is mentioned that incorporating geodesics leads to an increase of the accuracy. However, calculating the geodesics, which is equivalent to solving a time-varying ODE, is a time-consuming procedure. Several approaches are known in literature which avoid this step: for example, in [31] vector fields are used which fulfill the momentum conservation equation, in [32] the authors use one-parameter subgroups to approximate the geodesics and propose a fast method for calculating the vector field exponential. We refer to [33] or [34], and the reference therein, for a further study of the task of non-parametric image registration.

Finally, we want to mention that image registration is strongly connected with the pose-estimation task (also known as point set registration or 3D-3D alignment). Here, the problem is to find simultaneously the correspondence between the points of two given sets $\mathcal{X}, \mathcal{Y} \subset \mathbb{R}^3$ of $m \in \mathbb{N}$ elements and the best Euclidean transformation $g \in SE(3)$, such that $g \cdot \mathcal{X}$ and $\mathcal{Y}$ coincides; see Section 3.1 for an introduction.

One class of methods treats this task as an optimization problem on $SE(3) \times S_m$, where the symmetric group S_m represents the unknown correspondence between the elements of $\mathcal{X}$ and $\mathcal{Y}$. Here, often stochastic algorithms are used (cf. e.g. [35, 36]) for global optimization or alternating processes for local methods. For instance, the famous ICP algorithm solves first the correspondence sub-problem for fixed $g \in SE(3)$. Then, it searches the best pose while the correspondence is fixed and repeats both procedures in each iteration; see e.g. [37, 38] for an overview. But also exact methods for the noise-free case are available; cf. e.g. [39] and the reference therein.

Another class of methods represents the elements of $\mathcal{X}$ by a smooth function $p_{\mathcal{X}} : \mathbb{R}^3 \to \mathbb{R}$, which denotes the kernel density estimate of $\mathcal{X}$. Comparing both density functions $p_{\mathcal{X}}$ and $p_{\mathcal{Y}}$ reduces the pose-estimation task to an optimization problem on $SE(3)$; see Section 3.1 for an introduction. In many methods (cf. e.g. [40, 41, 42]), this Lie group is globally parameterized and an optimization method on vector spaces is used. In [43] linear and quadratic approximations of the matrix-exponential map are used to design a Newton method and in [44] an intrinsic Newton method on $SE(n)$ is presented, which incorporates the Riemannian geometry.

1.3 An Overview

The main achievements of this thesis are the following:

- **Development of an approximate-Newton algorithm for certain optimization problems on $SE(n)$ and $SA(n)$.** We are especially interested in cost functions involving the natural group action of $SE(n)$ and $SA(n)$ on the vector-space $\mathbb{R}^n$.
- **A Lie-group extension of a stochastic flow-tracking process.** In particular, the application of this process to gradient fields leads to a generalization of the Robbins-Monro algorithm.
- **Applying both methods to the image registration task.** We develop algorithms, which take into account the special forms of image representations. In particular, we consider the problem, how to handle the huge amount of data, which arises from high resoluted and often 3D images. Extensions to the multi-modal image registration problem are also given.

In the following, we give a more detailed description: in this work, we mainly consider optimization problems including two special groups of transformations in $\mathbb{R}^n$, the affine volume-preserving transformations $SA(n)$, and the Euclidean transformations $SE(n)$. In the first section of Chapter 2, we give an introduction of these Lie groups with a focus on their group action properties on $\mathbb{R}^n$. Additionally, each one of these two Lie groups is endowed with two different local parameterizations. In this context, we introduce a new parameterization for the $SL(n)$ (cf. Lemma 2.1) based only on the QR-factorization and not on the matrix-exponential map.

In the rest of Chapter 2, we present several algorithms for optimization problems on Lie groups. In Section 2.2, we construct a general class of cost functions, which involves smooth group actions of the Euclidean or volume-preserving transformation group and we develop a novel framework for the minimization of these cost functions. For this purpose, we use a modified (μ, ν)-Newton algorithm (cf. [45, 46]) and apply it to the respective Lie groups. The local parameterizations of these manifolds is chosen in such a way that we get a very efficient and easily implementable algorithm. Our main result in this chapter is that we prove local quadratic convergence of both methods under the condition that the corresponding cost function has no degenerated critical points (cf. e.g. Theorem 5.1).

In Section 2.3, we consider a special setting of optimization problems on Lie groups. We assume that the measurements of the cost function or its derivatives are disturbed by noise.

For these cases, we introduce the Robbins-Monro algorithm. In particular, we extend a result of Benaïm and Hirsch (cf. [47]) to a Lie group setting: let a Lie group G, its Lie algebra $\mathfrak{g}$ and a local parameterization $\mu : \mathfrak{g} \to G$ of G around $id \in G$ be given. We study iterations of the form

$$x_{k+1} = \mu(\gamma_k[F(x_k) + U_k])x_k, \quad x_0 \in G, \tag{1.2}$$

where $F : G \to \mathfrak{g}$ is a given vector field, $\gamma_k \in \mathbb{R}$, $k \in \mathbb{N}_0$ is a given sequence of step-sizes and $U_k \in \mathfrak{g}$, $k \in \mathbb{N}_0$ is a sequence of perturbations. In Theorem 2.13 we show that a rather general class of such perturbations U_k does not influence the limiting behavior of the flow tracking algorithm (1.2). More precisely, for certain classes of perturbations, the interpolation of $(x_k)_{k\in\mathbb{N}}$ by one-parameter groups yields an asymptotic pseudotrajectory of the flow Φ, induced by F. In Section 2.3.3, we extend the result to stochastic perturbations, which satisfy the "Martingale difference Noise" property (cf. Definition 2.4). In Corollary 2.16, we end up with a manifold-version of the Robbins-Monro type algorithms.

In Chapter 3, we formulate the abstract image registration task as an optimization problem on Lie groups. We begin with a description of some basic image processing techniques. In particular, we show how to construct a function representation of an image and how to smooth images. In the next section, we introduce some methods measuring the amount of differences between two given images. There will be a special interest when two images are made by different modalities and we compare the classical mutual information measure with a new distance function $\mathfrak{D}_\delta$. The latter one preserves some advantages of a metric, which is profitable for the optimization methods in later chapters. At the end of this chapter, we formulate a multiresolution approach of the image registration task. Connections to the pose-estimation problem are given, too.

An important part of our work tackles the process of image motion and image deformation. In Chapter 4, we model such a deformation with a group acting on the image: A given action of a finite-dimensional Lie group G onto a real vector space $\mathbb{R}^n$ generates an action onto the function space $C(\mathbb{R}^n, \mathbb{R})$ in a natural way (which we want to equate with the space of all images) by $(g \cdot f)(x) := f(g \cdot x)$ for all $g \in G$ and $f \in C(\mathbb{R}^n, \mathbb{R})$. However, every image processing algorithm has to work with a limited number of samples of f and not with the function itself. In contrast to other approaches [48, 49], it is more convenient to study the evaluation map

$$G \to \mathbb{R}^n, \quad g \mapsto (f(g \cdot x_1), \dots, f(g \cdot x_N))^\top \tag{1.3}$$

instead, where $\{x_i\}_{i=1}^N \subset \mathbb{R}^n$ is a set of previously given sample-points. In Chapter 4 we discuss the question under which conditions this map becomes an embedding of the Lie group G.
Of course, if we fix the set $\{x_i\}_{i=1}^N$, the answer depends on the particular choice of the image f and one can easily find examples of images f where (1.3) is never injective or an immersion. However, we show in Section 4.2 that the set of such exceptions is thin. In Theorem 4.9 we prove that $N = 2n$ sample-points are sufficient to get the immersion-property of the evaluation map (1.3) for a generic subset of functions f with respect to the strong topology. This result can be regarded as a special variant of the well-known theorem that immersions

are dense in $C^1_S(\mathcal{M},\mathcal{N})$, if the manifolds $\mathcal{M}$ and $\mathcal{N}$ satisfy $\dim \mathcal{N} \geq 2 \dim \mathcal{M}$ (cf. e.g. [50] p.53). However, we see in this chapter that a similar result for the injectivity of the evaluation map (1.3) requires additional geometric properties on the sample-points: In Theorem 4.12 we give necessary and sufficient conditions for the distribution of $N = 2n + 1$ sample-points such that (1.3) is an embedding for an generic choice of f. Consider, for instance, the special case that G is the Euclidean transformation group in $\mathbb{R}^3$ and the function f is fixed and taken from an appropriate generic set. Here, $N = 13$ sample-points are enough to detect any unknown transformation $g \in G$ from the evaluations $(f(g \cdot x_1), \ldots, f(g \cdot x_N))^\top$, if and only if not more than 5 of these sample-points lie on a line. We give a proof of a much more general version of this statement in Section 4.2.2.

In Section 4.2.3 we reverse the question: Now, we search for properties of a given image f, such that (1.3) is an embedding for a generic choice of the sample-points $\{x_i\}_{i=1}^N \subset \mathbb{R}^n$. Unfortunately, this question can only be partially answered: We give a sufficient condition for f such that (1.3) is generically an immersion. Then, the map (1.3) is at least locally injective, but a comparable, global statement is still missing.

In Chapter 5, we apply the modified (μ, ν)-Newton algorithm to the monomodal image registration task. A bottleneck in implementing such algorithms lies in the difficulty of effectively evaluating the higher dimensional integrals, which may suffer from the curse of dimensionality. In this chapter two new strategies to circumvent this problem are presented and discussed. The first one is incorporated in the new QMC-Newton algorithm, which compares two images in a sequence of points with low discrepancy. Then the cost function can be easily approximated by the Quasi-Monte Carlo method. The second strategy is incorporated in the new SB-Newton algorithm, which uses B-spline approximations of the images. Here, no image evaluations are necessary; the algorithms operate directly on the compressed (e.g. jpeg-like) data. Extensions to multimodal registration suing the mutual information measure are given too.

Our main result in this chapter is that we prove local quadratic convergence of both methods under the condition that the corresponding cost function has no degenerated critical points (cf. e.g. Theorem 5.1). Once more, this condition is satisfied for a generic set of images, which is shown in Theorem 5.4 and Theorem 5.7.

Our numerical simulations show that both strategies provide a high accuracy even in the case of highly compressed data. Additionally, we confirm the locally quadratic convergence of both methods in numerical experiments. When compared to one another, the QMC-Newton step is performed in less computational time than a SB-Newton step - at least in our implementation. But it turns out that the second method provides a higher accuracy in detecting the requested transformation.

It also turns out that the number of image evaluations of an entire algorithm is the most crucial part for the computation time. The lesson of our numerical experiments is that this part can even predominate the gain of the local quadratic convergence of a Newton method. Another crucial point is the size of the domain in which the algorithms converge quadratic. On the one hand, this domain can be quite small when it comes to unsmoothed images. On the other hand, the high accuracy is not necessary any more on the coarse smoothing levels and first-order algorithms become more attractive, if we apply a multi-scale approach and if we smooth the images on each level.

Heeding the results of Chapter 5, we study in Chapter 6 an additional type of registration algorithms which needs extraordinarily low image evaluations. We apply the Robbins-Monro algorithms from Section 2.3. The resulting method might be seen as a pre-registration step but these methods often produce sufficiently good results by themselves. In contrast to the introduced Newton algorithms, these Robbins-Monro types belong to stochastic optimization methods. Here, the "[...] fundamental philosophy [...] is that the loss function and gradient measurements are the dominant cost in the optimization process; the other calculations in the algorithms are considered relatively unimportant" (Spall 2000 in [51] p.9).

One straight forward application of the Robbins-Monro method yields a stochastic gradient-descent algorithm. We present variants of this algorithm for the monomodal and the multimodal registration task, both variants act on the Lie groups $SA(n)$ as well as on $SE(n)$. Another version of the Robbins-Monro method yields a stochastic Gauss-Newton method, which is available for the same registration tasks as before. In our numerical results the latter one shows a faster convergence behavior and it is more adaptive to the structure of the underlying Lie groups. We also develop an application of these algorithms to the multimodal and to the non-rigid registration task.

In comparison with the deterministic methods like the QMC- or SB-Newton, the stochastic algorithms are striking with their extraordinary small computation time, even if the images consist of huge 3D data. But there is a price to pay: Even for convex optimization problems on vector spaces, the Robbins-Monro methods only converge in probability. Thus, there is no stopping criterion which guarantees a certain amount of accuracy. Moreover, for non-convex problems, like the image registration task, there is a probability larger than zero that the algorithm will diverge. Hence, in practice, stochastic algorithms usually have a very pragmatic procedure: First, run a couple of iterations. Then either one takes the last iteration as the result or one rejects it and restart.

Nevertheless, in our experiments with real medical data, the stochastic methods constantly yield reasonable results. Besides, if we also take the computation time into consideration, the stochastic methods become an outstanding tool for the image registration task. Thus, we confirm a similar result of Klein et al. [52] who compared several registration methods using B-spline transformations.

Acknowledgment

First, I would like to thank my advisor Prof. Dr. Uwe Helmke for introducing me into that interesting field of image processing and for his support during the last five years of my work. Moreover, I would like to express my gratitude to PD. Dr. Otto Sauer from the department of radiation oncology. He provided me an insight into many application areas of medical image processing, supplied me with adequate pictorial material and supported me with his specialist knowledge and encouraging nature.

This work was supported in parts by the Interdisciplinary Center for Clinical Research (IZKF) through the project F-37-N (Organ Tracking) and by the Elite Network of Bavaria (ENB) with a position in the International Doctorate Program (IDP) in Engineering and Computer Sciences :“Identification, Optimization and Control with Applications in Modern Technologies”. I also send many thanks to all professors, students and members within our IDP.

My empathy and best wishes go to all my friends and colleagues who are still working on their phd-thesis: Hang on!

Chapter 2

Optimization Methods on Matrix Lie Groups

Constraint optimization is a well-established area of research in mathematics with an overwhelminged amount of applications and powerful techniques are available in the mathematical literature. Most of these techniques consider the constraints as a subset of a (higher dimensional) ambient vector space. In particular, the algorithms work on the embedded space and ignore the fact that the search space is unnecessarily high dimensional and that the constraint set underlie a geometrical structure, which can be used to set up the optimization algorithms. In the cases where the constraints form a manifold or a Lie group, Riemannian optimization techniques can be used in order to guarantee that the iteration stay on the constraint set and to exploit the including geometry. We refer to [29] for a recent overview of optimization algorithms on manifolds.

In this chapter, we will present a Newton and a stochastic algorithm for the optimization problem on Lie groups. Although the two approaches will work in this general setting, we will mainly focus on two special Lie groups: the Euclidean transformation group $SE(n)$ and the Special Affine Group $SA(n)$. The first section will give a short introduction in these groups. In Section 2.2 we will present the (μ, ν)-Newton algorithm, based on the work of Hüper and Trumpf [45]. Following the earlier works on Riemannian optimization on manifolds by Helmke and Moore [53], Shub [54], Manton [55], Helmke, Hüper and Trumpf [46] and also Absil [29], we use very simple local parameterizations of the Euclidean transformation group to compute an approximated version of the Hessian and to perform the Newton-step. In contrast to our approach, these previous works mainly deal with minimizing trace-functions on $SO(n)$ or on its homogeneous spaces. We show local quadratic convergence of the algorithms under suitable genericity conditions. Our algorithm for $SE(n)$ and $SA(n)$ seems to be new even in the case of minimizing standard trace functions.

In Section 2.3 we study optimization problems, where the evaluation of the cost function, and their derivatives are subject to stochastic perturbations. In the case of optimization on vector spaces, such perturbations can destroy the convergence of the Newton method, but variants such as the Robbins-Monro type algorithms are known to produce nevertheless satisfactory results. We refer to [56] or [57] for an overview in this field. We will extend a Robbins-Monro type algorithm, which was introduced by Benaïm and Hirsch in [47] for vector spaces, to Lie

groups.
More precisely, we assume that G is a Lie group equipped with a Riemannian metric. Then, a straight forward approach consists of iterations of the form

$$x_0 \in G, \quad x_{k+1} = \exp_{x_k}\left(\frac{1}{k+1}\left[\mathfrak{grad} f(x_k) + U_k\right]\right) \quad \text{for } k \in \mathbb{N}_0, \tag{2.1}$$

where $f : G \to \mathbb{R}$ is a smooth cost function, $\mathfrak{grad} f$ its gradient and $\exp_x$ is the Riemannian exponential map. In the perturbation-free case, i.e. when $U_k = 0$ for all $k \in \mathbb{N}$, this iteration is known as a gradient steepest descent algorithm, where the step size is given by the harmonic sequence. (See e.g. [58, 59] for a study of gradient-algorithms on manifolds.) In this section we focus on the influence of the perturbations U_k on the iteration (2.1). In particular, we will give a weak condition for $\{U_k\}_{k\in\mathbb{N}}$ such that any accumulation point of (2.1), if it exists, is still a critical point of f. Moreover, we will embed (2.1) in a more general setting, where we replace $\mathfrak{grad} f$ by a continuous vector field and $\exp_x$ by a local parameterization.

2.1 Preliminaries on Matrix Lie Groups

We introduce two special examples of matrix Lie groups, the affine volume-preserving transformations $SA(n)$ and the special Euclidean transformation $SE(n)$. For background on Lie groups and Lie algebras we refer to [60].

2.1.1 Affine Volume-Preserving Transformations

Affine transformations of $\mathbb{R}^n$ are maps of the form $x \mapsto Ax + t$ for a given matrix $A \in \mathbb{R}^{n\times n}$ and a translation vector $t \in \mathbb{R}^n$. In order to focus on volume-preserving transformations, we consider the special linear group $SL(n)$ of matrices in $\mathbb{R}^{n\times n}$ with determinant 1. The group of affine, volume-preserving transformations then can be identified with the Lie group of all $(n+1)\times(n+1)$-matrices of the form

$$M = \begin{pmatrix} A & t \\ 0 & 1 \end{pmatrix} \in \mathbb{R}^{(n+1)\times(n+1)}, \quad \text{with} \quad A \in SL(n),\ t \in \mathbb{R}^n \tag{2.2}$$

which defines the *special affine group* $SA(n)$. We want to point out that the group operation of $SA(n)$ is identical to the matrix-multiplication in $\mathbb{R}^{(n+1)\times(n+1)}$, while the set of volume-preserving transformations forms a group using the operation of composing functions. One can easily verify that both group representation are isomorphic. Additionally, using standard terminology from group theory, the construction in (2.2) states that $SA(n)$ is the semidirect product $SL(n) \ltimes \mathbb{R}^n$.
The associated Lie algebra $sa(n)$ of $SA(n)$ consists of all matrices of the form

$$\begin{pmatrix} \Omega & v \\ 0 & 0 \end{pmatrix} \in \mathbb{R}^{(n+1)\times(n+1)}, \quad \text{with} \quad \Omega \in sl(n),\ v \in \mathbb{R}^n, \tag{2.3}$$

where $sl(n)$ denotes the Lie algebra of $SL(n)$, i.e. the set of $n \times n$-matrices with trace zero. The corresponding Lie bracket is defined as

$$\left[\begin{pmatrix} \Omega_1 & v_1 \\ 0 & 0 \end{pmatrix}, \begin{pmatrix} \Omega_2 & v_2 \\ 0 & 0 \end{pmatrix}\right] := \begin{pmatrix} \Omega_1 & v_1 \\ 0 & 0 \end{pmatrix}\begin{pmatrix} \Omega_2 & v_2 \\ 0 & 0 \end{pmatrix} - \begin{pmatrix} \Omega_2 & v_2 \\ 0 & 0 \end{pmatrix}\begin{pmatrix} \Omega_1 & v_1 \\ 0 & 0 \end{pmatrix} \tag{2.4}$$

for all $\Omega_1, \Omega_2 \in sl(n)$ and $v_1, v_2 \in \mathbb{R}^n$. Moreover, we endow $sa(n)$ with the standard Euclidean product

$$\left\langle \begin{pmatrix} \Omega_1 & v_1 \\ 0 & 0 \end{pmatrix}, \begin{pmatrix} \Omega_2 & v_2 \\ 0 & 0 \end{pmatrix} \right\rangle_{sa(n)} := \operatorname{tr}(\Omega_1^\top \Omega_2) + v_1^\top v_2. \tag{2.5}$$

In the following sections, we will construct several algorithms on the Lie group $SA(n)$. In contrast to previous work as e.g. [10, 24], we will not use fixed local coordinates to parameterize the group. In our approach, we will use local parameterizations instead. Thus, our parameterizations change with the iteration points. This offers considerable advantages in the design of the algorithm. Recall that a local parameterization on an n-dimensional manifold $\mathcal{M}$ is a family $\{\mu_p\}_{p\in\mathcal{M}}$ of smooth maps $\mu_p : \mathbb{R}^n \to \mathcal{M}$ that satisfies $\mu_p(0) = p,\ p \in \mathcal{M}$, and defines a local diffeomorphism around 0. Such local parameterizations exist for every manifold and provide coordinate charts around each point of the manifold. On a Riemannian manifold, a standard set of local parameterizations is given by the so-called Riemannian normal coordinates that are defined by the Riemannian exponential map. In the sequel, our local parameterizations have the advantage of being more easily computable than the Riemannian exponential map, although they may not allow such immediate Riemannian geometry interpretations.

Let $\exp : \mathbb{R}^{n\times n} \to \mathbb{R}^{n\times n}$ denote the matrix-exponential map. Then, a set of local parameterizations of $SA(n)$ are then given by

$$\begin{aligned} &\mu_M : sl(n) \times \mathbb{R}^n \to SA(n) \\ &\mu_M(\Omega, v) := M \exp \begin{pmatrix} \Omega & v \\ 0 & 0 \end{pmatrix}, \end{aligned} \tag{2.6}$$

since μ satisfies $\mu_M(0,0) = M$ and $D\mu_M(0,0) = id$.

In order to construct a computationally more feasible local parameterization, we consider a first order approximation of this map. For this purpose, we decompose each Lie algebra element of $sl(n)$ by the decomposition

$$X = X_l + X_d + X_u$$

where X_d is a diagonal matrix and $X_u,\ X_l$ are strictly upper and lower triangular matrices, respectively. The next lemma will use this decomposition to construct a local parameterization of $SL(n)$ around the identity:

Lemma 2.1

Let X_Q denote the Q-factor in the QR-decomposition of an invertible matrix $X \in \mathbb{R}^{n\times n}$ and $X_d,\ X_u,\ X_l$ the corresponding diagonal, upper triangular and lower triangular matrices like it is defined before. Then, the map

$$\theta : sl(n) \to SL(n), \quad X \mapsto (I + X_l - X_l^\top)_Q \left[\exp(X_d) + X_u + X_l^\top\right]$$

is a local parameterization of $SL(n)$ with $D\theta(0) = id$.

Note that $I + \Omega$ is invertible for every skew-symmetric matrix Ω, thus, $(I + \Omega)_Q$ is always well defined. In particular, the function θ in Lemma 2.1 is well defined.

Proof
Since $\left[\exp(X_d) + X_u + X_l^\top\right]$ is an upper triangular matrix, we have

$$\det(\theta(X)) = \det\left(\left[\exp(X_d) + X_u + X_l^\top\right]\right) = \exp(\mathrm{tr}\ (X_d)) = 1$$

for all $X \in sl(n)$. Thus θ maps into $SL(n)$.
In order to prove $D\theta(0) = id$, we follow an argument of [46]: Let $Y \in so(n)$ be fixed and let $Q(t)$ and $R(t)$ denote the QR-decomposition of $I + tY, t \in \mathbb{R}$, i.e.

$$I + tY = Q(t)R(t), \quad \text{with} \quad Q(0) = I,\ R(0) = I.$$

We differentiate this equation with respect to t and get

$$Y = \dot{Q}(0) + \dot{R}(0).$$

Since Y and $\dot{Q}(0)$ are skew-symmetric, while $\dot{R}(0)$ is upper-triangular, we get $\dot{R}(0) = 0$. Thus,

$$\frac{d}{dt}(I + tY)_Q\big|_{t=0} = \dot{Q}(0) = Y. \tag{2.7}$$

Now, let $X \in sl(n)$ be fixed, then we have

$$\begin{aligned}\frac{d}{dt}\theta(tX_d)\big|_{t=0} &= \frac{d}{dt}\exp(tX_d)\big|_{t=0} = X_d,\\ \frac{d}{dt}\theta(tX_u)\big|_{t=0} &= \frac{d}{dt}\left[\exp(0) + tX_u\right]\big|_{t=0} = X_u\end{aligned}$$

and the equation (2.7) applied to $Y = X_l - X_l^\top$ yields

$$\frac{d}{dt}\theta(tX_l)\big|_{t=0} = \frac{d}{dt}(I + tX_l - tX_l^\top)\big|_{t=0} + \left[\exp(0) + tX_l^\top\right]\big|_{t=0} = X_l.$$

Thus, we have proven $D\theta(0) = id$. Therefore, θ is a local parameterization around $X = 0$. □

This lemma leads to the following system of local parameterization for $SA(n)$

$$\begin{aligned}&\nu_M^{QR} : sl(n) \times \mathbb{R}^n \to SA(n)\\ &\nu_M^{QR}(\Omega, v) := M\begin{pmatrix}\theta(\Omega) & (I + \frac{1}{2}\Omega)v\\ 0 & 1\end{pmatrix}.\end{aligned} \tag{2.8}$$

Lemma 2.2
The map ν_M^{QR} is a first order approximation of μ_M, i.e. we have

$$\nu_M^{QR}(0) = M = \mu_M(0) \quad \textit{and} \quad D\nu_M^{QR}(0) = D\mu_M(0) \quad \textit{for all } M \in SA(n).$$

Thus, ν_M^{QR} is a local parameterization of $SA(n)$.

Proof
By the definition of θ, we get

$$\nu_M^{QR}(0,0) = M \begin{pmatrix} \theta(0) & 0 \\ 0 & 1 \end{pmatrix} = MI = \nu_M(0,0).$$

Lemma 2.1 yields

$$\begin{aligned} \frac{d}{dt}\nu_M^{QR}(t\Omega, tv)\big|_{t=0} &= M \begin{pmatrix} \frac{d}{dt}\theta(t\Omega)\big|_{t=0} & \frac{d}{dt}[tv + \frac{1}{2}t^2\Omega v]\big|_{t=0} \\ 0 & 0 \end{pmatrix} \\ &= M \begin{pmatrix} \Omega & v \\ 0 & 0 \end{pmatrix} \\ &= \frac{d}{dt}\nu_M(t\Omega, tv)\big|_{t=0}. \end{aligned}$$

□

We end this subsection with an introduction of a special $SA(n)$ group action. By identifying a vector $x \in \mathbb{R}^n$ with its homogenous coordinates

$$\bar{x} = \begin{pmatrix} x \\ 1 \end{pmatrix},$$

the affine transformation $\rho_M : x \mapsto Ax + t$ can be represented by the map

$$\bar{x} \mapsto M\bar{x} = P \begin{pmatrix} A & t \\ 0 & 1 \end{pmatrix} \bar{x} \quad \text{with} \quad P = \begin{pmatrix} I_n & 0 \end{pmatrix} \in \mathbb{R}^{n\times(n+1)}.$$

Therefore, by varying $M \in SA(n)$ in ρ_M, the affine volume-preserving transformations induces a group action of $SA(n)$ on the vector space $\mathbb{R}^n$ by

$$\cdot : SA(n) \times \mathbb{R}^n \to \mathbb{R}^n, \quad M \cdot x := PM\bar{x}. \tag{2.9}$$

2.1.2 Rigid-Body Motions

An important subgroup of the volume-preserving transformations are the rigid-body motions. Here, we consider affine transformations $\rho : \mathbb{R}^n \to \mathbb{R}^n$, which can be decomposed into a rotation around the origin and a translation. Thus, we can write $\rho(x) = Ax + t$, where $A \in SO(n)$ is a rotation matrix. Here $SO(n)$ denotes the compact Lie group of $n \times n$ real matrices A satisfying $AA^\top = A^\top A = I$ and $\det A = 1$. Again, we can identify such rigid-body motions with the Lie group of all $(n+1) \times (n+1)$-matrices of the form

$$M = \begin{pmatrix} A & t \\ 0 & 1 \end{pmatrix} \in \mathbb{R}^{(n+1)\times(n+1)}, \quad \text{with} \quad A \in SO(n),\ t \in \mathbb{R}^n,$$

which defines the "Special Euclidean Group" $SE(n)$. Similarly to (2.3), the associated Lie Algebra $se(n)$ consists of all matrices of the form

$$\begin{pmatrix} \Omega & v \\ 0 & 0 \end{pmatrix} \in \mathbb{R}^{(n+1)\times(n+1)}, \quad \text{with} \quad \Omega \in so(n),\ v \in \mathbb{R}^n,$$

where $so(n)$ denotes the Lie algebra of the skew-symmetric $n \times n$ matrix. Thus, the Special Euclidean Group is a sub-Lie group of the Special Affine Group and the action of $SA(n)$ on $\mathbb{R}^n$ defined in (2.9) can be restricted to an action of $SE(n)$ on $\mathbb{R}^n$. The Lie bracket and the standard Euclidean product on $se(n)$ are also defined by restriction of the corresponding definitions in (2.4) and (2.5) on $sa(n)$.

Like in the previous subsection, we will introduce two sets of local parameterizations. For the first set, we consider again the matrix exponential map $\exp : se(n) \to SE(n)$, which provides us with a canonical map between the Lie algebra and the Lie group. This leads to the local parameterization around any $M \in SE(n)$ of the form (2.2) as

$$\begin{aligned} &\mu_M : so(n) \times \mathbb{R}^n \to SE(n) \\ &\mu_M(\Omega, v) := M \exp \begin{pmatrix} \Omega & v \\ 0 & 0 \end{pmatrix}. \end{aligned} \tag{2.10}$$

Note that the computation of the matrix exponential is expensive for large scale matrices but in the special cases of $n = 2, 3$ explicit formulas like the one of Rodriguez (cf. e.g. [28] p.27) are available.

By definition of the function $\theta : sl(n) \to SL(n)$ in Lemma 2.1, we get for the restriction $\theta|_{so(n)} : so(n) \to SO(n)$ the term $\theta|_{so(n)}(\Omega) = (I + \Omega)_Q$. Thus, the restriction of ν_M^{QR} in (2.8) to the Lie algebra $so(n) \times \mathbb{R}^n$ yields the following local parameterization of the $SE(n)$:

$$\begin{aligned} &\nu_M^{QR} : so(n) \times \mathbb{R}^n \to SE(n) \\ &\nu_M^{QR}(\Omega, v) := M \begin{pmatrix} (I + \Omega)_Q & (I + \frac{1}{2}\Omega)v \\ 0 & 1 \end{pmatrix}. \end{aligned} \tag{2.11}$$

Since this local chart ν_M^{QR} in (2.11) is just a restriction of the one in (2.8), we receive again the property that $D\nu_M^{QR}(0) = id$ holds, i.e. ν_M^{QR} is locally diffeomorphic around 0 and a valid first order approximation of the exponential map.

2.2 The (μ, ν)-Newton Algorithm

In this section, we propose a novel approximate-Newton algorithm for image registration that is based on the above mentioned local parameterizations. Our construction differs essentially from the well-known Riemannian Newton algorithm, which is based on knowledge of the geodesics to calculate the Hessian. Since the geodesics in $SA(n)$ are available only implicitly via the solutions of complicated nonlinear second-order differential equations, we try to avoid the Riemannian Newton method. Instead, we adapt a version of the approximate-Newton method as developed by Shub [54] and Hüper and Trumpf [45], which has been already applied successfully to several optimization problems (see e.g. [46]).

Let $\{\mu_M\}_{M \in G}$ be a set of local parameterizations of a Lie Group G. Thus, μ_M is defined on an open neighborhood $U \subset \mathbb{R}^n$ of $0 \in U$ such that $\mu_M : U \to G$ is locally diffeomorphic with $\mu_M(0) = M$. Additionally, we assume that $\mu(M, x) := \mu_M(x)$ is a smooth map. Let $\{\nu_M\}_{M \in G}$ be another set of local parameterizations, subject to the same conditions. The (μ, ν)-Newton iteration on G for a smooth objective function $\Phi : G \to \mathbb{R}$, then, is defined as

$$M_{k+1} = \nu_{M_k} \left(- \left(\mathrm{Hess}_{\Phi \circ \mu_{M_k}} (0) \right)^{-1} \nabla (\Phi \circ \mu_{M_k}) (0) \right), \qquad M_0 \in G, \tag{2.12}$$

where $\nabla h(0)$ and $\mathrm{Hess}_h(0)$ denote the standard gradient and Hesse operator of a smooth function $h : \mathbb{R}^n \to \mathbb{R}$, respectively. The iteration in (2.12) can be decomposed into the calculation of the Newton step $d = -(\mathrm{Hess}_{\Phi\circ\mu_{M_k}}(0))^{-1}\nabla(\Phi \circ \mu_{M_k})(0)$ and subsequent application of the local parameterization $M_{k+1} = \nu_{M_k}(d)$. The local parameterizations $\{\mu_M\}_{M\in G}$ are used in (2.12) to calculate a classical Newton-step in Euclidean coordinates of $\mathbb{R}^n$. The second parameterization $\{\nu_M\}_{M\in G}$ acts as a retraction from the tangent space onto the manifold to carry out the actual Newton step. A practical choice of μ can be made through the Riemannian exponential map, regarding the retraction ν any first order approximation of the exponential map would be sufficient. According to the special structure of (2.12) we will denote this iteration the (μ, ν)-Newton algorithm.
Local quadratic convergence of this method has been recently established; we refer to [45] or [46] for a proof.

Theorem 2.3
Let $\{\mu_M\}_{M\in\mathfrak{M}}$ and $\{\nu_M\}_{M\in\mathfrak{M}}$ be two set of local parameterizations of a smooth manifold $\mathfrak{M}$ such that $\mu(M, x) := \mu_M(x)$ and $\nu(M, x) := \nu_M(x)$ are smooth maps. Moreover, let $\Phi : \mathfrak{M} \to \mathbb{R}$ be three times continuously differentiable and $M^\star \in \mathfrak{M}$ a nondegenerate critical point of Φ.
Under the condition

$$D\mu_{M^\star}(0) = D\nu_{M^\star}(0)$$

there exists an open neighborhood $V \subset \mathfrak{M}$ of $M^\star$ such that the point sequence $\{M_k\}_{k\in\mathbb{N}_0}$ converges quadratically to $M^\star$ provided $M_0 \in V$.

2.2.1 Cost Functions with Group Actions

In many applications like computer vision, image processing or robotics, the problems, which include the Lie groups $SE(n)$ or $SA(n)$ share the following characteristic: The groups appear only in connection with their group action, like it is defined in (2.9). Therefore, we will now construct a quite general cost function form for these groups and apply Theorem 2.3.
We consider the set of functions

$$\mathcal{D}(\mathbb{R}^n) := \left\{ h \in C(\mathbb{R}^n, \mathbb{R}) \mid \sup_{x\in\mathbb{R}^n} |h(x)| e^{\|x\|^2} < \infty \right\} \tag{2.13}$$

and endow this set with the norm

$$\|h\|_{\mathcal{D}} := \sup_{x\in\mathbb{R}^n} |h(x)| e^{\|x\|^2}, \tag{2.14}$$

where $\| \,.\, \|$ denotes the Euclidean norm on $\mathbb{R}^n$. We denote with $\mathcal{D}'(\mathbb{R}^n)$ the dual space of $\mathcal{D}(\mathbb{R}^n)$, which consists of all continuous, linear maps of the form $\mathcal{D}(\mathbb{R}^n) \to \mathbb{R}$. Recall that both sets $\mathcal{D}(\mathbb{R}^n)$ and $\mathcal{D}'(\mathbb{R}^n)$ are Banach spaces.
In order to construct the cost function, let G denote either the Euclidean $SE(n)$ or the affine Lie group $SA(n)$. Moreover, let $f \in C(\mathbb{R}^n \times \mathbb{R}^n, \mathbb{R})$ be given such that the conditions

(i) $\mathrm{supp}(f) \subset \mathbb{R}^n \times \mathbb{R}^n$ is compact and

(ii) $f(\ .\ ,y) \in C^3$ for all $y \in \mathbb{R}^n$.

are satisfied. For a given functional $\varphi \in \mathcal{D}'(\mathbb{R}^n)$ we consider the cost function

$$\Phi : G \to \mathbb{R}, \quad \Phi(M) = \varphi\left(f \circ \varrho_M\right), \tag{2.15}$$

where $\varrho_M : \mathbb{R}^n \to \mathbb{R}^n \times \mathbb{R}^n$ denotes the function $\varrho_M(x) = (M \cdot x, x)$.
A few remarks are in order: Since f has bounded support the function $f \circ \varrho_M$ has compact support for each $M \in G$. Thus, we get $f \circ \varrho_M \in \mathcal{D}(\mathbb{R}^n)$ and the construction in (2.15) is well defined. At first glance, the weight $e^{\|x\|^2}$ in (2.14) seems to be unnecessary. However, it guarantees that the L^1-norm for each $h \in \mathcal{D}(\mathbb{R}^n)$ exists. Thus, the L^∞-norm for each $\varphi \in \mathcal{D}'(\mathbb{R}^n)$ exists. In particular, our further studies will include the case $\varphi(h) := \int_{\mathbb{R}^n} h(x)dx$, which is our main motivation. Of course, other weights in (2.14) may yield the same property. Moreover, we will see in the following lemma that the cost function Φ is also a C^3-map and we can apply the previously defined (μ,ν)-Newton algorithm.

Proposition 2.4

Let $I \subset \mathbb{R}$ be a non-empty, open interval, let $J \subset \mathbb{R}^n$ be a non-empty, compact subset and let $\varphi \in \mathcal{D}'(\mathbb{R}^n)$. Moreover, let $q : I \times J \to \mathbb{R}^n$ be a continuous function with supp$(q(x,\ .\ )) \subset J$ for all $x \in I$. If the partial derivative $\partial_1 q(x,y)$ of q with respect to the first entry exists for all $(x,y) \in I \times J$ and if $\partial_1 q$ is continuous, then the function

$$\tilde{\Phi}(x) := \varphi(q(x,\ .\ )) \tag{2.16}$$

is continuously differentiable with $\tilde{\Phi}'(x) = \varphi(\partial_1 q(x,\ .\ ))$.

Proof
Let $\varepsilon > 0$ and $x_0 \in I$ be arbitrarily given. Since φ is continuous, there exists $\delta > 0$ with

$$\|h\|_{\mathcal{D}} < \delta \quad \Rightarrow \quad |\varphi(h)| < \varepsilon \quad \text{for all } h \in \mathcal{D}(\mathbb{R}^n).$$

Due to the fact that $\partial_1 q$ is uniformly continuous on $(x_0-\varepsilon_1, x_0+\varepsilon_1) \times J$ if $\varepsilon_1 > 0$ is sufficiently small, there exists $\varepsilon_1 > 0$ with $\|\partial_1 q(x,\ .\ ) - \partial_1 q(x_0,\ .\ )\|_{\mathcal{D}} < \delta$ for all $|x - x_0| < \varepsilon_1$.
Let $x \in (x_0 - \varepsilon_1, x_0 + \varepsilon_1)$ and $y \in J$ be arbitrary. Thus, the *Mean Value Theorem* yields $\theta \in (x_0 - \varepsilon_1, x_0 + \varepsilon_1)$ with

$$\left|\frac{q(x,y) - q(x_0,y)}{x - x_0} - \partial_1 q(x_0,y)\right| e^{\|y\|^2} = |\partial_1 q(\theta,y) - \partial_1 q(x_0,y)|\, e^{\|y\|^2} < \delta.$$

Since $y \in J$ was arbitrarily chosen, we get

$$\left|\frac{\tilde{\Phi}(x) - \tilde{\Phi}(x_0)}{x - x_0} - \varphi(\partial_1 q(x_0,\ .\ ))\right| = \left|\varphi\left(\frac{q(x,\ .\ ) - q(x_0,\ .\ )}{x - x_0} - \partial_1 q(x_0,\ .\ )\right)\right| < \varepsilon$$

Hence, $\tilde{\Phi}(x) := \varphi(q(x,\ .\ ))$ is differentiable with $\tilde{\Phi}'(x) := \varphi(\partial_1 q(x,\ .\ ))$. In the same manner, one can show the continuity of $\tilde{\Phi}'$. □

Lemma 2.5
The function $\Phi : G \to \mathbb{R}$ defined in (2.15) is three times continuously differentiable.

Proof
Let $M \in G$ be arbitrarily chosen, $V \subset G$ an open neighborhood of M, $U \subset \mathbb{R}^m$ an open neighborhood of zero with $m = \dim G$ and $\mu_M : V \to U$ a chart of G with $\mu_M(M) = 0$. Thus, the function $\tilde{\Phi} : U \to \mathbb{R}$, $\tilde{\Phi} := \Phi \circ \mu_M^{-1}$ is of the form $\tilde{\Phi}(x) = \varphi(h(x, \ .\ ))$ with $h : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}^n$, $h(x,y) := f(\mu_M^{-1}(x)y, y)$.
It is sufficient to show that $\tilde{\Phi}$ is three times continuously differentiable in $x = 0$. Since each partial derivative of $\tilde{\Phi}$ is of the form (2.16), we can iteratively apply Proposition 2.4. This yields that $\tilde{\Phi}$ is three times partially differentiable and all third order partial derivatives are continuous. Thus, $\tilde{\Phi}$ is three times differentiable in $x = 0$. □

In order to simplify the notation, we will often represent the functional φ with the L^2-inner product and write

$$\Phi : G \to \mathbb{R}, \quad \Phi(M) = \int_{\mathbb{R}^n} \varphi(x) f(M \cdot x, x) dx := \varphi(f \circ \varrho_M). \tag{2.17}$$

We want to point out that $\varphi(x)$ is not to be mistaken with a function. Like the well-known Dirac delta impulse, it is a distribution; the term "$\varphi(x)$" is on its own merits meaningless.
In this work, we will particularly be interested in two special cases of this cost function. In the first one, we will set φ to the integration over the entire vector space $\mathbb{R}^n$ and get

$$\Phi : G \to \mathbb{R}, \quad \Phi(M) = \int_{\mathbb{R}^n} f(M \cdot x, x) dx.$$

In the second case, we will choose a finite set of sample-points $\{x_i\}_{i=1,\dots,N} \subset \mathbb{R}^n$ and set φ to the sum over the evaluations of $f(M \cdot x, x)$ in these points, i.e.

$$\Phi : G \to \mathbb{R}, \quad \Phi(M) = \sum_{i=1}^{N} f(M \cdot x_i, x_i).$$

The aim of this subsection is to compute the Newton-iteration (2.12) for the cost function (2.15). First, we have to decide which local parameterization we want to use for μ and ν in the iteration (2.12). In this work, $\{\mu_M\}$ is chosen as the exponential map, while (2.11) and (2.8) are chosen for the retraction map $\{\nu_M\}_{M \in G}$ in $SA(n)$ and $SE(n)$, respectively. In this combination, neither the Riemannian nor the matrix exponential map has to be evaluated at a point different from zero. In order to reduce the numerical costs, we use the QR-factorizations in (2.11) and (2.8) for the update part of the iteration.
Therefore, we have to consider the function $\Phi \circ \mu_M$ and its first and second derivatives in each iteration.

Lemma 2.6
Let G denote either the Euclidean or affine Lie group $SE(n) = SO(n) \ltimes \mathbb{R}^n$, $SA(n) = SL(n) \ltimes \mathbb{R}^n$, respectively. Let μ_M denote the local parameterizations (2.6), (2.10) and let Φ

denote the cost function (2.17). Endow the Lie algebras $\mathfrak{g} = sa(n), se(n)$ with their standard Euclidean inner product defined in (2.5). We will use the notaion $\nabla_1 f(x, y) \in \mathbb{R}^n$ for the gradient of $f(\, . \, , y)$ in x, when $y \in \mathbb{R}^n$ is fixed. Analogous, $H_1 f(x, y) \in \mathbb{R}^{n \times n}$ denotes the Hessian matrix of $f(\, . \, , y)$ in x, when $y \in \mathbb{R}^n$ is fixed. For a fixed $M \in G$, we get:

(a) The gradient of $\Phi \circ \mu$ in 0 is $\nabla(\Phi \circ \mu_M)(0, 0) = (\tilde{\Omega}, \tilde{v})$ with

$$\tilde{\Omega} = \int\limits_{\mathbb{R}^n} \varphi(M^{-1} \cdot z) \pi_{\mathfrak{k}} \left(\nabla_1 f(z, M^{-1} \cdot z) z^\top \right) dx, \quad \tilde{v} = \int\limits_{\mathbb{R}^n} \varphi(M^{-1} \cdot z) \nabla_1 f(z, M^{-1} \cdot z) dz. \tag{2.18}$$

Here $\pi_{\mathfrak{k}}$ denotes the projection from $gl(n)$ on the Lie algebra $\mathfrak{k} = so(n)$ for $G = SE(n)$ and $\mathfrak{k} = sl(n)$ for $G = SA(n)$, respectively:

$$\pi_{so(n)}(X) := \frac{1}{2}(X - X^\top), \qquad \pi_{sl(n)}(X) := X - \frac{tr\, X}{n} I_n.$$

(b) At a critical point $M \in G$, the Hessian operator $Hess_{\Phi \circ \mu_M}(0) : \mathfrak{g} \to \mathfrak{g}$ of $\Phi \circ \mu_M$ is $Hess_{\Phi \circ \mu_M}(0)(\Omega, v) = (\hat{\Omega}, \hat{v})$ with

$$\begin{aligned} \hat{\Omega} = \pi_{\mathfrak{k}} \Bigg(& \frac{1}{2} \Omega^\top \int\limits_{\mathbb{R}^n} \varphi(M^{-1} z) \nabla_1 f(z, M^{-1} z) z^\top dz + \frac{1}{2} \int\limits_{\mathbb{R}^n} \varphi(M^{-1} z) \nabla_1 f(z, M^{-1} z) z^\top dz \Omega^\top \\ & + \int\limits_{\mathbb{R}^n} \varphi(M^{-1} z) H_1 f(z, M^{-1} z) \Omega z z^\top dz + \int\limits_{\mathbb{R}^n} \varphi(M^{-1} z) H_1 f(z, M^{-1} z) v z^\top dz \Bigg) \\ \hat{v} = & \int\limits_{\mathbb{R}^n} \varphi(M^{-1} z) H_1 f(z, M^{-1} z)(\Omega z + v) dz. \end{aligned} \tag{2.19}$$

Proof
Let $v, t \in \mathbb{R}^n$. For the case $G = SE(n)$ let $\Omega \in so(n)$ and $A \in SO(n)$ be arbitrary. For the case $G = SA(n)$ let $\Omega \in sl(n)$ and $A \in SL(n)$. We set

$$\Omega_0 = \begin{pmatrix} \Omega & v \\ 0 & 0 \end{pmatrix}, \quad M = \begin{pmatrix} A & t \\ 0 & 0 \end{pmatrix} \text{ and } P = (I_n \ 0) \in \mathbb{R}^{n \times (n+1)}.$$

Then, the directional derivative of $\Phi \circ \mu_M$ is given by

$$\frac{d}{d\tau} \Phi \circ \mu_M(\tau\Omega, \tau v) = \int\limits_{\mathbb{R}^n} \varphi(x) \nabla_1 f \left(P \exp(\tau\Omega_0) M \bar{x}, x \right)^\top P \Omega_0 \exp\left(\tau\Omega_0\right) M \bar{x} dx. \tag{2.20}$$

Hence

$$\frac{d}{d\tau}\Big|_{\tau=0} \Phi \circ \mu_M(\tau\Omega, \tau v) = \int\limits_{\mathbb{R}^n} \varphi(x) \nabla_1 f(Ax + t, x)^\top \left(\Omega(Ax + t) + v \right) dx \tag{2.21}$$

$$\frac{d^2}{d\tau^2}\Big|_{\tau=0}\Phi\circ\mu_M(\tau\Omega,\tau v) = \int\limits_{\mathbb{R}^n}\varphi(x)\left(\Omega(Ax+t)+v\right)^\top \mathrm{H}_1 f(Ax+t,x)\left(\Omega(Ax+t)+v\right)dx$$
$$+\int\limits_{\mathbb{R}^n}\varphi(x)\nabla_1 f(Ax+t,x)^\top\left(\Omega^2(Ax+t)+\Omega v\right)dx. \tag{2.22}$$

After substituting $z = Ax + t$ we get

$$\frac{d}{d\tau}\Big|_{\tau=0}\Phi\circ\mu_M(\tau\Omega,\tau v) =$$
$$\operatorname{tr}\left[\int\limits_{\mathbb{R}^n}\varphi(M^{-1}z)z\nabla_1 f(z,M^{-1}z)^\top dz\Omega\right]+\left\langle\int\limits_{\mathbb{R}^n}\varphi(M^{-1}z)\nabla_1 f(z,M^{-1}z)dz, v\right\rangle.$$

Here, $\langle\ ,\ \rangle$ is the standard Euclidean product in $\mathbb{R}^n$. Since the gradient $(\tilde{\Omega},\tilde{v})$ is the unique vector of the tangent space with

$$\frac{d}{d\tau}\Big|_{\tau=0}\Phi\circ\mu_M(\tau\Omega,\tau v) = \operatorname{tr}\,(\Omega^\top\tilde{\Omega}) + v^\top\tilde{v}$$

we have proven (2.18).

In order to calculate the Hessian of $\Phi\circ\mu_M$ in zero, we substitute $z = Ax + t$ in Formula (2.22) and get

$$\frac{d^2}{d\tau^2}\Big|_{\tau=0}\Phi\circ\mu_M(\tau\Omega,\tau v) = \int\limits_{\mathbb{R}^n}\varphi(M^{-1}z)(\Omega z+v)^\top \mathrm{H}_1(z,M^{-1}z)(\Omega z+v)dz$$
$$+\int\limits_{\mathbb{R}^n}\varphi(M^{-1}z)\nabla_1 f(z,M^{-1}z)^\top\Omega^2 z dz$$
$$+\int\limits_{\mathbb{R}^n}\varphi(M^{-1}z)\nabla_1 f(z,M^{-1}z)^\top dz\Omega v.$$

Since the last summand is equal to $\tilde{v}\Omega v$ it vanishes at a critical point. Therefore, we obtain the Hessian $\mathcal{H}$ by polarizing the two first summands

$$\mathcal{H}_{\Phi\circ\mu_M(0)}(\Omega,v)(\hat{\Omega},\hat{v}) = \int\limits_{\mathbb{R}^n}\varphi(M^{-1}z)(\Omega z+v)^\top \mathrm{H}_1 f(z,M^{-1}z)(\hat{\Omega}z+\hat{v})dz$$
$$+\frac{1}{2}\operatorname{tr}\left[\left(\int\limits_{\mathbb{R}^n}\varphi(M^{-1}z)z\nabla_1 f(z,M^{-1}z)^\top dz\right)(\Omega\hat{\Omega}+\hat{\Omega}\Omega)\right]$$

which proves (2.19). □

Note that (2.19) yields the Hessian of $\Phi\circ\mu_M$ in 0 only at a critical point $M \in G$. In the sequel, we will use the same formula at an arbitrary point $M \in G$ and obtain a **modified**

Newton algorithm for Φ. Thus, the modified Newton-step in (2.12) requires to solve the following system of linear equations:

$$\int\limits_{\mathbb{R}^n} \varphi(M^{-1}z)\mathrm{H}_1 f(z,M^{-1}z)(\Omega z+v)dz = -\int\limits_{\mathbb{R}^n} \varphi(M^{-1}z)\nabla_1 f(z,M^{-1}z)dz \tag{2.23}$$

and

$$\begin{aligned}\pi_{\mathfrak{k}}\Bigg(\frac{1}{2}\Omega^\top \int\limits_{\mathbb{R}^n} \varphi(M^{-1}z)\nabla_1 f(z,M^{-1}z)z^\top dz + \frac{1}{2}\int\limits_{\mathbb{R}^n} \varphi(M^{-1}z)\nabla_1 f(z,M^{-1}z)z^\top dz\Omega^\top \\ + \int\limits_{\mathbb{R}^n} \varphi(M^{-1}z)\mathrm{H}_1 f(z,M^{-1}z)\Omega zz^\top dz + \int\limits_{\mathbb{R}^n} \varphi(M^{-1}z)\mathrm{H}_1 f(z,M^{-1}z)vz^\top dz)\Bigg) \\ = -\int\limits_{\mathbb{R}^n} \varphi(M^{-1}z)\pi_{\mathfrak{k}}(\nabla_1 f(z,M^{-1}z)z^\top)dz\end{aligned} \tag{2.24}$$

with the unknowns $(\Omega, v) \in \mathfrak{g}$.
In order to rewrite (2.23) and (2.24) as a linear equation in the components v_i and $\Omega_{i,j}$, we first focus on the Euclidean transformation group. Here, with $\Omega \in so(n)$ we obtain:

$$\int\limits_{\mathbb{R}^n} \varphi(M^{-1}z)\mathrm{H}_1 f(z,M^{-1}z)(\Omega z+v)dz = -\int\limits_{\mathbb{R}^n} \varphi(M^{-1}z)\nabla_1 f(z,M^{-1}z)dz \tag{2.25}$$

and

$$\begin{aligned}&\frac{1}{2}\int\limits_{\mathbb{R}^n} \varphi(M^{-1}z)\left(-\Omega\nabla_1 f(z,M^{-1}z)z^\top - z\nabla_1 f(z,M^{-1}z)^\top\Omega\right)dz \\ &+\frac{1}{2}\int\limits_{\mathbb{R}^n} \varphi(M^{-1}z)\left(-\nabla_1 f(z,M^{-1}z)z^\top\Omega - \Omega z\nabla_1 f(z,M^{-1}z)^\top\right)dz \\ &+\int\limits_{\mathbb{R}^n} \varphi(M^{-1}z)\left(\mathrm{H}_1 f(z,M^{-1}z)vz^\top - zv^\top\mathrm{H}_1 f(z,M^{-1}z)\right)dz \\ &+\int\limits_{\mathbb{R}^n} \varphi(M^{-1}z)\left(\mathrm{H}_1 f(z,M^{-1}z)\Omega zz^\top + zz^\top\Omega\mathrm{H}_1 f(z,M^{-1}z)\right)dz \\ &\qquad = -\int\limits_{\mathbb{R}^n} \varphi(M^{-1}z)(\nabla_1 f(z,M^{-1}z)z^\top - z\nabla f(z,M^{-1}z)^\top)dz.\end{aligned} \tag{2.26}$$

To evaluate the components of this system of linear equations, we use the abbreviations

$$\begin{aligned}\alpha_i &= \int\limits_{\mathbb{R}^n} \varphi(z)\frac{\partial f}{\partial x_i}(Az+t,z)dz, \\ \beta_{i,j} &= \int\limits_{\mathbb{R}^n} \varphi(z)(Az+t)_j\frac{\partial f}{\partial x_i}(Az+t,z)dz,\end{aligned}$$

$$\gamma_{i,j,k} = \int\limits_{\mathbb{R}^n} \varphi(z)(Az+t)_i \frac{\partial^2 f}{\partial x_j \partial x_k}(Az+t,z)dz, \tag{2.27}$$

$$\delta_{i,j,k,l} = \int\limits_{\mathbb{R}^n} \varphi(z)(Az+t)_i(Az+t)_j \frac{\partial^2 f}{\partial x_k \partial x_l}(Az+t,z)dz,$$

$$\epsilon_{i,j} = \int\limits_{\mathbb{R}^n} \varphi(z) \frac{\partial^2 f}{\partial x_i \partial x_j}(Az+t,z)dz,$$

for $1 \leq i,j,k,l \leq n$. To avoid confusion, here, the partial derivative of $f(x,y)$ are always taken with respect to the first component x. In particular, we have

$$\nabla_1 f(x,y) = \left(\frac{\partial f}{\partial x_i}(x,y)\right)_{i=1,\ldots,n} \quad \text{and} \quad \mathrm{H}_1 f(x,y) = \left(\frac{\partial^2 f}{\partial x_i \partial x_j}(x,y)\right)_{i,j=1,\ldots,n}.$$

We obtain:

Lemma 2.7

Let $(\Omega, v) \in so(n) \times \mathbb{R}^n$ be the modified Newton-direction for the cost function (2.17) in a point $M \in SE(n)$. Then, the components $\Omega_{k,l}$, $1 \leqslant k,l \leqslant n$ of Ω and v_k, $1 \leqslant k \leqslant n$ of v satisfy

$$\sum_{k>l}(\gamma_{l,k,i} - \gamma_{k,l,i})\Omega_{k,l} + \sum_k \epsilon_{i,k} v_k = -\alpha_i \tag{2.28}$$

for $1 \leqslant i \leqslant n$ and

$$\begin{aligned} \frac{1}{2}\sum_{k>j}(\beta_{i,k}+\beta_{k,i})\Omega_{k,j} - \frac{1}{2}\sum_{k<j}(\beta_{i,k}+\beta_{k,i})\Omega_{j,k} - \frac{1}{2}\sum_{k>i}(\beta_{j,k}+\beta_{k,j})\Omega_{k,i} \\ +\frac{1}{2}\sum_{k<i}(\beta_{j,k}+\beta_{k,j})\Omega_{i,k} - \sum_{k>l}(\delta_{i,k,l,j} - \delta_{j,l,k,i} + \delta_{i,l,k,j} - \delta_{i,k,l,j})\Omega_{k,l} \\ - \sum_k(\gamma_{j,k,i} - \gamma_{i,k,j})v_k = \beta_{i,j} - \beta_{j,i} \end{aligned} \tag{2.29}$$

for $1 \leqslant i < j \leqslant n$.

Note that the unknowns of this system are v_i and $\Omega_{i,j}$ for $i > j$. Therefore, a unique solution of the linear system corresponds to a unique element of the $so(n)$. We summarized the (μ,ν)-Newton algorithm for minimizing the cost function (2.17) with respect to the Euclidean group in Table 2.1.

Let us return to the case of volume-preserving transformations $G = SA(n)$. The modified Newton-equation (2.23) and (2.24) has now the form:

$$\int\limits_{\mathbb{R}^n} \varphi(M^{-1}z)\mathrm{H}_1 f(z, M^{-1}z)(\Omega z + v)dz = -\int\limits_{\mathbb{R}^n} \varphi(M^{-1}z)\nabla_1 f(z, M^{-1}z)dz \tag{2.30}$$

Table 2.1: Approximate-Newton Algorithm for SE(n) Group Actions

Step 1.
Make an initial guess $M_0 \in SE(n)$ and set $m = 0$.

Step 2.
Calculate $\alpha_i,\ \beta_{i,j},\ \gamma_{i,j,k},\ \delta_{i,j,k,l}$ and $\epsilon_{i,j}$ for all $1 \leqslant i, j, k, l \leqslant n$ as defined in equation (2.27).

Step 3.
Solve the linear system which consists of the equations

$$\sum_{k>l} (\gamma_{l,k,i} - \gamma_{k,l,i})\Omega_{k,l} + \sum_{k} \epsilon_{i,k} v_k = -\alpha_i \quad \text{for all } 1 \leqslant i \leqslant n$$

and

$$\begin{aligned} \sum_{k>j} \frac{1}{2}(\beta_{i,k} + \beta_{k,i})\Omega_{k,j} - \sum_{k<j} \frac{1}{2}(\beta_{i,k} + \beta_{k,i})\Omega_{j,k} - \sum_{k>i} \frac{1}{2}(\beta_{j,k} + \beta_{k,j})\Omega_{k,i} \\ + \sum_{k<i} \frac{1}{2}(\beta_{j,k} + \beta_{k,j})\Omega_{i,k} - \sum_{k>l}(\delta_{i,k,l,j} - \delta_{j,l,k,i} + \delta_{i,l,k,j} - \delta_{i,k,l,j})\Omega_{k,l} \\ - \sum_{k}(\gamma_{j,k,i} - \gamma_{i,k,j})v_k = \beta_{i,j} - \beta_{j,i} \quad \text{for all } 1 \leqslant i < j \leqslant n \end{aligned}$$

with the unknowns v_i and $\Omega_{i,j},\ j > i$.

Step 4.
Construct the $n \times n$ matrix Ω with the entries $\Omega_{i,j}$. In the case of $j > i$ use the solution of Step 3. Else, set

$$\Omega_{i,j} = \begin{cases} -\Omega_{j,i} & \text{for } j < i \\ 0 & \text{for } j = i. \end{cases}$$

Compute

$$M_{m+1} := \nu^{QR}_{M_m}(\Omega, v),$$

where ν^{QR} is defined in (2.11).

Step 5.
Set $m = m + 1$ and goto Step 2.

and

$$\begin{aligned}
&\frac{1}{2}\Omega^\top \int\limits_{\mathbb{R}^n} \varphi(M^{-1}z)\nabla_1 f(z, M^{-1}z)z^\top dz + \frac{1}{2}\int\limits_{\mathbb{R}^n} \varphi(M^{-1}z)\nabla_1 f(z, M^{-1}z)z^\top dz\Omega^\top \\
&+ \int\limits_{\mathbb{R}^n} \varphi(M^{-1}z)\mathrm{H}_1 f(z, M^{-1}z)\Omega z z^\top dz + \int\limits_{\mathbb{R}^n} \varphi(M^{-1}z)\mathrm{H}_1 f(z, M^{-1}z) v z^\top dz \\
&-\frac{1}{n} I_n \int\limits_{\mathbb{R}^n} \varphi(M^{-1}z)\left(z^\top\Omega^\top\nabla_1 f(z, M^{-1}z) + z^\top \Omega \mathrm{H}_1 f(z, M^{-1}z)z + z^\top \mathrm{H}_1 f(z, M^{-1}z)v\right) dz \\
&\qquad = -\int\limits_{\mathbb{R}^n} \varphi(M^{-1}z)\nabla_1 f(z, M^{-1}z)z^\top dz + \frac{1}{n} I_n \int\limits_{\mathbb{R}^n} \varphi(M^{-1}z)z^\top \nabla_1 f(z, M^{-1}z)dz.
\end{aligned} \tag{2.31}$$

Again, we can calculate the components of this system using the coefficients in (2.27). We end up with an analogue version of Lemma 2.7 in the case of a volume-preserving transformations.

Lemma 2.8
Let $(\Omega, v) \in sl(n) \times \mathbb{R}^n$ be the modified Newton-direction for the cost function (2.17) in a point $M \in SA(n)$. Then the components $\Omega_{k,l}$, $1 \leqslant k, l \leqslant n$, $(k,l) \neq (n,n)$ of Ω and v_k of v satisfy for each $1 \leqslant i \leqslant n$

$$\sum_{k \neq l} \gamma_{l,k,i}\Omega_{k,l} + \sum_{k \neq n}(\gamma_{k,k,i} - \gamma_{n,n,i})\Omega_{k,k} + \sum_k \epsilon_{i,k} v_k = -\alpha_i \tag{2.32}$$

and for all $1 \leqslant i, j \leqslant n$, $(i,j) \neq (n,n)$ the following equations:

$$\begin{aligned}
\frac{1}{2}\sum_k \beta_{i,k}\Omega_{j,k} + \frac{1}{2}\sum_k \beta_{k,j}\Omega_{k,i} + \sum_{(k,l)\neq(n,n)} \delta_{j,l,k,i}\Omega_{k,l}& \\
-\delta_{j,n,n,i}\sum_{k\neq n}\Omega_{k,k} + \sum_k \gamma_{j,k,i} v_k = -\beta_{i,j}&
\end{aligned} \tag{2.33}$$

for $i \neq n,\ j \neq n,\ i \neq j$,

$$\begin{aligned}
\frac{1}{2}\sum_k \beta_{n,k}\Omega_{j,k} + \frac{1}{2}\sum_{k\neq n} \beta_{k,j}\Omega_{k,n} + \sum_{(k,l)\neq(n,n)} \delta_{j,l,k,n}\Omega_{k,l}& \\
-(\delta_{j,n,n,i} + \frac{1}{2}\beta_{n,j})\sum_{k\neq n}\Omega_{k,k} + \sum_k \gamma_{j,k,n} v_k = -\beta_{n,j}&
\end{aligned} \tag{2.34}$$

for $i = n,\ j \neq n$,

$$\begin{aligned}
\frac{1}{2}\sum_{k\neq n} \beta_{i,k}\Omega_{n,k} + \frac{1}{2}\sum_k \beta_{k,n}\Omega_{k,i} + \sum_{(k,l)\neq(n,n)} \delta_{n,l,k,i}\Omega_{k,l}& \\
-(\delta_{n,n,n,i} + \frac{1}{2}\beta_{i,n})\sum_{k\neq n}\Omega_{k,k} + \sum_k \gamma_{n,k,i} v_k = -\beta_{i,n}&
\end{aligned} \tag{2.35}$$

Table 2.2: Approximate-Newton Algorithm for SA(n) Group Actions

Step 1.
Make an initial guess $M_0 \in SA(n)$ and set $m = 0$.

Step 2.
Calculate $\alpha_i,\ \beta_{i,j},\ \gamma_{i,j,k},\ \delta_{i,j,k,l}$ and $\epsilon_{i,j}$ for all $1 \leqslant i, j, k, l \leqslant n$ as defined in equation (2.27).

Step 3.
Solve the linear system described in (2.32) - (2.36) with the unknowns v_i and $\Omega_{i,j}$, $(i, j) \neq (n, n)$.

Step 4.
Construct the $n \times n$ matrix Ω with the entries $\Omega_{i,j}$. In the case of $(j, i) \neq (n, n)$ use the solution of Step 3. Else, set

$$\Omega_{n,n} = -\sum_{k \neq n} \Omega_{k,k}$$

and compute

$$M_{m+1} := \nu^{QR}_{M_m}(\Omega, v),$$

where ν^{QR} is defined in (2.8).

Step 5.
Set $m = m + 1$ and goto Step 2.

for $j = n,\ i \neq n$ *and*

$$\frac{1}{2}\sum_k \beta_{i,k}\Omega_{j,k} + \frac{1}{2}\sum_k \beta_{k,j}\Omega_{k,i} + \sum_{(k,l)\neq(n,n)} \left(\delta_{j,l,k,i} - \frac{1}{n}\left(\beta_{k,l} + \sum_m \delta_{l,m,k,m}\right)\right)\Omega_{k,l}$$
$$- \left(\delta_{j,n,n,i} - \frac{1}{n}\left(\beta_{n,n} + \sum_m \delta_{n,m,n,m}\right)\right)\sum_{k\neq n}\Omega_{k,k} + \sum_k \left(\gamma_{j,k,i} - \frac{1}{n}\sum_l \gamma_{l,l,k}\right) v_k$$
$$= -\beta_{i,j} + \frac{1}{n}\sum_k \beta_{k,k} \qquad (2.36)$$

for $i = j,\ i, j \neq n$.

We summarized the (μ, ν)-Newton algorithm for minimizing the cost function (2.17) with respect to the Euclidean group in Table 2.2.
We want to point out that the Newton step presented uses an approximated version of the

Hessian. However, at each critical point of the cost function we have an exact evaluation of the Hessian. Approximations of the Hessian appear also in a couple of classical algorithms like the Gauss-Newton method, the Levenberg-Marquard algorithm (see e.g. [29] for an overview) or the optimization technique presented in [61]. In contrast to our method, such algorithms do not perform a Newton step at a critical point and are not necessarily local quadratically convergent.

2.3 Stochastic Optimization Methods

The introduced (μ, ν)-Newton algorithm in the previous section has a basic drawback: It is assumed that we can exactly evaluate or measure the cost function Φ and its derivatives. In some application, these measurements are, however, corrupted by noise. It may also happen that an exact measurement is possible, but it might require extraordinary much computation time. In such cases, deterministic optimization methods are not the best choice.

In this section, we will introduce a special stochastic optimization method, the Robbins-Monro algorithm, in order to tackle these kinds of problems. In the literature several approaches are known to introduce the Robbins-Monro algorithm and they usually lead to different convergence conditions; we refer to [56] or [57] for an overview. Throughout this section we will follow the approach examined by Benaïm and Hirsch in [47] and extend their results to Lie groups.

In Subsection 2.3.1 and 2.3.2 we lay the ground for this approach. We study a quite general iteration on Lie groups, which is subject to a time-dependent vector field of perturbations. In Subsection 2.3.3 we will use this result for optimization problems on manifolds. In the last subsection of this chapter, we focus on special kinds of stochastic perturbations and extend the classical Robbins-Monro algorithm on manifolds.

2.3.1 Preliminaries on Asymptotic Pseudotrajectories

In this subsection, we recall the definition and basic properties of asymptotic pseudotrajectories. For more details we refer to [47]. Throughout this work we use the abbreviation $\mathbb{R}^+ := [0, \infty)$.

Definition 2.1 (Semiflow)
Let $(\mathcal{M}, d)$ denote a metric space. A semiflow on $\mathcal{M}$ is a continuous function $\Phi : \mathbb{R}^+ \times \mathcal{M} \to \mathcal{M}$ with the properties

$$\begin{aligned} \Phi(0, x) &= x, \quad \text{for all } x \in \mathcal{M} \text{ and} \\ \Phi(s, \Phi(t, x)) &= \Phi(s + t, x), \quad \text{for all } x \in \mathcal{M},\ s, t \in \mathbb{R}^+. \end{aligned}$$

We use the abbreviation $\Phi_t(x) := \Phi(t, x)$ and $\Phi^x(t) := \Phi(t, x)$. The set S_Φ denotes all trajectories of Φ, i.e.

$$S_\Phi := \{\Phi^x(\, . \,) \mid x \in \mathcal{M}\} \subset C^0(\mathbb{R}^+, \mathcal{M}).$$

Recall the case, where $\mathcal{M}$ coincides with a Riemannian manifold. Then, there exists a connection between semiflows and vector fields: Each complete vector field $F : \mathcal{M} \to T\mathcal{M}$

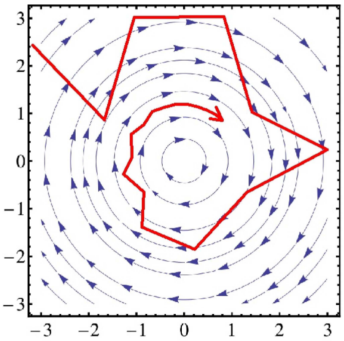


Figure 2.1: Roughly sketched example of a vector field (blue) in $\mathbb{R}^2$ and the image of an asymptotic pseudotrajectory of the corresponding semiflow (red).

on $\mathcal{M}$ with tangent bundle $T\mathcal{M}$ has a unique solution $\alpha_x : \mathbb{R}^+ \to \mathcal{M}$ for each initial value problem of the form

$$\dot{\alpha}_x(t) = F(\alpha_x(t)), \quad , \alpha_x(0) = x.$$

Exploiting the uniqueness of these solutions, we can verify that

$$\Phi(t, x) := \alpha_x(t) \tag{2.37}$$

satisfies the semiflow properties. Thus (2.37) yields an injective map between the complete vector fields and the semiflows.

Definition 2.2 (Asymptotic Pseudotrajectory)
Let $(\mathcal{M}, d)$ denote a metric space. A continuous function $X : \mathbb{R}^+ \to \mathcal{M}$ is an asymptotic pseudotrajectory of a semiflow $\Phi : \mathbb{R}^+ \times \mathcal{M} \to \mathcal{M}$ if

$$\lim_{t\to\infty} \sup_{0\leqslant h\leqslant T} d(X(t+h), \Phi_h(X(t))) = 0$$

for any $T > 0$.

This definition of asymptotic pseudotrajectories was first given by Benaïm and Hirsch [47]. Let $\mathcal{M}$ be a metric space and let $X : \mathbb{R}^+ \to \mathcal{M}$ be an asymptotic pseudotrajectory of a given semiflow $\Phi : \mathbb{R}^+ \times \mathcal{M} \to \mathcal{M}$. The Definition 2.2 implies heuristically that the curve X do not need to be a trajectory of Φ, but each part $\{X(t+h) \mid 0 \le h \le T\}$, $T > 0$ will more and more behave like a part of a trajectory of Φ for $t \to \infty$. We will give an accurate description of this statement in Theorem 2.9. A roughly sketched example of an asymptotic pseudotrajectory is given in Fig. 2.1.
Let the map $\theta^t : C(\mathbb{R}^+, \mathcal{M}) \to C(\mathbb{R}^+, \mathcal{M})$, $t > 0$ denote the translation of curves in t time units, i.e.

$$\theta^t(X)(s) := X(t+s).$$

If X is an asymptotic pseudotrajectory and $T > 0$ is fixed, then $\theta^t(X)(h)$ shadows $\Phi_h(X(t))$ on the interval $[0, T]$ and the error in $[0, T]$ tend uniformly to zero if t is increasing. The proof of the following theorem can be found in [47]:

Theorem 2.9
Let $(\mathcal{M}, d)$ be a metric space and let $X : \mathbb{R}^+ \to \mathcal{M}$ be a continuous function whose image has compact closure in $\mathcal{M}$. Moreover, let the function space $C^0(\mathbb{R}^+, \mathcal{M})$ be endowed with the metic

$$\tilde{d}(f, g) : \sum_{k\in\mathbb{N}} \frac{1}{2^k} \min(1, \tilde{d}_k(f, g)), \quad \text{for all } f, g \in C^0(\mathbb{R}^+, \mathcal{M}),$$

where $\tilde{d}_k(f, g) = \sup_{t\in[0,k]} d(f(t), g(t))$. Consider the following assertions:

(i) X is an asymptotic pseudotrajectory of Φ.

(ii) X is uniformly continuous and every limit point[1] of $\{\theta^t(X)\}$ is in S_Φ.

(iii) The sequence $\{\theta^t(X)\}_{t\geq 0}$ is relatively compact in $C^0(\mathbb{R}^+, \mathcal{M})$.

Then (i) and (ii) are equivalent and imply (iii).

2.3.2 Flow-Tracking on Lie Groups

Benaïm and Hirsch examined in [47] the influence of a certain type of perturbation $(U_k)_{k\in\mathbb{N}_0} \subset \mathbb{R}^n$ on iteration processes of the form

$$x_{k+1} = x_k + \gamma_k\big(F(x_k) + U_k\big), \tag{2.38}$$

for a given vector field $F : \mathbb{R}^n \to \mathbb{R}^n$ on the Euclidean space. In this subsection we will extend the mentioned results to vector fields on Lie groups. In particular, we will study the iteration

$$x_{k+1} = \mu\left(\gamma_k\left[F(x_k) + U_k\right]\right) x_k, \quad x_0 \in G, \tag{2.39}$$

with the following setting:

- G is a n-dimensional, connected Lie group endowed with a right-invariant Riemannian metric. This is, there exists a scalar product on each tangent space T_gG such that

  $$\langle (T_{id}R_g)(v), (T_{id}R_g)(w)\rangle_{T_gG} = \langle v, w\rangle_{T_{id}G}, \quad \text{for all } v, w \in T_{id}G,$$

  where $R_g : G \to G$, $h \mapsto hg$ denotes the right-translation with $g \in G$. In particular, the Riemannian metric induces a distance function $d : G \times G \to \mathbb{R}$ such that (G, d) becomes a metric space. The topology on G induced by d coincides with the original manifold topology of G (cf. Jost [62] p.16). The corresponding Lie algebra is denoted by $\mathfrak{g}$.

- The exponential map $\exp : \mathfrak{g} \to G$ has the property that the function $x : \mathbb{R} \to G$, $x(t) = \exp(tv)$ is the unique solution of the initial value problem $\dot{x} = vx$, $x(0) = id$ in G for all $v \in \mathfrak{g}$. In particular, one should not mistake exp for the Riemannian exponential map.

[1] By a limit point of $\{\theta^t(X)\}$ we mean the limit in $C^0(\mathbb{R}^+, \mathcal{M})$ of a locally uniformly convergent sequence $\theta^{t_k}(X)$, $t_k \to \infty$.

- $F : G \to \mathfrak{g}$ is continuously differentiable and complete, i.e. each initial value problem of the form $\dot{x} = F(x)x$, $x(0) = x_0 \in G$ has a unique solution $x : \mathbb{R} \to G$,
- $(U_k)_{k\in\mathbb{N}_0} \subset \mathfrak{g}$ is a given sequence of perturbations.
- $(\gamma_k)_{k\in\mathbb{N}_0} \subset \mathbb{R}^+$ is a given real-valued sequence with

  $$\lim_{k\to\infty} \gamma_k = 0 \quad \text{and} \quad \sum_{k=0}^{\infty} \gamma_k = \infty.$$

  the so called gain sequence. We set

  $$\tau_0 = 0 \quad \text{and} \quad \tau_k = \sum_{n=0}^{k-1} \gamma_n \quad \text{for all } k \in \mathbb{N}.$$

- $\mu : \mathfrak{g} \to G$ is a local parameterization around the identity with

  $$\mu(0) = id \quad \text{and} \quad D\mu(0) = id.$$

Recall that the tangent bundle TG of G is trivial, i.e. $TG \cong \mathbb{R}^n \times G$. Thus, there exists a one to one correspondence between the vector fields $G \to TG$ on G and functions of the form $G \to \mathfrak{g}$. More precisely, if F maps G into $\mathfrak{g}$, then the mapping

$$g \mapsto F(g)g$$

defines a vector field on G and vice versa. Therefore, the iteration (2.39) might be seen as a straight forward generalization of (2.38). The example one should keep in mind is $F(g) = -(\mathfrak{grad} f(g))g^{-1}$, where $\mathfrak{grad} f(g) \in T_gG$ denotes the gradient of a smooth, real-valued function $f : G \to \mathbb{R}$ and $\gamma_k := \frac{1}{k+1}$. Then (2.39) becomes a noisy gradient-descent algorithm. Here the aim would be to find conditions for the perturbations which guarantee that the iteration x_k converges to a critical point of f.

In the case of an arbitrary function $F : G \to \mathfrak{g}$, one would like that the iterations $(x_k)_{k\in\mathbb{N}}$ generated by (2.39) "follow" the flow which is induced by F, no matter whether F has zeros or not. Therefore we will construct an interpolation curve $X : \mathbb{R}^+ \to G$ of the sequence $(x_k)_{k\in\mathbb{N}}$. We will show that X is an asymptotic pseudotrajectory of the semiflow induced by F under appropriate conditions on the perturbation $(U_k)_{k\in\mathbb{N}}$. It will turn out that the local parameterization μ has no influence on this result.
More precisely, we interpolate $(x_k)_{k\in\mathbb{N}}$ in two ways: One resulting interpolation curve $\overline{X} : \mathbb{R}^+ \to G$ is constant in each interval $[\tau_k \; \tau_{k+1})$ and the other $X : \mathbb{R}^+ \to G$ uses the exponential map. They are defined by

$$\begin{aligned} X(\tau_k + s) &:= \exp\left(\frac{s}{\gamma_k} \exp^{-1}(x_{k+1}x_k^{-1})\right) x_k, \quad \text{and} \\ \overline{X}(\tau_k + s) &:= x_k, \qquad \text{for all } k \in \mathbb{N}_0, \quad s \in [0, \gamma_k). \end{aligned} \tag{2.40}$$

Additionally, we interpolate the perturbations by a piecewise constant function $\overline{U} : \mathbb{R}^+ \to \mathfrak{g}$.

$$\overline{U}(\tau_k + s) := U_k, \qquad \text{for all } k \in \mathbb{N}_0, \quad s \in [0, \gamma_k). \tag{2.41}$$

One might see (2.40) as an extension of the case, where G coincides with a vector space. Then, X corresponds with a linear interpolation of the sequence $\{x_k\}_{k\in\mathbb{N}}$. We want to point out that X is not well-defined, in general, since the exponential map exp is, in general, neither injective, nor surjective. However we will see in the next proposition that under mild conditions, the term $\gamma_k [F(x_k) + U_k]$ is in the injectivity radius the exponential map. In this case, $\overline{X}$ becomes a well-defined function.

Proposition 2.10

Let $(x_k)_{k\in\mathbb{N}_0}$ denote the iteration defined in (2.39). Assume that the sets $\{F(x_k) \mid k \in \mathbb{N}_0\} \subset \mathfrak{g}$ and $\{U_k \mid k \in \mathbb{N}_0\} \subset \mathfrak{g}$ are bounded. Then the interpolation process X of the sequence $(x_k)_{k\in\mathbb{N}}$, given by (2.40), is well-defined if t is sufficiently large.

Proof

Let $r > 0$ denote the injectivity radius of the exponential map and let $B_r(0) \subset \mathfrak{g}$ denote the open ball in the Lie algebra $\mathfrak{g}$ around 0 with radius r. Since the local parameterization $\mu : \mathfrak{g} \to G$ satisfies $\mu(0) = id$ and $D\mu(0) = id$, there exists an open neighborhood $V \subset \mathfrak{g}$ of 0 such that $\mu|_V$ is diffeomorphic. Without loss of generality, we assume $\mu(V) \subset \exp(B_r(0))$. Since $(F(x_k) + U_k)_{k\in\mathbb{N}_0}$ is bounded, we have $\lim_{k\to\infty} \gamma_k[F(x_k) + U_k] = 0$. This implies that $\gamma_k[F(x_k) + U_k] \in V$, if k is large enough. Hence $\exp^{-1}(x_{k+1}x_k) = \exp^{-1}(\mu(\gamma_k[F(x_k) + U_k]))$ is well defined for k sufficiently large. □

The aim of this subsection is to specify the conditions on the perturbations U_k such that the interpolation curve X is an asymptotic pseudotrajectory of the semiflow induced by F. The following definition yields a helpful tool for this task:

Definition 2.3 (Moderate Trouble Functions)

We call a map $u : \mathbb{R}^+ \to \mathbb{R}^n$, $n \in \mathbb{N}$, a *moderate trouble function* (MTF) if u is integrable on each compact interval $I \subset \mathbb{R}^+$ and

$$\lim_{t\to\infty} \triangle(t, T) = 0$$

with

$$\triangle(t, T) := \sup_{0\leqslant h\leqslant T} \| \int_t^{t+h} u(s)ds\| \quad \text{for any } T > 0.$$

Recall that we are particularly interested in the case $u = \overline{U}$ defined in (2.41). In [47] it was already mentioned that the interpolated perturbations $\overline{U}(t)$ is a MTF if and only if

$$\lim_{n\to\infty} \sup \left\{ \| \sum_{i=n}^{k-1} \gamma_{i+1} U_{i+1} \| \;\middle|\; k = n, \dots, m(\tau_n + T) \right\} = 0 \tag{2.42}$$

is satisfied for all $T > 0$, where $m(t) := \sup\{k \in \mathbb{N} \mid \tau_k \leqslant t\}$. This is of particular interest, since in applications the perturbations are fixed and can not be influenced by the user. Hence, we have to choose the gain sequence carefully to guarantee that $\overline{U}(t)$ is a MTF.
The next two lemmata show some properties of the class of MTF functions.

Lemma 2.11
Let $u : \mathbb{R}^+ \to \mathbb{R}^n,\ n \in \mathbb{N}$ be a MTF and let $A : \mathbb{R}^+ \to \mathbb{L}(\mathbb{R}^n, \mathbb{R}^n)$ be bounded. Assume that for each $T > 0$ there exists $M > 0$ with

$$\mathcal{L}(A, [t, t+T]) < M \quad \text{for all } t \in \mathbb{R}^+, \tag{2.43}$$

where $\mathcal{L}(A, [a, b])$ denotes the total variation of the the function $A(t)$ on the interval $[a, b]$. Then the map $t \mapsto A(t)u(t)$ is also a MTF.

Proof
Let $T > 0$ be arbitrary but fixed. We have to show that

$$\lim_{t\to\infty} \Psi(t) = 0, \quad \text{with } \Psi(t) := \sup_{0\leqslant h\leqslant T} \| \int_t^{t+h} A(s)u(s)ds\|. \tag{2.44}$$

Since all norms in $\mathbb{R}^n$ are equivalent, we can assume that $\|.\|$ is the maximum-norm. Let $a_{i,j}(t)$ and $u_j(t)$ denote the entries of $A(t)$ and $u(t)$. Then we get

$$\| \int_t^{t+h} A(s)u(s)ds\| = \max_{1\leq i\leq n} \left| \sum_{j=1}^{n} \int_t^{t+h} a_{i,j}(s)u_j(s)ds \right| \leq \max_{1\leq i,j\leq n} \left| n \int_t^{t+h} a_{i,j}(s)u_j(s)ds \right|$$

Thus, the function

$$t \mapsto \sup_{0\leqslant h\leqslant T} \max \left\{ |n \int_t^{t+h} a_{i,j}(s)u_j(s)ds| \;\middle|\; i, j = 1, \dots, n \right\}$$

is an upper bound of the function Ψ in (2.44). Each coefficient $u_j(t)$ is a MTF and the inequality (2.43) is still satisfied if we replace $A(t)$ by an arbitrary entry $a_{i,j}(t)$. Thus it is enough to prove that the lemma holds for $n = 1$.
Now let $n = 1$. With the natural identification $\mathbb{L}(\mathbb{R}, \mathbb{R}) = \mathbb{R}$, we have $A : \mathbb{R}^+ \to \mathbb{R}$. Since A is of bounded variation on each interval $[t, t+T],\ t \in \mathbb{R}^+$, there are two monotonically increasing functions $B_t, C_t : [t, t+T] \to \mathbb{R}$ with

$$A(s) = B_t(s) - C_t(s), \quad \text{for all } s \in [t, t+T]$$

(cf. [63] p. 682). In particular, we can set $B_t(s) := \frac{1}{2}(\mathcal{L}(A, [t, s]) + A(s))$ and $C_t(s) := \frac{1}{2}(\mathcal{L}(A, [t, s]) - A(s))$. Therefore, both functions satisfy

$$\sup_{t\leqslant s\leqslant T} |B_t(s)|, \sup_{t\leqslant s\leqslant T} |C_t(s)| \leqslant \frac{1}{2}(M + \tilde{M}), \quad \tilde{M} := \sup_{t\in\mathbb{R}^+} |A(t)|.$$

We get

$$\sup_{0\leqslant h\leqslant T}|\int_t^{t+h} A(s)u(s)ds| \leqslant \sup_{0\leqslant h\leqslant T}|\int_t^{t+h} B_t(s)u(s)ds| + \sup_{0\leqslant h\leqslant T}|\int_t^{t+h} C_t(s)u(s)ds|. \quad (2.45)$$

According to the *Mean Value Theorem of Bonnet* (cf. [63] p.734) the following statement is true: Let $I = [\alpha, \beta] \subset \mathbb{R}$ be compact, let $f : I \to \mathbb{R}$ be monotone and let $g : I \to \mathbb{R}$ be integrable on I. Then, there exists $\xi \in I$ such that

$$\int\limits_a^b f(s)g(s)ds = f(a)\int\limits_a^\xi g(s)ds + f(b)\int\limits_\xi^b g(s)ds.$$

Due to the monotony of B_t and the integrability of u, we can apply this theorem. Hence, there is a point $\xi \in [t, t+h]$ with

$$\int_t^{t+h} B_t(s)u(s)ds = B_t(t)\int_t^\xi u(s)ds + B_t(t+h)\int_\xi^{t+h} u(s)ds, \quad h \in [0,T]$$

which implies

$$\begin{aligned}|\int_t^{t+h} B_t(s)u(s)ds| &\leqslant \frac{1}{2}(M+\tilde{M})\left[|\int_t^\xi u(s)ds| + |\int_t^{t+h} u(s)ds| + |\int_t^\xi u(s)ds|\right] \\ &\leqslant \frac{3}{2}(M+\tilde{M})\sup_{0\leqslant h\leqslant T}|\int_t^{t+h} u(s)ds|.\end{aligned}$$

In the same fashion we get

$$|\int_t^{t+h} C_t(s)u(s)ds| \leqslant \frac{3}{2}(M+\tilde{M})\sup_{0\leqslant h\leqslant T}|\int_t^{t+h} u(s)ds|.$$

Using (2.45), we conclude

$$|\int_t^{t+h} A(s)u(s)ds| \leqslant 3(M+\tilde{M})\sup_{t\leqslant h\leqslant t+T}|\int_t^{t+h} u(s)ds|.$$

Hence, the MTF-property of $u(t)$ implies the MTF-property of $A(t)u(t)$. □

Lemma 2.12

Let G be a finite-dimensional Lie group endowed with a Riemannian metric and let $\mathfrak{g}$ be its Lie algebra. Assume that the function $u : \mathbb{R}^+ \to \mathfrak{g}$ is bounded and a MTF. Furthermore, let the function $y_t : \mathbb{R}^+ \to G,\ t \in \mathbb{R}^+$ denotes the unique solution of the initial value problems

$$\frac{d}{ds}y_t(s) = u(t+s)y_t(s), \quad y_t(0) = id. \quad (2.46)$$

Then the family $\{y_t\}_{t\in\mathbb{R}^+}$ converges locally uniformly to the function $y_\star \equiv id$. Thus, for each $T > 0$ we get

$$\lim_{t\to\infty}\sup_{0\leqslant h\leqslant T} d(y_t(h), id) = 0, \quad (2.47)$$

where $d : G \times G \to \mathbb{R}$ denotes the distance function induced by the Riemannian metric.

Proof
Since the exponential mal exp is a smooth function and $D\exp(0) = id$, there exists $R > 0$ such that $D\exp(v) : \mathfrak{g} \to T_vG$ is bijective. We define the linear operator $L(v) : \mathfrak{g} \to \mathfrak{g}$ for each $v \in \mathfrak{g}$ with $\|v\| \leq R$ by

$$L(v)w := D\exp(v)^{-1}\left(w \exp(v)\right).$$

Following Hilgert and Neeb (cf. [60] p.39) we get the series expansion

$$L(v)^{-1} = \sum_{k=1}^{\infty} \frac{1}{k!}(-ad(v))^{k-1},$$

where the sum is locally uniformly convergent in the (finite-dimensional) space of all endomorphisms of $\mathfrak{g}$. Hence, there is a real-valued sequence $(a_k)_{k\in\mathbb{N}}$ with

$$L(v) = \sum_{k=1}^{\infty} a_k(-ad(v))^{k-1}.$$

We obtain $a_1 = 1$ and therefore

$$L(v) = id + \left(\sum_{k=2}^{\infty} a_k(-ad(v))^{k-2}\right) ad(v).$$

Without loss of generality, let $R > 0$ be smaller than the convergence radius of the power series $\sum_{k=1}^{\infty} a_k z^k$. Then we get a bound $M > 0$ such that $\|\sum_{k=2}^{\infty} a_k(ad(v))^{k-2}\| < M$ for all $v \in \mathfrak{g}$ with $\|ad(v)\| < R$. As a result we get

$$\|L(v)w\| \leqslant \|w\| + M\|w\| \cdot \|v\|, \quad \text{for all } v \in \mathfrak{g} \text{ with } \|ad(v)\| < R \text{ and } w \in \mathfrak{g}.$$

Now, let $\alpha_t : I_t^{\max} \subset \mathbb{R} \to \mathfrak{g}$, $t \geq 0$ be the maximal solution of the initial value problem

$$\dot{\alpha}_t(s) = \left[D\exp(\alpha_t(s))\right]^{-1}\left(u(t+s)\exp(\alpha_t(s))\right), \quad \alpha_t(0) = 0. \tag{2.48}$$

In particular, the right hand-side side of this differential equation is well defined for $\|\alpha_t(s)\| < R$. For each $T > 0$ we will show $[0, T] \subset I_t^{\max}$ if t is sufficiently large and

$$\lim_{t\to\infty} \sup_{0\leqslant h\leqslant T} \|\alpha_t(s)\| = 0. \tag{2.49}$$

With these properties at hand, the function $y_t(s) := \exp(\alpha_t(s))$ satisfies the initial value problem (2.46) and, due to the continuity of the exponential map, the convergence of α_t yields (2.47).
Since the differential equation (2.48) has the form $\dot{\alpha}_t(s) = L(\alpha_t(s))u(t+s)$, we estimate

$$\| \int_0^h \dot{\alpha}_t(s)ds\| \leqslant \| \int_0^h u(s+t)ds\| + M \int_0^h \|u(s+t)\| \cdot \|\alpha_t(s)\|ds$$

for all $h > 0$ such that $\|ad(\alpha_t(s))\| < R$ for all $0 \leqslant s \leqslant h$. Let $B > 0$ denote the bound of the function u. Then we obtain

$$\|\alpha_t(h)\| \leqslant \| \int_0^h u(s+t)ds\| + MB \int_0^h \|\alpha_t(s)\|ds.$$

Using the *Gronwall Lemma*, this implicit inequality yields

$$\begin{aligned} \|\alpha_t(h)\| &\leqslant \| \int_0^h u(s+t)ds\| + MB \int_0^h \| \int_0^s u(\tau+t)d\tau\| e^{MB(h-s)}ds \\ &\leqslant \sup_{0\leqslant\tau\leqslant h} \| \int_0^\tau u(s+t)ds\| \left(1 + MB \int_0^h e^{MB(h-s)}ds\right). \end{aligned} \tag{2.50}$$

Now let $T > 0$ be arbitrary and fixed. Since u is a MTF, the inequality (2.50) yields $\|ad(\alpha_t(s))\| < R$ for all $s \in I_t^{\max} \cap [0,T]$ if t is sufficiently large. Moreover, since $\alpha_t(s)$ is assumed to be the maximal solution of (2.48), we can apply the *Global Existence and Uniqueness Theorem* (cf. e.g. [64] p.84). Thus, (2.50) and the MTF-property yield $[0,T] \subset I_t^{\max}$ if t is sufficiently large. Finally, (2.50) and the MTF-property of u yield (2.49). □

Theorem 2.13
Let G be a finite-dimensional Lie group endowed with a right-invariant Riemannian metric and let $\mathfrak{g}$ be its Lie algebra. Assume that $F : G \to \mathfrak{g}$ is continuously differentiable and complete. Consider the iteration (2.39) with the piecewise constant interpolation $\overline{U}$, defined by (2.41). Assume that

A1 $\overline{U}(t)$ *is a MTF,*

A2 $\overline{U}(t)$ *is bounded,*

A3 $(x_k)_{k\in\mathbb{N}_0}$ *is bounded and*

A4 F *is uniformly continuous on a closed neighborhood of* $\{x_k \mid k \in \mathbb{N}_0\}$*, or*

A4' G is geodesically complete[2].

Then the interpolation process $X(t)$ of the sequence $(x_k)_{k\in\mathbb{N}}$ given by (2.40), is well-defined if t is sufficiently large. Moreover, $X(t)$ is an asymptotic pseudotrajectory of the flow Φ induced by F.

Proof
The *Theorem of Hopf-Rinow* (see e.g. [62] p.29) states that A4' is equivalent to the fact that G is complete as a metric space. Therefore, assumption A4' implies that there exists a compact set including the sequence $(x_k)_{k\in\mathbb{N}_0}$. Since F is continuous, the set $\{F(x_k) \mid k \in \mathbb{N}_0\}$ is bounded. If assumption A4 is satisfied, F is uniformly continuous on a neighborhood W of $\{x_k \mid k \in \mathbb{N}_0\}$. Since $(x_k)_{k\in\mathbb{N}_0}$ is bounded, we can without loss of generality assume that

[2] A Riemannian mainfold is *geodesically complete* if any geodesic $c(t)$ with $c(0) = p$ is defined for all $t \in \mathbb{R}$.

W is bounded. Thus the set $\{F(x_k) \mid k \in \mathbb{N}_0\}$ is again bounded and the well-definedness of $X(t)$ follows directly from Proposition 2.10.
According to (2.40), the interpolation $X(t)$ forms piecewise a one-parameter group. Therefore, it satisfies

$$\begin{aligned}\dot{X}(\tau_k + s) &= \frac{1}{\gamma_k} \exp^{-1}\left(x_{k+1}x_k^{-1}\right) X(\tau_k + s)\\ &= \frac{1}{\gamma_k} \exp^{-1}\left(\mu\left(\gamma_k\left[F(x_k) + U_k\right]\right)\right) X(\tau_k + s), \quad \text{for all } k \in \mathbb{N}_0,\ s \in [0, \gamma_k).\end{aligned}$$

Applying the *Taylor Theorem* to the function $t \mapsto (\exp^{-1} \circ \mu)(tv_k)$, $v_k := \gamma_k[F(x_k) + U_k]$ yields

$$\dot{X}(\tau_k + s) = \frac{1}{\gamma_k}\left(v_k + \frac{1}{2}D^2(\exp^{-1} \circ \mu)(\theta_k v_k)(v_k, v_k)\right) X(\tau_k + s)$$

for an appropriate $\theta_k \in [0, 1]$, $k \in \mathbb{N}_0$ and $s \in [0, \gamma_k)$. We get

$$\dot{X}(\tau_k + s) = \frac{1}{\gamma_k}\left(\gamma_k\left[F(x_k) + U_k\right] + \gamma_k V_k\right) X(\tau_k + s), \quad \text{for all } k \in \mathbb{N}_0,\ s \in [0, \gamma_k),$$

where

$$\|V_k\| \leqslant \frac{1}{2}\gamma_k \|F(x_k) + U_k\|^2 \max\left\{\|D^2(\exp^{-1} \circ \mu)(v)\| \;\middle|\; \|v\| \leqslant \|F(x_k) + U_k\|\right\}. \tag{2.51}$$

Let $\overline{V}(t)$ denote the constant interpolation of the elements V_k, $k \in \mathbb{N}$, like it is done in (2.41) for the elements U_k. We obtain that $X(t)$ satisfies the differential equation

$$\dot{X}(t) = \left[F(\overline{X}(t)) + \overline{U}(t) + \overline{V}(t)\right] X(t) \tag{2.52}$$

for almost all $t \in \mathbb{R}^+$. According to (2.51) we obtain $\lim_{k\to\infty} V_k = 0$. Thus, $\overline{V}(t)$ is also a MTF.
First we consider the function $y(t) := \exp^{-1}(X(t)\overline{X}(t)^{-1})$. Using the differential equation (2.52) and exploiting the fact that $\overline{X}(t)$, $\overline{U}(t)$ and $\overline{V}(t)$ are constant on each interval $[\tau_k, \tau_{k+1})$, we get that $X(t)$ is part of a one-parameter group on each interval $[\tau_k, \tau_{k+1})$. Thus,

$$\begin{aligned}y(t) &= \exp^{-1}\left(\exp\left((t - \tau_k)\left[F(\overline{X}(t)) + \overline{U}(t) + \overline{V}(t)\right]\right) X(\tau_k)\overline{X}(t)^{-1}\right)\\ &= (t - \tau_k)(F(\overline{X}(t)) + \overline{U}(t) + \overline{V}(t))\\ &= \int_{\tau_k}^{t} F(\overline{X}(s))ds + \int_{\tau_k}^{t}\left(\overline{U}(s) + \overline{V}(s)\right) ds,\end{aligned}$$

for all $t \in [\tau_k, \tau_{k+1})$, $k \in \mathbb{N}$. In both cases, A4 or A4', F is uniformly continuous and closed neighboorhood of $(x_k)_{k\in\mathbb{N}}$. Thus, there exists $K > 0$ with $\|F(x_k)\| < K$ for all $k \in \mathbb{N}_0$. Let $\tilde{T} := \sup_k \gamma_k$, then

$$\|y(t)\| \leqslant \gamma_{k+1}K + \underbrace{\sup_{0\leqslant h\leqslant\tilde{T}} \|\int_{\tau_k}^{\tau_k+h}\left(\overline{U}(s) + \overline{V}(s)\right) ds\|}_{=:\triangle(\tau_k,\tilde{T})} \quad \text{for all } t \in [\tau_k, \tau_{k+1}),\ k \in \mathbb{N}.$$

In particular, we get $\lim_{t\to\infty} y(t) = 0$. Let $\delta > 0$ be arbitrary. Due to the continuity of $\exp : \mathfrak{g} \to G$, there exists $\tilde{\delta} > 0$ with $d(\exp(v), 0) < \delta$ for all $\|v\| < \tilde{\delta}$. Let $d : G \times G \to \mathbb{R}$ denote the distance function induced by the Riemannian metric. If $k \in \mathbb{N}$ is large enough, we conclude

$$d(X(t), x_k) = d(X(t)\overline{X}(t)^{-1}, id) = d(\exp(y(t)), id) \leqslant \delta \quad \text{for all } t \in [\tau_k, \tau_{k+1}). \tag{2.53}$$

Hence, X is also bounded.

By assumption A4', there exists a compact set $\mathfrak{K} \subset G$ including the trajectories $\{X(t) \mid t \in \mathbb{R}^+\}$ and $\left\{\overline{X}(t) \mid t \in \mathbb{R}^+\right\}$. Since F is continuous, it is also uniformly continuous on $\mathfrak{K}$. If assumption A4 is satisfied, F is uniformly continuous on a closed neighborhood $W \subset G$ of $\{x_k \mid k \in \mathbb{N}_0\}$. Since (2.53) is satisfied, there exists $t_0 \geq 0$ such that W includes the trajectories $\{X(t) \mid t \geq t_0\}$ and $\left\{\overline{X}(t) \mid t \geq t_0\right\}$. Without loss of generality, we assume $t_0 = 0$ and set $\mathfrak{K} = W$ in this case.

Let $T > 0$ be fixed and let $\varepsilon > 0$ be arbitrarily chosen. Then, there is a $\delta > 0$ with $\|F(a) - F(b)\| < \varepsilon$ for all $d(a, b) < \delta$, and $a, b \in \mathfrak{K}$. Using (2.53), there is also $\tilde{t} > 0$ with $d(X(t), \overline{X}(t)) < \delta$ for all $t > \tilde{t}$. Hence,

$$\lim_{t\to\infty} \sup_{0 \leqslant h \leqslant T} \| \int_t^{t+h} F(X(s)) - F(\overline{X}(s)) ds \| \leqslant \varepsilon T$$

Since $\varepsilon > 0$ was arbitrarily chosen, we get that $F(X(t)) - F(\overline{X}(t))$ is a MTF.

In order to show the uniform continuity of $X(t)$, we use the right-invariance of the metric d and obtain

$$\begin{aligned} d(X(t+h), X(t)) &\leqslant \mathcal{L}(X, [t, t+h]) := \int_t^{t+h} \|\dot{X}(s)\| ds \\ &\leqslant \int_t^{t+h} \left(\|F(\overline{X}(s))\| + \|\overline{U}(s)\| + \|\overline{V}(s)\|\right) ds \end{aligned}$$

Recall that the terms $F(\overline{X}(t))$, $\overline{U}(t)$ and $\overline{V}(t)$ are bounded. So, there exists a $M > 0$ with

$$d(X(t+h), X(t)) \leqslant Mh.$$

This implies the uniform continuity of $X(t)$.

Now we define the functions $x_t(s) := X(t+s)$ for each $t > 0$. Let $z_t : \mathbb{R}^+ \to G$ be the unique solution of the initial value problem

$$\dot{z}_t(s) = F(x_t(s))z_t(s), \quad z_t(0) = x_t(0) \tag{2.54}$$

and let $w_t : \mathbb{R}^+ \to G$ be defined by $w_t(s) = x_t(s)^{-1}z_t(s)$. Hence, w_t satisfies the initial value problem $w_t(0) = id$ and

$$\begin{aligned} \dot{w}_t(s) &= -x_t(s)^{-1}\dot{x}_t(s)x_t(s)^{-1}z_t(s) + x_t(s)^{-1}\dot{z}_t(s) \\ &= -x_t(s)^{-1}\left[F(\overline{X}(t+s)) + \overline{U}(t+s) + \overline{V}(t+s)\right] z_t(s) + x_t(s)^{-1}F(X(t+s))z_t(s) \\ &= x_t(s)^{-1}\left[F(X(t+s)) - F(\overline{X}(t+s)) - \overline{U}(t+s) - \overline{V}(t+s)\right] x_t(s) \cdot w_t(s) \end{aligned}$$

$$= Ad_{X(t+s)}\left[F(X(t+s)) - F(\overline{X}(t+s)) - \overline{U}(t+s) - \overline{V}(t+s)\right] w_t(s)$$

for almost all $s \in \mathbb{R}^+$. We have already shown that $t \mapsto F(X(t)) - F(\overline{X}(t))$ and $t \mapsto \overline{V}(t)$ are MTFs. Since $X(t)$ is bounded and uniformly continuous, the map $A : \mathbb{R} \to \mathbb{L}(\mathfrak{g}, \mathfrak{g})$, $A(t) := Ad_{X(t)}$ satisfies the conditions of Lemma 2.11. Therefore, the map

$$t \mapsto Ad_{X(t)}\left[F(X(t)) - F(\overline{X}(t)) - \overline{U}(t) - \overline{V}(t)\right]$$

is also a MTF and we can apply Lemma 2.12 to w_t, which yields

$$\lim_{t\to\infty} w_t = \lim_{t\to\infty} x_t^{-1} z_t = id. \tag{2.55}$$

Finally, let $x_\star$ denote a limit point of $\{x_t\}_{t\in\mathbb{R}^+}$. Using (2.55) and (2.54), we obtain

$$\dot{x}_\star(s) = F(x_\star(s))x_\star(s).$$

Hence, $x_\star$ is an element of S_Φ, where Φ is the flow induced by F. Theorem 2.9 shows that X is an asymptotic pseudotrajectory of Φ. □

Theorem 2.13 might be seen as an extension of the case where G is equal to a finite-dimensional, real vector space. This case was already examined by Hirsch and Benaïm in [47]. They proved that for $G = \mathbb{R}^n$ the conditions A1 and A3 are sufficient to conclude that the interpolation process X is an asymptotic pseudotrajectory of the flow Φ induced by F. Hence, one might assume that condition A2 is redundant for Theorem 2.13. We want to end this subsection with an example where one needs condition A2 to obtain the same result in the case where G is a Lie group.

Example:
We set $G = SO(3)$, $\mu = \exp$ and $F \equiv 0$. Moreover, let

$$E_1 := \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad E_2 := \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{pmatrix}, \quad E_3 := \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix}$$

define a basis of the Lie algebra $so(3)$. We consider the perturbations $(U_k)_{k\in\mathbb{N}} \subset so(3)$ and the gain-sequence $(\gamma_k)_{k\in\mathbb{N}} \subset \mathbb{R}^+$ defined by

$$U_k := \begin{cases} -\sqrt{k+1}E_2 & \text{for } k = 0 \bmod 4 \\ -\sqrt{k}E_1 & \text{for } k = 1 \bmod 4 \\ \sqrt{k-1}E_2 & \text{for } k = 2 \bmod 4 \\ \sqrt{k-2}E_1 & \text{for } k = 3 \bmod 4 \end{cases} \quad \text{and} \quad \gamma_k := \begin{cases} \frac{1}{k+1} & \text{for } k = 0 \bmod 4 \\ \frac{1}{k} & \text{for } k = 1 \bmod 4 \\ \frac{1}{k-1} & \text{for } k = 2 \bmod 4 \\ \frac{1}{k-2} & \text{for } k = 3 \bmod 4. \end{cases}$$

Using (2.41), we get that the constant interpolation $\overline{U} : \mathbb{R}^+ \to so(3)$ of $(U_k)_{k\in\mathbb{N}}$ is a MTF and unbounded. We examine the iteration

$$x_{k+1} = \exp(\gamma_k U_k)x_k, \quad x_0 = id \tag{2.56}$$

and show that its interpolation $X : \mathbb{R}^+ \to SO(3)$ via (2.52) is not an asymptotic pseudo-trajectory of the flow induced by F. For this purpose, we consider the sub-sequence $(y_k)_{k\in\mathbb{N}}$

defined by $y_k := x_{4k}$ and the interpolation curve $Y : \mathbb{R}^+ \to SO(3)$ of $(y_k)_{k\in\mathbb{N}_0}$ defined by (2.52). A Taylor-expansion of the exponential-map yields

$$\exp\left(\frac{E_1}{\sqrt{k+1}}\right)\exp\left(\frac{E_2}{\sqrt{k+1}}\right)\exp\left(\frac{-E_1}{\sqrt{k+1}}\right)\exp\left(\frac{-E_2}{\sqrt{k+1}}\right) = I_3 + \frac{E_1E_2 - E_2E_1}{k+1} + W_k.$$

with $\lim_{k\to\infty}(k+1)W_k = 0$ (cf.[60] p.23). Therefore, using $E_3 = [E_1, E_2]$, the subsequence y_k satisfies

$$y_{k+1} = \exp\left(\frac{1}{k+1}E_3 + \tilde{W}_k\right) y_k, \quad y_0 = id,$$

with a suitable $\tilde{W}_k$ having the property $\lim_{k\to\infty}(k+1)\tilde{W}_k = 0$. Hence, y_k is an iteration of the form (2.39) with gain sequence $\tilde{\gamma}_k = \frac{1}{k+1}$, vector field $\tilde{F} \equiv E_3$ and perturbations $\tilde{U}_k = \tilde{W}_k$. Moreover, the constant interpolation of $\tilde{U}_k$ is a bounded MTF and we can apply Theorem 2.13 to the interpolation curve Y. This yields that Y is an asymptotic pseudotrajectory of the flow induced by $\tilde{F}$. Thus, neither the interpolation curve Y of $(y_k)_{k\in\mathbb{N}}$ nor the interpolation curve X of $(x_k)_{k\in\mathbb{N}_0}$ is an asymptotic pseudotrajectory of the flow induced by the initial vector field $F \equiv 0$.
To sum up, the interpolation of the iteration (2.56) is not an asymptotic pseudotrajectory of the flow induced by the vector field $F \equiv 0$ although the constant interpolation of the perturbations $(U_k)_{k\in\mathbb{N}}$ is a MTF. Therefore, the condition A2 in Theorem 2.13 is in general necessary.

2.3.3 Tracking Methods on Manifolds

In the previous section we extended the result of flow tracking in Euclidean spaces of Hirsch and Benaïm to finite-dimensional Lie groups. At first sight an additional generalization to differentiable manifolds $\mathcal{M}$ seems to be the natural next step. However two problems appear in a further extension: First, if the tangent bundle of the manifold is not trivial, we have to use parallel-transport to extend the MTF-property (2.42), since U_k is a perturbation in the tangent space $T_{x_k}\mathcal{M}$. This would destroy the advantage of the MTF-property that (2.42) is independent of the sequence $(x_k)_{k\in\mathbb{N}}$.
Second, Theorem 2.13 makes a statement about the distance between the interpolation $h \mapsto X(t+h)$ and the integral curve of the vector field F, which starts in $X(t)$. In our proof the group operation was an essential tool to approximate this distance independently of the current point $X(t)$.
One way to extend Theorem 2.13 to differentiable manifolds in spite of the mentioned difficulties is to strengthen condition A3. Assuming the sequence $(x_k)_{k\in\mathbb{N}}$ converges, we will show in the next corollary that the vector field F is zero in the limit. Since we can locally flatten a differentiable manifold $\mathcal{M}$, no additional structures on $\mathcal{M}$ are needed for this result.

Corollary 2.14
Let $\mathcal{M}$ be a differentiable, n-dimensional manifold, $\mathcal{U} \subset \mathcal{M}$ an open set and $\mu : \mathcal{U} \times \mathbb{R}^n \to \mathcal{M}$ a smooth local parameterization of $\mathcal{U}$. Moreover, let $(\gamma_k)_{k\in\mathbb{N}} \subset \mathbb{R}^+$ be a given gain-sequence and $(U_k)_{k\in\mathbb{N}} \subset \mathbb{R}^n$ a given sequence of perturbations such that its constant interpolation

function (2.41) is a bounded MTF. For a given continuous vector field $F : \mathcal{M} \to \mathbb{R}^n$ and a given starting point $x_0 \in \mathcal{U}$ we consider the iteration

$$x_{k+1} := \mu\left(x_k, \gamma_k \left[F(x_k) + U_k\right]\right). \tag{2.57}$$

Assuming $(x_k)_{k\in\mathbb{N}}$ has a limit $x_\star \in \mathcal{U}$, then the vector field F vanishes in $x_\star$.

Proof

We will make use of the abbreviations $\mu_x(v) := \mu(x,v)$, $\nu_k := \mu_{x_\star}^{-1} \circ \mu_{x_k}$ and $y_k = \mu_{x_\star}^{-1}(x_k)$. Applying *Taylor's Theorem* to the iteration (2.57) yields

$$\begin{aligned} y_{k+1} &= \nu_k\left(\gamma_k\left[F(x_k)+U_k\right]\right) \\ &= y_k + D\nu_k(0)\left(\gamma_k\left[F(x_k)+U_k\right]\right) + \frac{1}{2}D^2\nu_k\left(\theta_k\gamma_k\left[F(x_k)+U_k\right]\right)\left(\gamma_k\left[F(x_k)+U_k\right]\right)^2 \end{aligned}$$

for suitable $(\theta_k)_{k\in\mathbb{N}} \subset [0,1]$. We have

$$y_{k+1} = y_k + \gamma_k\left[\tilde{F}(y_k) + \tilde{U}_k\right] \tag{2.58}$$

with

$$\tilde{F}(y) := D(\mu_{x_\star}^{-1} \circ \mu_{\mu_{x_\star}(y)})(0)F(\mu_{x_\star}(y))$$

and

$$\tilde{U}_k := D\nu_k(0)U_k + \frac{1}{2}D^2\nu_k\left(\theta_k\gamma_k\left[F(x_k)+U_k\right]\right)\left(\gamma_k\left[F(x_k)+U_k\right], \left[F(x_k)+U_k\right]\right). \tag{2.59}$$

Let $\overline{Y}(t)$ denote the constant interpolation[3] of $(y_k)_{k\in\mathbb{N}}$. Since $\tilde{F}(y_k) + \tilde{U}_k$ is bounded, the iteration (2.58) yields that each $T > 0$ has a bound $\tilde{M} < 0$ with

$$\mathcal{L}(\overline{Y}, [t, t+T]) < \tilde{M}, \quad \text{for all } t \in \mathbb{R}^+.$$

Therefore, the map $A : \mathbb{R}^+ \to \mathbb{L}(\mathbb{R}^n, \mathbb{R}^n)$, $A(t) := D(\mu_{x_\star}^{-1} \circ \mu_{\mu_{x_\star}(\overline{Y}(t))})(0)$ satisfies (2.43) and we can apply Lemma 2.11. It yields that the constant interpolation of $D\nu_k(0)U_k$ is also a MTF. Since the second summand in (2.59) converges to zero for $k \to \infty$, we conclude that the constant interpolation of $\tilde{U}_k$ is a MTF. Hence, we can apply Theorem 2.13 to the iteration (2.58), which yields that the linear interpolation $Y(t)$ of $(y_k)_{k\in\mathbb{N}}$ is an asymptotic pseudotrajectory of $\tilde{F}$.

Due to $x_k \to x_\star$, we have also $\lim_{t\to\infty} Y(t) = 0$, which now implies $\tilde{F}(0) = 0$. Since $\tilde{F}(0) = \lim_{k\to\infty} D\nu_k(0)F(x_k)$ and μ is smooth, we conclude $F(x_\star) = 0$. □

Unfortunately, there is no guarantee that the sequence in (2.57) will converge. Even in the case of optimization on vector spaces, convergence-statements are only known for very restrictive properties to the vector field F (see e.g. [56] Section 4.3.2). We will deal with this particular difficulty in Section 6.2. The root-finding scheme for vector fields on manifolds is summarized in Table 2.3. In the cases, when $\mathcal{M}$ forms a Lie group G, this scheme can be simplified: We can set $\mathcal{U} = \mathcal{M} = G$ and $\mu(x,v) := \nu(v)x$, where we identify the Lie algebra $\mathfrak{g}$ with $\mathbb{R}^n$ and $\nu : \mathfrak{g} \to G$ is a local parameterization of the identity. Here, the third stopping criterion is trivial.

[3] During this proof, we will use the terms constant and linear interpolation with respect to the gain-sequence $(\gamma_k)_{k\in\mathbb{N}}$, as introduced on page 36.

Table 2.3: Root-Finding Scheme for Vector Fields on Manifolds

Given: n-dimensional manifold $\mathcal{M}$, $\mathcal{U} \subset \mathcal{M}$ open, $\mu : \mathcal{U} \times \mathbb{R}^n \to \mathcal{M}$ smooth local parameterization, $F : \mathcal{M} \to \mathbb{R}^n$ continuous.

Start:
Make an initial point $x_0 \in \mathcal{M}$. Set $k = 0$.

Repeat:

$$\begin{aligned} x &\longleftarrow \mu\left(x, \gamma_k \left[F(x) + U_k\right]\right) \\ k &\longleftarrow k + 1 \end{aligned}$$

Stop: if one of the following criteria is satisfied

- $(x_k)_{k \in \mathbb{N}}$ satisfies a convergence criterion,
- k reached a user-defined number or
- $x \notin \mathcal{U}$.

2.3.4 The Robbins-Monro Algorithm

In the previous section, we pointed out the advantage of the MTF-property for perturbations $(U_k)_{k \in \mathbb{N}_0}$ in an iteration of the form (2.57). However, in many applications the nature of perturbations is not deterministic but stochastic. In this case, we can interpret $(U_k)_{k \in \mathbb{N}_0}$ as a sequence of random vectors (noise), which disturb the detection of the desired update $F(x_k)$ in (2.57). Hence, the question arises about the probability that a realization of the sequence of random vectors $(U_k)_{k \in \mathbb{N}_0}$ satisfy the MTF-property. We will concern this question in this subsection. The following definition can be found in [65] p.14.

Definition 2.4 (Martingale difference Noise)
Let $(\Omega, \mathcal{F}, P)$ be a probability space and $(\mathcal{F}_k)_{k \in \mathbb{N}}$ a nondecreasing sequence of sub-σ-algebras of $\mathcal{F}$. The stochastic process $(x_k)_{k \in \mathbb{N}}$ given by (2.39) satisfies the *Robbins-Monro* or *Martingale difference Noise* condition if

(i) $(\gamma_k)_{k \in \mathbb{N}}$ is a deterministic sequence

(ii) each $U_k \in \mathfrak{g}$ is measurable with respect to $\mathcal{F}_k$

(iii) $E(U_{k+1} | \mathcal{F}_k) = 0$.

One example for the sequence of sub-σ-algebras one should keep in mind is, when $\mathcal{F}_k$ is identical the σ-algebra generated by $U_1, \ldots, U_k$. Then, (ii) is always satisfied and $(\mathcal{F}_k)_{k \in \mathbb{N}}$ is nondecreasing. The proof of the following theorem can be found in [65] p.15. It connects the Martingale difference Noise condition with the MTF-property.

Proposition 2.15
Let $(x_k)_{k\in\mathbb{N}}$ given by (2.39) satisfy the Marginale difference Noise condition. Suppose that for some $q \geqslant 2$

$$\sup_{k\in\mathbb{N}} E(\|U_{k+1}\|^q) < \infty \quad \textit{and} \quad \sum_{k=1}^{\infty} \gamma_k^{1+q/2} < \infty. \tag{2.60}$$

Then the constant interpolation $\overline{U} : \mathbb{R}^+ \to \mathbb{R}^n$ of the perturbations $(U_k)_{k\in\mathbb{N}}$ given by (2.41) is a MTF with probability 1.

Now, we have all necessaries at hand to formulate the gradient-based stochastic optimization algorithm on manifolds: Let $\mathcal{M}$ denote a n-dimensional manifold, $\mathcal{U} \subset \mathcal{M}$ an open subset and $\mu : \mathcal{U} \times \mathbb{R}^n \to \mathcal{M}$ a smooth local parameterization of $\mathcal{U}$. For a given, real-valued function $f \in C^1(\mathcal{M}, \mathbb{R})$ and a stochastic process $(U_k)_{k\in\mathbb{N}} \subset \mathbb{R}^n$ we examine the iteration

$$x_{k+1} = \mu_{x_k}(\gamma_k \left[-\nabla(f \circ \mu_{x_k})(0) + U_k\right]), \quad x_0 \in \mathcal{U}. \tag{2.61}$$

Corollary 2.16
Let $(x_k)_{k\in\mathbb{N}}$ given by (2.61) satisfy the Martingale difference Noise condition with respect to the probability space $(\Omega, \mathcal{F}, P)$. Moreover, let the sample space $\Omega \subset \mathbb{R}^n$ be bounded and let the sequence $(\gamma_k)_{k\in\mathbb{N}} \subset \mathbb{R}^+$ satisfy

$$\sum_{k=1}^{\infty} \gamma_k = \infty \quad \textit{and} \quad \sum_{k=1}^{\infty} \gamma_k^{1+q/2} < \infty \tag{2.62}$$

for some $q \geqslant 2$. Suppose, $(x_k)_{k\in\mathbb{N}}$ converges to a point $x_\star \in \mathcal{U}$. Then $x_\star$ is a critical point of f with probability 1.

Proof
Since $\Omega \subset \mathbb{R}^n$ is bounded, each realization $(U_k)_{k\in\mathbb{N}} \subset \Omega$ is also bounded. Proposition 2.15 yields that the constant interpolation $\overline{U}(t)$ of the perturbations $(U_k)_{k\in\mathbb{N}}$ given by (2.41) is a MTF with probability 1. Therefore, with probability 1, we can apply Corollary 2.14 to the iteration (2.61) and the vector field $F(x) := \nabla(f \circ \mu_x)(0)$. It yields that the vector field F vanishes at the limiting point $x_\star$ with probability 1. Since $D(f \circ \mu_x)(0) = Df(x) \circ D\mu_x(0)$ and $D\mu_{x^\star}(0)$ is non-degenerate, the corollary is proven. □

Chapter 3

A Geometric Model for Image Representation

For two given images, the so-called reference R and the template T, the task of image registration can be formulated in the following way: Find a transformation φ, which best matches R to T. We will see in this chapter that each gray-scale image can be represented by a real-valued function. In this case the registration task can be written as an optimization problem of the form

$$\min_{\varphi \in G} \mathfrak{D}\left(R \circ \varphi, T\right), \quad G \subset \mathrm{Diff}(\mathbb{R}^n), \tag{3.1}$$

where $R, T \in C(\mathbb{R}^n, \mathbb{R})$ and $\mathrm{Diff}(\mathbb{R}^n)$ denotes the set of all diffeomorphisms in $\mathbb{R}^n$. The map $\mathfrak{D} : C(\mathbb{R}^n, \mathbb{R}) \times C(\mathbb{R}^n, \mathbb{R}) \to \mathbb{R}$ is user-defined in order to compare the two functions $R \circ \varphi$ and T.

In this chapter, we describe the necessary steps from the abstract image registration task towards an optimization problem (3.1) on a Lie group G. Since there is no general agreement in the literature how to tackle this problem, we try to summarize the main approaches. In the first section, we show the connection between the pose-estimation problem and the image registration task. In the second section, we describe some basic image processing techniques which allow us to construct a function-representation of an image. Another crucial point in many applications is how to compare two given images and how to choose meaningful distance function $\mathfrak{D}$. In Section 3.3, we give a short overview of this task and introduce a new distance function for the special case when the images are made by different modalities. In the last section, we summarize the chapter and discuss some further aspects of the minimization problem (3.1).

3.1 From 3D Pose-Estimation to Image Registration

The 3D pose-estimation problem appears in many applications, like object recognition, data fusion or computer vision. For two given finite sets $\mathcal{X} = \{x_1, \ldots, x_m\} \subset \mathbb{R}^3$ and $\mathcal{Y} = \{y_1, \ldots, y_m\} \subset \mathbb{R}^3$ the task arises to estimate an Euclidean transformation $M \in SE(n)$ which transforms $\mathcal{X}$ best into $\mathcal{Y}$. This task becomes particularly challenging in the case, where the point-wise correspondence is unknown. Let S_m denote the set of bijections of the form $\pi : \{1, \dots, m\} \to \{1, \dots, m\}$, then we can formulate the 3D pose-estimation problem in terms of the optimization problem

$$\min_{M \in SE(3), \pi \in S_m} \sum_{i=1}^{m} \|M \cdot x_{\pi(i)} - y_i\|^2 \tag{3.2}$$

This problem is known to be non-convex and NP-hard. If the first component M of the cost function is fixed, the remaining problem is a discrete combinatorial optimization and the complexity increases exponentially in m. In the literature, this sub-problem is sometimes solved using a branch-and-bound method, see e.g. [66] and [67]. There exist also a couple of global optimization algorithms for (3.2), see e.g. [42], but due to their complexity, these methods work only for small m in acceptable time. Moreover, these algorithms are often highly sensitive to outliers and noise. Therefore, a common approach in the literature is to construct an appropriate approximation of the cost function in (3.2). We refer to [35, 44, 42, 38] and the references therein for an overview. In particular, we mention the Iterative Closed Point (ICP) algorithm [37], which is often seen as the state-of-the-art method for this problem, but it yields only local minima like many other approaches.

One way to get rid of the combinatorial component in the problem (3.2) is to re-interpret the samples x_i and y_i: The elements $x_i \in \mathcal{X}$, for instance, can be seen as measurements of a random vector with the density-function p_x. Following Duda and Hart in [68] we get an approximation of this density by

$$p_{\mathcal{X}}(z) \approx \frac{1}{m} \sum_{i=1}^{m} \phi(z, x_i), \tag{3.3}$$

where $\phi : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ is a user-defined kernel function. For instance, the Gaussian kernel has the form

$$\phi(z, x) = (\pi\sigma^2)^{-n/2} \exp\left(-\frac{1}{2\sigma^2}\|z - x\|^2\right)$$

for fixed $\sigma > 0$. Thus, instead of comparing the point-wise distances between $x_i \in \mathcal{X}$ and $y_j \in \mathcal{Y}$ as it is done in (3.2), we can compare the densities of x and y:

$$\min_{M \in SE(3)} \int\limits_{\mathbb{R}^3} \left(p_{M \cdot \mathcal{X}}(z) - p_{\mathcal{Y}}(z)\right)^2 dz, \tag{3.4}$$

where we used (3.3) for the densities. The advantage of this cost function is that the dependence on the permutation is eliminated. In the literature, also variations of this basic approach are discussed. For instance, one can change the L^2-norm in (3.4) by another distance function; cf. [44] and the references therein. We agree with the authors in [40, 41, 44] and [38] to favor the optimization problem (3.4) instead of (3.2), since (3.4) does not include a hard discrete optimization problem. Moreover, the statistical interpretation reflects the appearance of noise and outliers in many applications.

In the following sections we will introduce another application, the so-called image registration problem, which will end up with the same optimization problem (3.4). The only difference of the image registration problem to (3.4) will be that $p_{\mathcal{X}}$ and $p_{\mathcal{Y}}$ are interpreted as images instead of densities. Thus, although it is not the main intention of our work, each of the algorithms, which we will develop in the Chapters 5 and 6 can be also applied to the pose-estimation task.

3.2 Representation of an Image

A gray-scale image $\hat{I}$ relates every element of a n-dimensional lattice, to a natural number. Throughout this work, we assume that this lattice is the intersection of $\mathbb{Z}^n$ with a given bounded domain $V \subset \mathbb{R}^n$, the so called region of interest. Thus, we set the distance between two neighboring elements, also called pixels, equal one. With these preliminaries at hand, we get a first, digital form of the image $\hat{I}$:

$$\hat{I} : V \cap \mathbb{Z}^n \to \mathbb{N}_0. \tag{3.5}$$

We usually interpret this function in such a way that $\hat{I}(x) = 0$ if the pixel in x is black. Moreover, the larger the value of $\hat{I}(x)$ gets, the brighter the pixel is (see Fig. 3.1).

Although standard formulations of the registration problem focus on two- or three-dimensional images (e.g. $n = 2, 3$), the subsequent mathematical analysis is valid in any dimension. Standard sampling techniques from signal processing allow to extend $\hat{I}$ to a smooth, band-limited function on V (see e.g. [69]) and to recover the discrete version of the image without any loss of information. Therefore, we can interpret $\hat{I}$ as a discretized version of a smooth, real-valued function

$$I : \mathbb{R}^n \to [0, \infty), \quad I \in C^\infty(\mathbb{R}^n, \mathbb{R}). \tag{3.6}$$

One can obtain this second, differentiable form I of an image via convolution of $\hat{I}$ with a n-dimensional sinc-function. Thus, for the sinc-function

$$\operatorname{sinc}(x_1, \dots, x_n) := \frac{\sin(\pi x_1) \dots \sin(\pi x_n)}{\pi^n x_1 \dots x_n}$$

we have

$$I(x) := \hat{I}[x] \star \operatorname{sinc}(x) := \sum_{k \in \mathbb{Z}^n} \hat{I}[k] \operatorname{sinc}(x - k). \tag{3.7}$$

Here, we define $\hat{I}$ by $\hat{I}(x) = 0$ for all $x \in \mathbb{Z}^n \setminus V$ for this approach. With this construction at hand, one can easily see that $I(x) = \hat{I}(x)$ holds for all $x \in \mathbb{Z}^n$. Moreover, if the differentiable form (3.6) of an image instead of (3.5) is initially given, then we can construction the discretized form by $\hat{I}[x] := I(x)$. If additionally I was band-limited, then it is possible to show the identity $I(x) = \hat{I}[x] \star \operatorname{sinc}(x)$ for all $x \in \mathbb{R}^n$ (cf. [69]). Typically an approximation of sinc is used in applications to make the computation of I more efficient and to get smoothed versions of the image $\hat{I}$ (, see e.g. [28] for details). Moreover, one can use (3.7) to get an explicit form of the derivatives of I.

Images of the form (3.5) or (3.6) can be regarded as elements in a real vector space where, in the first case, the dimension is equal to the number of pixels, which can get very large. For

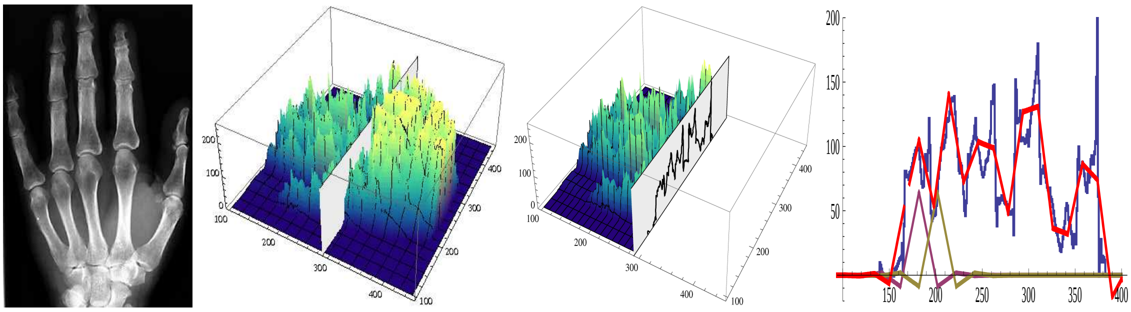


Figure 3.1: (Function representation of an image.) The original image with 250×350 pixels is shown on the left. To the right of it, we plot the graph of its function representation of the form (3.6). On the next picture, we cut this graph with a plane (gray). The resulting one-dimensional graph is shown on the right (blue). We get the red curve by an approximation of the blue one, using wavelet basis-functions like the violet and green ones and their translations.

example, the image in Fig. 3.1 is of the size 250×350, so the dimension is 87500. In various image processing tasks it is therefore reasonable to start with a reduced, or smoothed form of the image and raise the amount of structures step by step. This is known in the literature as a multiresolution or an image pyramid approach. More precisely, we set a finite-dimensional vector space $\mathcal{S} \subset L^2(\mathbb{R}^n, \mathbb{R})$ of k-times continuously differentiable functions in advance and project an image of the form (3.6) with respect to the L^2-norm on $\mathcal{S}$

$$I \in \mathcal{S} \subset C^k(\mathbb{R}^n, \mathbb{R}). \tag{3.8}$$

In this work, we will focus on spline function spaces for constructing $\mathcal{S}$: In order to generate these spaces, we employ the so called B-splines (cf. [70]). In one dimension, these B-splines $B^k(x) : \mathbb{R} \to \mathbb{R}$ are defined by

$$B^0(x) := \begin{cases} 1 & \text{if } -\frac{1}{2} \leqslant x \leqslant \frac{1}{2} \\ 0 & \text{else} \end{cases} \quad \text{and} \quad B^k(x) := \int_{\mathbb{R}} B^{k-1}(s) B^0(s-x) ds, \ k \in \mathbb{N}.$$

We will use their tensor-product to extend these functions to n-dimensional spaces

$$B^k(x_1, \ldots, x_n) := B^k(x_1) \cdot \ldots \cdot B^k(x_n). \tag{3.9}$$

Fig. 3.3 shows the graphs of the splines of first and second order in one dimension. For us, the most important fact of B-splines is that they becomes smoother when the order increases. The corresponding spline function space $\mathcal{S}^k_\lambda$ can now be defined as the vector space generated by all integer-translations of (3.9) (cf. [70])

$$\mathcal{S}^k_\lambda := \left\{ \sum_{\lambda \cdot r \in \mathbb{Z}^n} c_r B^k \left(\frac{x_1}{\lambda} - r_1, \ldots, \frac{x_n}{\lambda} - r_n \right) \,\middle|\, c \in l^2 \right\}, \quad \lambda \in \mathbb{N}. \tag{3.10}$$

The condition that $c \in l^2$ ensures that the L^2-norm of all functions in $\mathcal{S}^k_\lambda$ exists.
Note that we will only work with images that have a bounded support $V \subset \mathbb{R}^n$. Thus, it is sufficient to consider finitely many linear combinations of tensor product functions B^k.

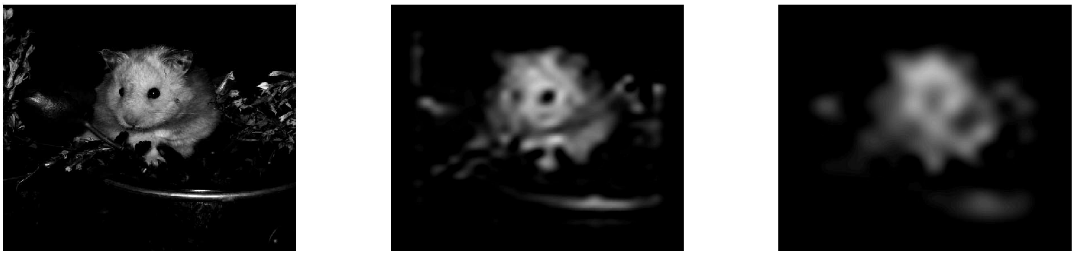


Figure 3.2: (Original and two spline-approximations of an image.) The original gray-scale image with 500×500 pixels is given on the left side. In the middle and on the right side, we project the left image to the spline function spaces $\mathcal{S}_{15}^3$ and $\mathcal{S}_{30}^3$, respectively.

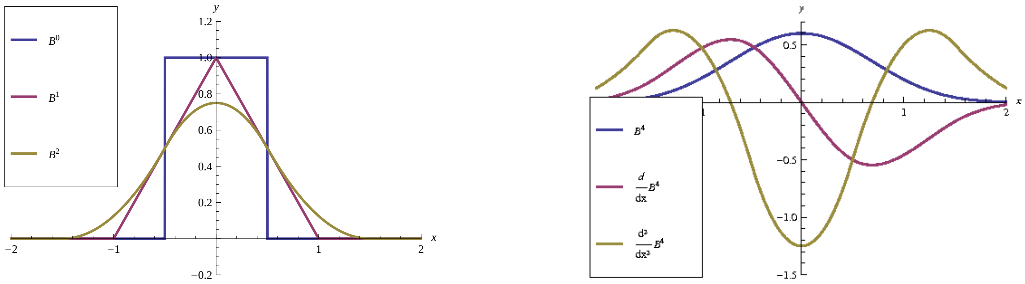
$$\mathcal{S}_{\lambda,V}^k := \left\{ \sum_{\lambda \cdot r \in \mathbb{Z}^n \cap V} c_r B^k \left(\frac{x_1}{\lambda} - r_1, \ldots, \frac{x_n}{\lambda} - r_n \right) \;\middle|\; c_r \in \mathbb{R} \right\}, \quad \lambda \in \mathbb{N}. \tag{3.11}$$

In this case, the condition $c \in l^2$ is always fulfilled and we have $\mathcal{S}_{\lambda,V}^k \subset \mathcal{S}_\lambda^k$. In this work, we will use $\mathcal{S} = \mathcal{S}_{\lambda,V}^k$ for the finite-dimensional vector space $S \subset L^2(\mathbb{R}^n, \mathbb{R})$ in eq. (3.8) and we write $P_{\lambda,V}^k : L^2(\mathbb{R}^n, \mathbb{R}) \to \mathcal{S}_{\lambda,V}^k$ for the orthogonal projection. The parameter $\lambda \in [0, \infty)$ is used in image processing to neglect high frequency informations of images (cf. [71]). That is, if λ increases, the projection $I_\lambda := P_{\lambda,V}^k I$ of an image I is smoothed (see Fig. 3.2 for an illustration).

Of course, there are several additional properties of splines (see [72] for an overview) which make them an excellent tool for image processing. First and foremost, splines offer a quite good cost-performance tradeoff in comparison to other interpolation methods (cf. [72]). Further, splines have a lesser tendency to oscillate, which is due to the minimum curvature property (cf. [73]). Moreover, if we increase the degree k of the B-splines for $I_1 \in \mathcal{S}_1^k$, we get the form (3.7) of the image by the limiting process $k \to \infty$ (see e.g. [74]).

In the literature, all three kinds (3.5), (3.6) and (3.8) of image representations are in use. The first form (3.5) appears e.g in several filtering, smoothing or restoration algorithms (see e.g. [75]) but also in image registration methods (see e.g. [10]) where the necessary derivatives are approximated by finite differences. Representation (3.6) is used in edge-detection algorithm (see e.g. [28, 76]) which is the starting point for several object reconstruction [27] and camera calibration methods. However, the third form (3.8) becomes more and more popular in modern image-processing methods. To give some examples: Forster [77] invented a new kind of complex B-spline for image denoising, and a mixture of splines and wavelets are used in a new form of data reduction called “Compressed Sensing” (cf. [78, 79]). Moreover, Unser studied the multiscale-processing (cf. [80, 81, 71]), which is used e.g. for a registration algorithm in [24].

In this work, we perform each algorithm with images of the form (3.8), i.e. with their B-spline representation $I \in \mathcal{S}_\lambda^k$. Therefore, we can easily embed each method in a multiscale-approach and work with smoothed images in each step. But in the convergence analysis of our algorithms considerations we will also allow images of the general form (3.6).



$$B^1(x) = \begin{cases} 1+x & \text{for} \quad -1 < x \leqslant 0 \\ 1-x & \text{for} \quad 0 < x \leqslant 1 \\ 0 & \text{else} \end{cases}$$

$$B^2(x) = \begin{cases} \frac{1}{8}(9+12x+4x^2) & \text{for} \quad -\frac{3}{2} < x \leqslant -\frac{1}{2} \\ \frac{1}{4}(3-4x^2) & \text{for} \quad -\frac{1}{2} < x \leqslant +\frac{1}{2} \\ \frac{1}{8}(9-12x+4x^2) & \text{for} \quad \frac{1}{2} < x \leqslant \frac{3}{2} \\ 0 & \text{else} \end{cases}$$

$$B^3(x) = \begin{cases} \frac{1}{6}(8+12x+6x^2+x^3) & \text{for} \quad -2 < x \leqslant -1 \\ \frac{1}{6}(4-6x^2-3x^3) & \text{for} \quad -1 < x \leqslant 0 \\ \frac{1}{6}(4-6x^2+3x^3) & \text{for} \quad 0 < x \leqslant 1 \\ \frac{1}{6}(8-12x+6x^2-x^3) & \text{for} \quad 1 < x \leqslant 2 \\ 0 & \text{else} \end{cases}$$

$$B^4(x) = \begin{cases} \frac{1}{384}(5+2x)^4 & \text{for} \quad -\frac{5}{2} < x \leqslant -\frac{3}{2} \\ \frac{1}{96}(55-20x-120x^2-80x^3-16x^4) & \text{for} \quad -\frac{3}{2} < x \leqslant -\frac{1}{2} \\ \frac{1}{192}(115-120x^2+48x^4) & \text{for} \quad -\frac{1}{2} < x \leqslant \frac{1}{2} \\ \frac{1}{96}(55+20x-120x^2+80x^3-16x^4) & \text{for} \quad \frac{1}{2} < x \leqslant \frac{3}{2} \\ \frac{1}{384}(-5+2x)^4 & \text{for} \quad \frac{3}{2} < x \leqslant \frac{5}{2} \\ 0 & \text{else} \end{cases}$$

Figure 3.3: The first image shows the B-splines of order zero, one and two. The second image shows the B-spline of order four, with its first and second derivative.

3.3 Image Distance Measures

Although the last decades have provided an enormous variety of different registration algorithms, there is a general agreement on how to measure the distance between two image. For monomodal registration, the sum of squared differences (SSD) can be regarded as the state-of-the-art approach. In the other case, where two images are made by different modalities, the mutual information (MI) is usually taken for a comparison. In some applications, also additional informations of the images are available for the comparison. For instance, in some applications a set of user identified points which are, in some sense, meaningful are taken into account for the comparison of two images. But in this work, we will consider image distance measures based only on pixel data. In contrast to the previous section, now our main focus will lie on medical imaging.

3.3.1 Sum of Squared Differences

Perhaps the most straightforward way to compare two images R and T of the discrete form (3.5), made by equal or similar modality, is to consider the difference-image $(R-T)^2$. The more this image turns black, the more R coincides with T. Therefore the sum of all pixels of $(R-T)^2$, the so called SSD-measure, provides a good characterization of the correspondence. By passing from discrete to continuous images (3.6), as discussed in the previous section, this measure becomes the L^2-norm, i.e.

$$\mathfrak{D}_{\mathrm{SSD}}(R,T) := \frac{1}{|V|}\int_V \big(R(x)-T(x)\big)^2 dx. \tag{3.12}$$

Here V denotes region of interest and $|V|$ its n-dimensional volume. In the literature, this measure is the most common choice, since it can be easily calculated, and an algorithm to solve the registration problem (3.1) can be chosen in such a way to exploit the structure of $\mathfrak{D}_{SSD}$. For example, in [24, 82] and [83] Levenberg-Marquard or Gauss-Newton methods are presented to solve the minimization problem. Furthermore, Benhimane and Malis constructed in [61] an efficient second order algorithm for monomodal registration using the distance measure (3.12). Without exception, all those algorithms show an excellent convergence behavior with comparatively low numerical cost. (See e.g. [84] for a comparison of different Gauss-Newton methods.)

3.3.2 Mutual Information

One important subproblem in image registration tackles the case, where the two given images are made from different modalities. For example, one image may be from a MR-scan, while the other is a CT-data set (cf. Fig. 3.4). Apart from some edge-detecting algorithms (see [20] or [85]), the mutual information

$$\mathfrak{D}_{\mathrm{MI}}(R,T) := H(\rho_R) + H(\rho_T) - H(\rho_{R,T}) \tag{3.13}$$

is the standard tool for comparing these kind of images. Here, ρ_R, ρ_T and $\rho_{R,T}$ denote the gray-value densities of R, T and the joint gray-value density, respectively. H represents the entropy-function. To avoid confusion, for the definition in (3.13) we need another way of defining images than in the previous section: For a given region of interest $V \subset \mathbb{R}^n$, we assume that $x \in V$, or $x \in V \cap \mathbb{Z}^n$ in the case of discrete images, is a uniformly distributed random vector. Thus, in (3.13), $R(x)$ and $T(x)$ are not considered as real-valued functions of the form $V \to \mathbb{R}$, but as random variables. Therefore, the densities and entropies appearing in (3.13) are well defined objects (cf. [86, 10]).

In the case when R and T are discrete images, an explicit formula of the entropies is available:

$$H(\rho_S) = -\sum_{i\in\mathbb{N}_0^q} p(i)\log(p(i)) \quad \text{with} \quad p(i) := P(S(x)=i) := \frac{|S^{-1}(i)|}{|V \cap \mathbb{Z}^n|}, \quad i \in \mathbb{N}_0^q \tag{3.14}$$

for $S = R$, $S = T$ with $q = 1$ and $S = (R,T)$ with $q = 2$, respectively. Here, $S^{-1}(i)$ denotes the preimage of i under S.

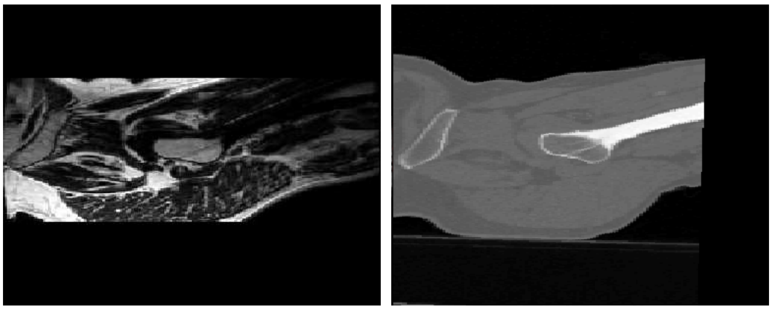


Figure 3.4: Medical example of two images, made by different modalities. Both images show the same cross-section of the hip of a patient. The left one is made by MR-imaging, and the right one by CT-imaging.

Unfortunately, there are no explicit formulas to calculate the density or entropy of a continuous image like in form (3.6). Up to now, even a standard approach to approximate the entropy is missing in the literature. In [10], an approximation of H is given via

$$H(\rho_S) \approx \tilde{H}_m(\rho_S) := -\frac{1}{m}\sum_{k=1}^{m} \log\left(\frac{1}{m}\sum_{j=1}^{m} g_{q,\Sigma}\left(S(Y_k) - S(X_j)\right)\right), \quad m \in \mathbb{N} \tag{3.15}$$

for $S = R$, $S = T$ and $S = (R, T)$. Here, $g_{q,\sigma}$ is the q-variate Gaussian density with the covariance-matrix Σ and mean 0, i.e.

$$g_{q,\Sigma} : \mathbb{R}^q \to \mathbb{R}, \quad g_{q,\Sigma}(x) := (2\pi)^{-q/2}(\det \Sigma)^{-1/2} \exp\left(-\frac{1}{2}x^\top \Sigma x\right), \quad \Sigma \in \mathbb{R}^{q\times q}.$$

The samples $X_k,\ Y_k$ in (3.15) are several points in the support of the image. In the literature, these points are usually chosen by random.

In [85], Haber and Modersitzki mentioned several drawbacks regarding this measure. Most of them occur because one would need a smooth function S to handle efficient algorithms, like gradient or Newton methods, for optimizing cost functions of the form $\mathrm{Diff}(\mathbb{R}^n) \to \mathbb{R}$, $\varphi \mapsto H(\rho_{S\circ\varphi})$. However, it is a difficult task to get a sufficiently good approximation of the density-function of $S \circ \varphi$. We mention three more drawbacks. First, there are numerical problems which appear because of the interplay between the log and the exp function. For example, if l and m are large, then the sum in (3.15) is (numerically) not even commutative. Second, calculating the first and second derivatives of $H(\rho_{S\circ\varphi})$ with respect to φ provides quite long expressions and are therefore of high numerical cost. Finally, since $\mathfrak{D}_{\mathrm{MI}}$ does not satisfy the triangle inequality or other properties like positive definiteness, we cannot interpret $\mathfrak{D}_{\mathrm{MI}}$ as a metric or distance function.

3.3.3 δ-Distance

Because of the problems with the mutual information measure mentioned in the previous subsections, we introduce another measure for multimodal registration. Our basic idea is that in a medical picture, all pixels of one sort of tissue are approximately related to the same gray-value. This is true no matter which current modality is chosen to acquire the image. Therefore, let R and T be images of the same anatomy, made by different modalities. If they have a perfect match, i.e. if one image lies exactly on the top of the other, each pair of pixels (x, y) in the region of interest (ROI) satisfies

$$R(x) - R(y) = 0 \Leftrightarrow T(x) - T(y) = 0. \tag{3.16}$$

In the following, we will introduce an image distance measure, which penalizes differences from this property.

Definition 3.1

A L^2-integrable function of the form $\delta : \mathbb{R} \to \mathbb{R}$ is called a *window-function*, if δ is non-negative, the L^2-norm is equal one and $\delta(x) = \delta(-x)$ for all $x \in \mathbb{R}$. For a given window-function, we define the *δ-distance* for images R and T of the form (3.6) or (3.8) by

$$\mathfrak{D}_\delta(R,T) := \frac{1}{|V|^2} \iint\limits_{V\times V} \Big(\delta(R(x) - R(y)) - \delta(T(x) - T(y))\Big)^2 dxdy. \tag{3.17}$$

In our applications, we chose a cubic B-spline for the window-function δ. The benefit of such window-functions is that we weight the differences in the gray values in each image. This is particularly helpfull, since we know the used modalities to create the images in most applications. Thus, we can adjust the window-function to the respective problem (cf. Section 6.2).

The following lemma shows the connection between the δ-distance and the perfect match property (3.16).

Proposition 3.1

Suppose that the images $R, T : \mathbb{R}^n \to \mathbb{R}$ are continuous functions. If the window-function δ satisfies $\delta(x) = \delta(0) \Rightarrow x = 0$, then $\mathfrak{D}_\delta(R,T) = 0$ implies the property (3.16).

If the window-function δ is of the form $\delta(x) := \frac{1}{2\varepsilon}\chi_{[-\varepsilon,\varepsilon]}(x)$, where $\chi_{[-\varepsilon,\varepsilon]}$ denotes the characteristic function of the interval $[-\varepsilon, \varepsilon]$, then $\mathfrak{D}_\delta(R,T) = 0$ is equivalent to

$$|R(x) - R(y)| < \varepsilon \Leftrightarrow |T(x) - T(y)| < \varepsilon \quad \textit{for all } x, y \in V. \tag{3.18}$$

Proof

Since R and T are continuous, $\mathfrak{D}_\delta(R,T) = 0$ is equivalent to

$$\delta(R(x) - R(y)) = \delta(T(x) - T(y)), \quad \text{for all } x, y \in V. \tag{3.19}$$

If δ satisfies $\delta(x) = \delta(0) \Rightarrow x = 0$, this implies (3.16). If δ is of the form $\frac{1}{2\varepsilon}\chi_{[-\varepsilon,\varepsilon]}$, we have $\delta(x) \neq 0$ if, and only if $|x| \geq \varepsilon$. Thus, (3.19) is equivalent to (3.18). □

Obviously there is a difference between the “perfect match” and the result of the registration problem, which has the task to search for a deformation φ such that $\mathfrak{D}(R \circ \varphi, T)$ becomes minimal. In Table 3.1 we have summarized the characterizations of the perfect match properties for the different image distance measures $\mathfrak{D}$. For real image data R there is usually no

Image distance	Characterization of a perfect match
SSD	$R(x) = T(x)$ for all $x \in ROI$
MI	$\mathbb{E}_{\rho_{R,T}}\left[\log \frac{\rho_{R,T}}{\rho_R \rho_T}\right] = 1, \quad \mathbb{E}$: Expectation value
δ-Distance	$R(x) = R(y) \Leftrightarrow T(x) = T(y)$ for all $x \in ROI$

Table 3.1: Characterizations of a perfect match for different image distance measures.

deformation such that these perfect match properties are satisfied for $R \circ \varphi$, since R could be disturbed by imaging errors or medical treatment. Additionally a given modality does not distinguish between all different sorts of tissue, which contradicts relation (3.16). However, minimizing $\mathfrak{D}(R \circ \varphi, T)$ is equivalent to the search of a deformation φ such that the perfect match characterization for T and $R \circ \varphi$ is satisfied as fully as possible. Therefore, relation (3.16) is not a necessary condition; it just describes the idealistic goal, while the realistic goal is to find the minimum of the function defined in eq. (3.17).
In the following, we will also study a discrete version of the δ-distance for the case when both images are discrete.

Definition 3.2
Let the reference and template images are of the form $\hat{R}, \hat{T} : V \cap \mathbb{Z}^n \to \mathbb{N}_0$, $V \subset \mathbb{R}^n$. Then, the *discrete δ-distance* is given by

$$\mathfrak{D}_\delta(\hat{R}, \hat{T}) := \frac{1}{|V \cap \mathbb{Z}^n|^2} \sum_{i,j \in V \cap \mathbb{Z}^n} \Big(\delta(\hat{R}[i] - \hat{R}[j]) - \delta(\hat{T}[i] - \hat{T}[j]) \Big)^2, \tag{3.20}$$

where we set $\delta(x) := \chi_{[-0.5,0.5]}(x)$.

Using the abbreviation $|\hat{I}^-(\alpha)|$ for the number of all pixels in the image $\hat{I}$ which have the gray-value α, then we get another representation of the discrete δ-function.

$$\mathfrak{D}_\delta(\hat{R}, \hat{T}) = \frac{1}{|V \cap \mathbb{Z}^n|^2} \left(\sum_{\alpha \in \mathbb{N}_0} |\hat{R}^-(\alpha)|^2 + \sum_{\beta \in \mathbb{N}_0} |\hat{T}^-(\beta)|^2 - 2 \sum_{\alpha,\beta \in \mathbb{N}_0} |\hat{R}^-(\alpha) \cap \hat{T}^-(\beta)|^2 \right). \tag{3.21}$$

We want to point out that in contrast to Definition 3.1, we fixed the window-function in this case. However, if we exchange the window-function by another one, whose support is also bounded by the interval $(-1, 1)$, then the discrete δ-distance would only differ by a constant factor. We will see in the next proposition that this kind of distance measure does not only penalize differences from the perfect match property (3.16). It also satisfies the properties of a metric.

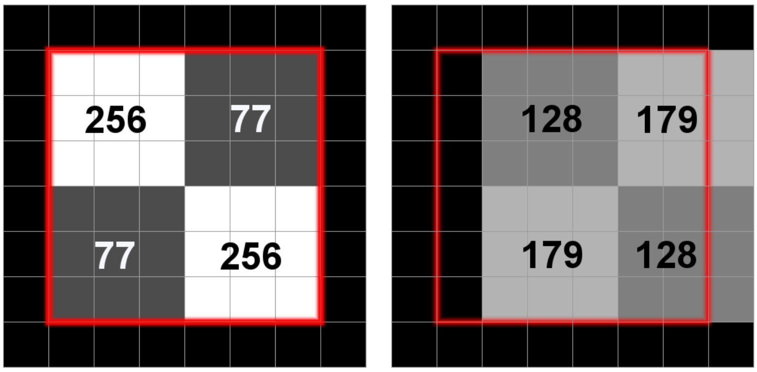


Figure 3.5: Reference and template image, each with 8 × 8 pixels. The numbers represent the gray-value of the fields and the red boundary describes the ROI

Proposition 3.2

On the space of all discrete images of the form $\hat{I} : V \cap \mathbb{Z}^n \to \mathbb{N}_0$ we define the equivalence relation

$$\hat{I}_1 \sim \hat{I}_2 \quad :\Leftrightarrow \quad \hat{I}_1 = f \circ \hat{I}_2 \textit{ for a bijective function } f : \mathbb{N}_0 \to \mathbb{N}_0.$$

Then, the extension of the discrete δ-distance $\mathfrak{D}_\delta$ defined by (3.20) to the equivalence classes via $\mathfrak{D}_\delta([\hat{R}], [\hat{T}]) := \mathfrak{D}_\delta(\hat{R}, \hat{T})$ is well defined. Moreover,

$$\Big([\hat{R}], [\hat{T}]\Big) \mapsto \sqrt{\mathfrak{D}_\delta([\hat{R}], [\hat{T}])}$$

is a metric on the space of all equivalence classes.

Proof

From (3.21) we get directly the well-definedness of $\mathfrak{D}_\delta([\hat{R}], [\hat{T}])$. Now, we consider the equation $\mathfrak{D}_\delta(\hat{R}, \hat{T}) = 0$ for two discrete images $\hat{R}$ and $\hat{T}$. Using (3.20), this equation is satisfied if and only if

$$\delta(\hat{R}[i] - \hat{R}[j]) = \delta(\hat{T}[i] - \hat{T}[j]) \quad \text{for all } i, j \in V \cap \mathbb{Z}^n. \tag{3.22}$$

Since the images $\hat{R}$and $\hat{T}$ are of the form $V \cap \mathbb{Z}^n \to \mathbb{N}_0$, we have $\hat{R}[i] - \hat{R}[j] \in \mathbb{Z}$ and $\hat{T}[i] - \hat{T}[j] \in \mathbb{Z}$. Using $\delta = \chi_{[-0.5,0.5]}$, the statement (3.22) is equivalent to

$$\hat{R}[i] = \hat{R}[j] \quad \Leftrightarrow \quad \hat{T}[i] = \hat{T}[j] \quad \text{for all } i, j \in V \cap \mathbb{Z}^n. \tag{3.23}$$

Hence, $\hat{R} \sim \hat{T}$ implies (3.23) and therefore $\mathfrak{D}_\delta(\hat{R}, \hat{T}) = 0$. fs $\mathfrak{D}_\delta(\hat{R}, \hat{T}) = 0$ is satisfied, the argumentation above yields (3.23). Therefore, we can construct a bijection $f : \mathbb{N}_0 \to \mathbb{N}_0$ with $f(\hat{T}[j]) = \hat{R}[j]$ for all $j \in V \cap \mathbb{Z}^n$. Thus, $\mathfrak{D}_\delta(\hat{R}, \hat{T}) = 0$ is equivalent to $\hat{R} \sim \hat{T}$. We get the remaining properties of a metric directly from (3.20). □

Example:

To illustrate the use of the δ-distance in image-registration, we give a toy example in Fig. 3.5.

The right image shows the reference R and the left image depicts the template T, the region of interest is the 6×6 array in the middle. The four numbers in each image denote the gray scale in the particular 3×3 square. Therefore in the region of interest, R has the gray-values $\alpha = 0, 128, 179$, and simultaneously T has the gray values $\beta = 77, 256$. We get

$$\begin{aligned}\mathfrak{D}_\delta(R,T) =& \frac{5}{36^2}\big((6^2+15^2+15^2)+(18^2+18^2)-2(3^2+12^2+3^2+3^2+3^2+12^2)\big)\\ =& \frac{15}{8}.\end{aligned}$$

It is of particular interest that the result does not depend on the specific gray-values of the images, but only on the amount of pixels sharing the same gray-value. In comparison, we also consider the case where the template is translated one pixel to the left, which can be seen as a perfect match between both images. Now, the term in the integral of eq. (3.17) vanishes for each pair of pixels (x, y) in the region of interest and the measure becomes zero.

$$\begin{aligned}\mathfrak{D}_\delta(R,T) =& \frac{5}{36^2}\big((18^2+18^2)+(18^2+18^2)-2(18^2+0^2+18^2+0^2)\big)\\ =& 0.\end{aligned}$$

Thus, if we move the reference one pixel to the left, one would intuitively identify a good alignment of both images. In this case, the δ-distance yields also the lowest value.

The main advantage of $\mathfrak{D}_\delta$ is its relation to the classical SSD-measure: Due to a comparison of eq.(3.17) with eq.(3.12) the multimodal registration task for two images R and T can be interpreted as a monomodal registration of two images $\tilde{R}$ and $\tilde{T}$ by doubling the dimension, namely

$$\tilde{R}(x,y) := \delta(R(x)-R(y)) \quad \text{and} \quad \tilde{T}(x,y) := \delta(T(x)-T(y)). \tag{3.24}$$

Similarly to $\mathfrak{D}_{\text{SSD}}$, the essential structure of $\mathfrak{D}_\delta$ consists of the L^2-norm and thus all established techniques in monomodal registration are applicable to the multimodal case.
Additionally, we demonstrate in the following proposition that there is also a relation between the $\mathfrak{D}_\delta$ and $\mathfrak{D}_{\text{MI}}$.

Proposition 3.3

Let $g_{q,\Sigma}$ denote the q-variate Gaussian density with mean zero and variance-matrix Σ, and let the samples $\{X_j\}_{j=1}^m$, $\{Y_k\}_{k=1}^m$ be independently and uniformly distributed in the intersection of the supports of the images R and T. Using the abbreviation

$$G_{k,m,\sigma} = \frac{1}{m}\sum_{j=1}^m g_{1,\sigma}\big(R(Y_k)-R(X_j)\big)g_{1,\sigma}\big(T(Y_k)-T(X_j)\big),$$

the exponential of the approximated joint entropy $\tilde{H}_m(\rho_{R,T})$ in (3.15) is given by the geometric mean of all $G_{k,m,\sigma}$'s:

$$\exp(-\tilde{H}_m(\rho_{R,T})) = \sqrt[m]{\prod_{k=1}^m G_{k,m,\sigma}}. \tag{3.25}$$

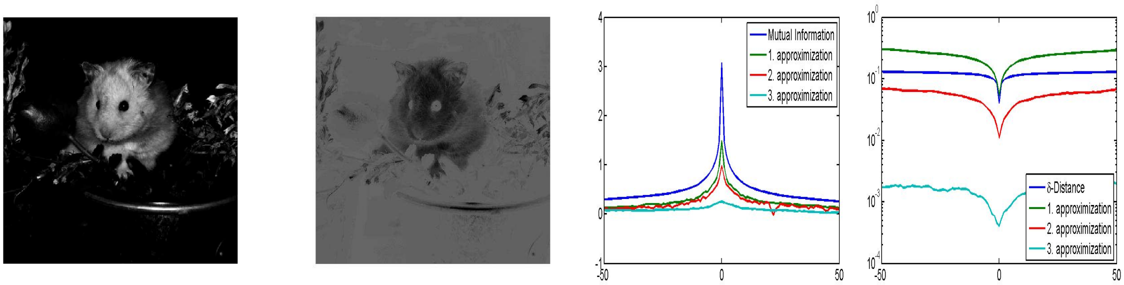


Figure 3.6: Comparison of the mutual information measure with the δ-Distance. We took the first image for the reference R and the second image for the template T. Third image: Mutual information and its approximation, using (3.15) with $(m, n, \sigma) = (5000, 5000, 5)$, $(1000, 1000, 25)$, $(200, 200, 625)$, of R and T while translating R horizontally form -50 pixel to $+50$ pixel. Fourth image: δ-distance for the same appreciation of assets. We used $\delta(x) = 5\chi_{[-0.1,0.1]}(x)$ and $\delta(x) = g_{1,\sigma}(x)$ with $\sigma = 5, 25, 625$. In the cases $\delta = g_{1,\sigma}$, we used a Monte-Carlo approximation of the integral in (3.17) is used with 5000^2, 1000^2 and 200^2 samples.

Furthermore, by setting $\delta(x) = g_{1,\sigma}(x)$, the scalar product of $\tilde{R}(x, y)$ and $\tilde{T}(x, y)$ with respect to the L^2-norm can be approximated by the arithmetic mean of all $G_{k,m,\sigma}$'s.

$$\left\langle \tilde{R}(x, y), \tilde{T}(x, y) \right\rangle_{L^2} = \lim_{m \to \infty} \frac{1}{m} \sum_{k=1}^{m} G_{k,m,\sigma}. \tag{3.26}$$

Proof

Equation (3.25) follows directly from the definition of $\tilde{H}$ in (3.15) and (3.26) is a consequence of the Monte-Carlo approach for the approximation of integrals [87]. □

Proposition 3.3 gives rise to suspicion that a similar result like (3.26) should also hold for the approximation of the entropy. However, as far as we know, there is no convergence analysis of $\tilde{H}_m \to H$, only the approximation of the densities ρ_R, ρ_T and $\rho_{R,T}$ using the Parzen window method is discussed in [10, 86] or [68].

Proposition 3.3 hints at the common behavior of $\mathfrak{D}_{\mathrm{MI}}$ and $\mathfrak{D}_\delta$, if we disturb the reference image with a volume-preserving transformation φ, i.e. if we consider the map $\varphi \mapsto \mathfrak{D}(R \circ \varphi, T)$. If we choose the mutual information measure for $\mathfrak{D}$, we can exploit the fact that $H(\rho_{R\circ\varphi})$ and $H(\rho_T)$ remain constant under the volume-preserving transformation φ. Therefore, the extrema of $\mathfrak{D}_{\mathrm{MI}}(R \circ \varphi, T)$ are equal to the extrema of the joint entropy $\varphi \mapsto H(\rho_{R\circ\varphi,T})$, which can be approximated by the geometric mean of the $G_{k,m,\sigma}$. If we choose the δ-distance for studying the map $\varphi \mapsto \mathfrak{D}(R \circ \varphi, T)$, we can use (3.24) and get

$$\mathfrak{D}_\delta(R \circ \varphi, T) = \frac{1}{|V|^2} \left(\|\tilde{R} \circ \varphi\|^2_{L^2} + \|\tilde{T}\|^2_{L^2} - 2 \left\langle \tilde{R} \circ \varphi, \tilde{T} \right\rangle_{L^2} \right).$$

Again, the first two terms on the right side remain constant under volume-preserving transformations of R. Therefore, the extrema of $\mathfrak{D}_\delta(R \circ \varphi, T)$ are equal to the extrema of the scalar-product, which can be approximated by the arithmetic mean.

We finish this section with a comparison of the mutual information measure with the δ-distance.

Example:
In this example, we use the left image in Fig. 3.2 for the reference R and the left image in Fig. 3.6 for the template T. Here, T is constructed by an affine distortion of the gray values of R,

$$T(x) := -\frac{1}{2}R(x) + 100.$$

Let $\varphi_t : \mathbb{R}^2 \to \mathbb{R}^2$, $\varphi_t(x, y) := (x, y + t)$ be the horizontal translation of $t \in \mathbb{Z}$ pixels. In the middle of Fig. 3.6 we display the graph of the function $t \mapsto \mathfrak{D}_{MI}(R \circ \varphi_t, T)$, where we used (3.14) to calculate the appearing entropies. We also calculated the graph where the entropies are approximated using formula (3.15). We choose $m = 5000$ and $\sigma = 5$ for the first approximation. The samples X_k, Y_k are chosen independently and uniformly randomly in the whole image domain $[1,\ 520]^2 \cap \mathbb{Z}^2$ (but we use the samples for all t). In the second and third approximation, we choose $m = 1000$, $\sigma = 25$ and $m = 200$, $\sigma = 625$, respectively. On the right side of Fig. 3.6 we display the graph of the function $t \mapsto \mathfrak{D}_\delta(R \circ \varphi_t, T)$ where we use the discrete version (3.21) for the measure $\mathfrak{D}_\delta$ (blue graph). Additionally, we calculate $\mathfrak{D}_\delta$ by the formula (3.17) using $\delta(x) = g_{1,\sigma}(x)$ for $\sigma = 5, 25, 625$. The integral appearing in (3.17) was approximated with a Monte-Carlo method using the same samples as in the corresponding approximation of the mutual information.
All eight graphs show a global extremum in $t = 0$. This is a desirable property for the use in multimodal image registration, since for $t = 0$ the images $R \circ \varphi_t$ and T are identical, despite the transformation of the gray values in R. Thus, both measure can detect the optimal alignment of R and T. We also notice that many local extrema appear in the approximation of $\mathfrak{D}_\delta(R \circ \varphi_t, T)$ or $\mathfrak{D}_{MI}(R \circ \varphi_t, T)$, if we reduce the number of samples. (This effect is for the most part independent from a variation of σ.) In the middle of Fig. 3.6 we notice that the approximation (3.15) converges to (3.14) for the simultaneous limits $m \to \infty$ and $\sigma \to 0$.

In applications, the choice of σ is rather problematic (see e.g. [10]). An optimization process is suggested in [88] to get the best value for the tuning-parameter σ, but usually it is considered to be given a priori, e.g. by the distribution of the gray-values in the original images of the form (3.5) (see e.g. [52, 89, 90]). In comparison, we see no convergence to $\mathfrak{D}_\delta(R \circ \varphi_t, T)$ (blue graph) on the right side of Fig. 3.6 if we use finer approximations. This is not surprising, since we used a different window-function $\delta(x)$ for each graph. Therefore, the approximations are related to different image distance measures of the form (3.17). Convergence is only given for a fixed σ (and therefore for a fixed $\delta(x)$) and for $m \to \infty$. The blue graph would still not be the limiting curve, since its window-functions have a different form for each σ. However, in this example we can use a lot of different types of window-functions to detect the best alignment of R and T. In the case of real medical applications, additional information is available besides the raw pixel data, like the recording technology or the appearing sorts of tissue. This information can be used for a fine-tuning of the window-function $\delta(x)$.

3.4 The Optimization Problem

With these preliminaries at hand, we can state the final formulation of the registration task. Consider two continuous images $R : V_R \to \mathbb{R}$ and $T : V_T \to \mathbb{R}$ of the form (3.8) or of (3.6), the

Table 3.2: Multiresolution Approach for Image Registration

Given: images $R, T \in C(\mathbb{R}^n, \mathbb{R})$ with compact support.
Parameter: $k \in \mathbb{N}$, $\{\lambda_i\}_{i\in\mathbb{N}} \subset [0, \infty)$ with $\lim\limits_{i\to\infty} \lambda_i = 0$

Step 1.
Choose an initial point $g_1 \in G$. Set $i = 1$.

Step 2.
Set

$$g_{i+1} = \arg\min_{g\in G} \mathfrak{D}^V(P^k_{\lambda_i} R \circ \rho_g, P^k_{\lambda_i} T) \tag{3.28}$$

and use g_i for the initial guess to solve the optimization problem.

Step 3.
Set $i = i + 1$ and goto step 2.

so-called reference and template, with the given domains $V_R, V_T \subset \mathbb{R}^n$. Additionally, specify the group of deformations, e.g. $G = SE(n)$ or $G = SA(n)$ and specify one of the similarity measures $\mathfrak{D}^V_{SSD}$, $\mathfrak{D}^V_{MI}$ or $\mathfrak{D}^V_{\delta}$ investigated in Section 3.3. Here, the superscripted letter "V" points out that each measure depends on a given domain $V \subset \mathbb{R}^n$. The problem is now to find an element $g^\star \in G$, such that $R \circ \rho_{g^\star}$ matches best to T in the domain $V := \rho_{g^\star}(V_R) \cap V_T$, i.e. we look for

$$\arg\min_{g\in G} \mathfrak{D}^V(R \circ \rho_g, T). \tag{3.27}$$

It is mentioned in [24] that the types of applications, to which the registration is addressed, can be categorized in data fusion, motion estimation and the detection of significant differences. With the previous arguments, the task of finding an adequate measure becomes even more delicate, especially for the third application. The optimization problem (3.27) incorporates all pixels in V to find a good alignment, even those which carry entirely different information. That is why landmarks are still incorporated in several registration algorithms which are sometimes detected automatically, but also provided by medical experts (see e.g. [91]).
Another restriction of the optimization problem (3.27) is due to the domains of definition V_R and V_T. We assume that we can find the whole template in the reference. That is, if $g^\star \in G$ is the solution of (3.27) we have $\overline{V_T} \subset \rho_{g^\star}(V_R)$. This has several reasons: First, it simplifies the problem since the joint domain V is now constant in a certain neighborhood around $g^\star$. Secondly, we have to avoid deformations for which $V_T \cap \rho_g(V_R) = \varnothing$ since this would imply an undesirable extremum. Another reason is inherent in approximations of the distance measures, i.e. if we use either (3.15) to calculate the mutual information or a Monte-Carlo approach to approximate the integrals in (3.12) or (3.17). In these cases we deal with a set of samples in the domain V and if V varied during the optimization process, the set of samples would also vary. Hence, the cost function of (3.27) would have discontinuities,

which we want to avoid. Last but not least, the assumption $\overline{V}_T \subset \rho_{g^\star}(V_R)$ conforms with most of the medically relevant applications. For example, in radiation therapy the first image is used to define the region of interest (ROI), which contains the tumor and organs at risk. This region can be relatively small compared to the domain of definition of the image. The registration task is now to find the ROI in a second image, which is made sufficiently large to guarantee that the ROI is included. In such a case, we would set $V =$ ROI. However, it is known that a small variation of the ROI affects the result of the optimization problem (3.27), see e.g. [92].

If the images R and T are given in form (3.5) or, after an interpolation step, in form (3.6), it is a common approach in image processing not to solve (3.27) directly but to embed this optimization in a multiresolution approach (see [24] for an introduction). In our work, we use the projection $P^k_\lambda : L^2(\mathbb{R}^n, \mathbb{R}) \to \mathcal{S}^k_\lambda$ and choose a decreasing sequence $(\lambda_1, \lambda_2, \ldots) \subset [0, \infty)$. The registration task (3.27) will then be solved by using the scheme in Table 3.2.

This approach is common in image registration (see e.g. [89, 24, 90, 33]): If λ_i is large, a lot of the high frequency part of I is missing in $P^k_{\lambda_i} I$ (see Fig. 3.2). Therefore, the cost function of (3.28) is usually smoother than the cost function of (3.27) and the scheme usually avoids local minima. Moreover, in many examples the registration task (3.28) gives quite good approximations of (3.27), even if λ_i is large. But these are only numerical observations. As far as we know, there exists no convergence analysis of this registration approach.

Of course, B-splines are not the only way of implementing a multiresolution scheme; see e.g. [93, 15, 11, 94] for other approaches in image processing. But besides the outstanding interpolation and approximation properties (see e.g. [72]), a reduction of the high frequency part (i.e increasing λ in $P^k_\lambda I$) implies a reduction of information (i.e. $\mathcal{S}^k_\lambda$ is low dimensional). We will present algorithms which exploit this property.

Of course, an optimization method for solving (3.28) is still needed. Therefore, the following chapters will mainly deal with the problem (3.28) and not with the entire multiresolution approach.

Chapter 4

Topics in Exact Image Registration

In this chapter, we discuss the registration task in the noise-free case, when the template image $T : V_T \to \mathbb{R}$ is a copy of the reference image $R : V_R \to \mathbb{R}$, deformed only by an unknown affine or Euclidean transformation. That is $T(x) = R(g \cdot x)$ for a $g \in G = SA(n), SE(n)$ and for all $x \in V_T$.

In this case, we need no iterative method to calculate g by given images R and T. In the so-called *principal axes-based registration method*, the pixels are interpreted as mass-points where the mass is equal to the gray-value. Therefore, we only have to compare the center of mass and the principal axes of both images to calculate g. We refer to [10] for an examination of this algorithm. Moreover, an extension of this method to the multimodal registration task is given in [95].

Two matters need to be discussed in the case of noise-free monomodal registration: First, the principal axis registration is very sensitive to the image data. For example, if the template T is a small image section of the reference R, the result of the principle axis method can be meaningless. Second, the principal axis registration encompasses all pixels of an image, whereas the dimension of G is comparatively small. For example, the left picture in Fig. 3.2 has more than $2.7 \cdot 10^5$ pixels, whereas the dimension of the admissible set $SE(2)$ is only 3. In this chapter, we tackle the problem of how much information of the template is necessary to recover the relative position of the template in the reference. More precisely, if the image R is of the form (3.6), how many sample-points $\mathcal{X} = (x_1, \dots, x_m) \in \mathbb{R}^{n\times m}$ are necessary such that the evaluation map

$$\phi_{R,\mathcal{X}} : G \to \mathbb{R}^m, \quad M \mapsto \big(R(PM\bar{x}_1), \dots, R(PM\bar{x}_m)\big)^\top \quad \text{with} \quad P = \begin{pmatrix} I_n & 0 \end{pmatrix} \in \mathbb{R}^{n\times(n+1)} \tag{4.1}$$

becomes an embedding of G in $\mathbb{R}^{n\times m}$? Here, $\bar{x}$ denotes again the homogenous coordinates of the vector $x \in \mathbb{R}^n$.

Of course, the answer to this question depends on the image R. For example, if we have a black image $R \equiv 0$, then (4.1) is not an embedding for no choice of $\mathcal{X} \in \mathbb{R}^{n\times m}$ and one can easily think of more examples, like a white circle with black background. However, we will show that these exceptions form a thin set in the function space $C(\mathbb{R}^n, \mathbb{R})$.

4.1 Preliminaries on Transversality Theory

We start with some preliminaries on transversality theory. For a more detailed discussion we refer to [50]. Let $\mathcal{M}, \mathcal{N}$ be manifolds and $A \subset \mathcal{N}$ a submanifold of $\mathcal{N}$. A differential map $f : \mathcal{M} \to \mathcal{N}$ is *transverse* to A and denoted by $f \pitchfork A$ if

$$T_y A + T_x f(T_x \mathcal{M}) = T_y \mathcal{N} \tag{4.2}$$

for all $x \in \mathcal{M},\ y \in \mathcal{N}$ with $f(x) = y \in A$. That is, the tangent space $T_y\mathcal{N}$ of $\mathcal{N}$ at y is spanned by the tangent space $T_y\mathcal{N}$ of A at y and the image of the tangent space $T_x\mathcal{M}$ of $\mathcal{M}$ at x through the tangent map $T_x f$. Therefore, we can regard the transversality property as an extension of the "regular value" concept and the next theorem as a generalization of the *Submersion Theorem*; its proof can be found e.g. in [50].

Theorem 4.1
Let $f : \mathcal{M} \to \mathcal{N}$ be a C^r map, $r \geqslant 1$ and $A \subset \mathcal{N}$ a C^r submanifold. If f is transverse to A and $f(\mathcal{M}) \cap A \neq \varnothing$, then $f^{-1}(A)$ is a submanifold of $\mathcal{M}$. The codimension of $f^{-1}(A)$ is the same as the codimension of A in $\mathcal{N}$.

For a given subset $L \subset \mathcal{M}$ we use the notation $f \pitchfork_L A$ to express that (4.2) is satisfied for all $x \in L,\ y \in \mathcal{N}$ with $f(x) = y \in A$.
Next, we introduce two topologies for the set $C^r(\mathcal{M}, \mathcal{N})$ of all r-times continuously differentiable functions from $\mathcal{M}$ to $\mathcal{N}$, namely the weak topology and the strong topology. For the so-called *weak topology* $C^r_W(\mathcal{M}, \mathcal{N})$, we define the subbasis as follows: For each $f \in C^r(\mathcal{M}, \mathcal{N})$, for each pair of local charts (φ, U), (ψ, V) on $\mathcal{M}$, $\mathcal{N}$, for each compact set $K \subset U$ with $f(K) \subset V$ and for each $0 < \varepsilon < \infty$, we construct the open subbasic neighborhood

$$\mathcal{U}^r(f; (\varphi, U), (\psi, V), K, \varepsilon) =$$
$$\left\{ g \in C^r(\mathcal{M}, \mathcal{N}) \;\middle|\; g(K) \subset V, \sup_{x \in K, 0 \leqslant k \leqslant r} \|D^k(\psi \circ f \circ \varphi^{-1})(x) - D^k(\psi \circ g \circ \varphi^{-1})(x)\| < \varepsilon \right\}.$$

Then, each open set in $C^r_W(\mathcal{M}, \mathcal{N})$ is composed of arbitrary unions of finite intersections of these sets. In the special case when $\mathcal{M}$ is compact and $\mathcal{N}$ has a metric $d_{\mathcal{N}}$, the topological space $C^r_W(\mathcal{M}, \mathcal{N})$ is metrizable. Moreover, the metric is complete provided that $\mathcal{N}$ is a complete metric space (cf. [50]).
In order to construct an open basic neighborhood with respect to the *strong topology* $C^r_S(\mathcal{M}, \mathcal{N})$ we define $\mathcal{U}^r(f; \Phi, \Psi, K, \varepsilon)$ for each $f \in C^r(\mathcal{M}, \mathcal{N})$, each local finite set of charts $\Phi = \{\varphi_i, U_i\}_{i \in \Lambda}$ on $\mathcal{M}$, each set of charts $\Psi = \{\psi_i, V_i\}_{i \in \Lambda}$ on $\mathcal{N}$, each family $K = \{K_i\}_{i \in \Lambda}$ of compact subsets of $\mathcal{M}$ with $K_i \subset U_i$, and each family of positive numbers $\varepsilon = \{\varepsilon_i\}_{i \in \Lambda}$:

$$\mathcal{U}^r(f; \Phi, \Psi, K, \varepsilon) =$$
$$\left\{ g \in C^r(\mathcal{M}, \mathcal{N}) \;\middle|\; \forall_{i \in \Lambda}\ g(K_i) \subset V_i, \sup_{x \in K_i, 0 \leqslant k \leqslant r} \|D^k(\psi_i \circ f \circ \varphi_i^{-1})(x) - D^k(\psi_i \circ g \circ \varphi_i^{-1})(x)\| < \varepsilon_i \right\}.$$

An open set in $C^r_S(\mathcal{M}, \mathcal{N})$ is composed of arbitrary unions of these sets.

We want to underline the difference between the two topologies with an example: The sequence of functions $f_n \in C(\mathbb{R}, \mathbb{R})$, $f_n(x) = \frac{1}{n}x^2$, $n \in \mathbb{N}$ converges to $f \equiv 0$ with respect to the weak topology, but it does not converge with respect to the strong topology. In other words, the weak (or "compact-open") topology does not control the limiting behavior of the functions.

Another advantage of the strong topology is that it satisfies the Baire property. (A proof of this and a more detailed comparison of both topologies can be found in [50].) Yet, both topologies are identical if M is compact.

The following proposition may be seen as an analog version of Theorem 4.1 with focus on the function space. Another proof may be found in [96] p.52.

Proposition 4.2

Let $f : \mathcal{M} \to \mathcal{N}$ be a C^r map, $r \geqslant 1$ and $A \subset \mathcal{N}$ a C^r submanifold. If f is transverse to A and $f(\mathcal{M}) \cap A \neq \varnothing$, then there is an open neighborhood $U \subset C^r_S(\mathcal{M}, \mathcal{N})$ of f with $g(\mathcal{M}) \cap A \neq \varnothing$ for all $g \in U$.

Proof

Denote with m, n and q the dimensions of $\mathcal{M}$, $\mathcal{N}$ and A, respectively. Let $x_0 \in \mathcal{M}$ such that $f(x_0) \in A$. Since A is a submanifold of $\mathcal{N}$, there is a chart $\psi : \tilde{W} \subset \mathcal{N} \to W \subset \mathbb{R}^n$ of $\mathcal{N}$ in $f(x_0)$ and a q-dimensional subspace $V \subset \mathbb{R}^n$ such that $\psi^{-1}(V \cap W) = A \cap \tilde{W}$. Let $\pi : \mathbb{R}^n \to \mathbb{R}^n$ denote the projection from $\mathbb{R}^n$ onto the orthogonal complement X of V. Moreover, let $\varphi : \tilde{Y} \subset \mathcal{M} \to Y \subset \mathbb{R}^m$ be a chart of $\mathcal{M}$ in x_0 with $f(\tilde{Y}) \subset \tilde{W}$. Since f is transverse to A, $\varphi(x_0)$ is a regular point of $\tilde{f} := \pi \circ \psi \circ f \circ \varphi^{-1}$. Let $K \subset Y$ denote a compact subset of Y with an interior point $\varphi(x_0) \in K$.

Consider the map $\mathfrak{F} : C^r(K, X) \times K \to X$, $(g, x) \mapsto g(x)$. We can equip $C^r(K, X)$ with the supremum norm

$$\|g\|_{C^r(K,X)} := \sup \left\{ \|D^k g(x)\| \;\middle|\; x \in K,\ k = 0, \ldots, r \right\},$$

with respect to which the set $C^r(K, X) \times K$ is a Banach space. Moreover, with respect to this norm, the map $\mathfrak{F}$ is continuously differentiable with

$$D\mathfrak{F}(g, x)(\varepsilon, h) = Dg(x)h + \varepsilon(x)$$

and the second component $h \mapsto D\mathfrak{F}(\tilde{f}, x_0)h$ is surjective. Thus, the *Implicit Function Theorem* yields an open neighborhood $\tilde{U} \subset C^r(K, X)$ of $\tilde{f}$ and a map $\mathfrak{H} : C^r(K, X) \to X$ with $\mathfrak{F}(g, \mathfrak{H}(g)) = 0$ for all $g \in \tilde{U}$.

Due to the continuity of the map $\mathfrak{G} : C^r_S(\mathcal{M}, \mathcal{N}) \to C^r_S(K, X)$ $g \mapsto \pi \circ \psi \circ g \circ \varphi^{-1}|_K$, the set $U := \mathfrak{G}^{-1}(\tilde{U})$ is an open neighborhood of f. Moreover, we get $g(\mathcal{M}) \cap A \neq \varnothing$ for all $g \in U$. □

The following theorem is a well known result from the transversality theory and its proof can be found e.g. in [50].

Theorem 4.3 (Transversality Theorem)

Let $\mathcal{M}$, $\mathcal{N}$ be manifolds and $A \subset \mathcal{N}$ a submanifold. Let $1 \leqslant r \leqslant \infty$. Then:

(a) *$\pitchfork^r(\mathcal{M},\mathcal{N};A) := \{f \in C^r(\mathcal{M},\mathcal{N}) \mid f \pitchfork A\}$ is residual in $C^r(\mathcal{M},\mathcal{N})$ for both the strong and the weak topologies.*

(b) *Suppose A is closed in $\mathcal{N}$. If $L \subset \mathcal{M}$ is closed [resp. compact] then $\pitchfork^r_L(\mathcal{M},\mathcal{N};A) := \{f \in C^r(\mathcal{M},\mathcal{N}) \mid f \pitchfork_L A\}$ is dense and open in $C^r_S(\mathcal{M},\mathcal{N})$ [resp. $C^r_W(\mathcal{M},\mathcal{N})$].*

From the *Transversality Theorem* we can formulate the following result for mappings between function spaces. Some of the arguments can be found in the *Globalization Theorem* (see e.g. [50]).

Lemma 4.4
Let $\mathcal{M}_1$, $\mathcal{M}_2$, $\mathcal{N}_1$, $\mathcal{N}_2$ be C^∞ Riemannian manifolds, $\mathcal{M}_2$ connected and let $A \subset \mathcal{N}_2$ be a submanifold. Moreover, let the map $\mathfrak{F} : C^r_S(\mathcal{M}_1,\mathcal{N}_1) \to C^s_S(\mathcal{M}_2,\mathcal{N}_2)$ be continuous. Suppose that for each compact set $K \subset \mathcal{M}_2$, there is a nonempty, compact set $L \subset \mathcal{M}_1$ such that for all $f, g \in C^r(\mathcal{M}_1,\mathcal{N}_1)$ the implication

$$f|_L \equiv g|_L \quad \Rightarrow \quad \mathfrak{F}(f)|_K \equiv \mathfrak{F}(g)|_K \tag{4.3}$$

holds. If the set

$$\pitchfork^r(\mathcal{M}_1,\mathcal{N}_1;\mathfrak{F},A) := \big\{f \in C^r(\mathcal{M}_1,\mathcal{N}_1) \;\big|\; \mathfrak{F}(f) \pitchfork A\big\}$$

is dense in $C^r(\mathcal{M}_1,\mathcal{N}_1)$ with respect to the weak topology, then $\pitchfork^r(\mathcal{M}_1,\mathcal{N}_1;\mathfrak{F},A)$ is also generic in $C^r(\mathcal{M}_1,\mathcal{N}_1)$ with respect to the strong topology.

Proof
Let $d_{\mathcal{M}_2}$ denote the Riemannian metric in $\mathcal{M}_2$ and let $y \in \mathcal{M}_2$ be arbitrary. We denote by $B_k(y) = \{z \in \mathcal{M}_2 \mid d_{\mathcal{M}_2}(z,y) \leq k\}$ the closed ball in $\mathcal{M}_2$ around y with radius $k > 0$. For each $k \in \mathbb{N}$, we define the set

$$\pitchfork^r_k(\mathcal{M}_1,\mathcal{N}_1;\mathfrak{F},A) := \Big\{f \in C^r(\mathcal{M}_1,\mathcal{N}_1) \;\big|\; \mathfrak{F}(f)\big|_{B_k(y)} \pitchfork A\Big\}.$$

Then, property (4.3) states that there is a sequence of compact sets $L_k \subset \mathcal{M}_1$ that satisfies

$$f|_{L_k} \equiv g|_{L_k} \quad \Rightarrow \quad \mathfrak{F}(f)|_{B_k(y)} \equiv \mathfrak{F}(g)|_{B_k(y)}$$

for all $f, g \in C^r(\mathcal{M}_1,\mathcal{N}_1)$. Moreover, for each $k \in \mathbb{N}$, there is a compact set $K_k \subset \mathcal{M}_1$ with the property that L_k lies in the interior of K_k, i.e. $L_k \subset \operatorname{int}(K_k)$.
Now, let $f \in C^r(\mathcal{M}_1,\mathcal{N}_1)$ be arbitrary. Since $\pitchfork^r_k(\mathcal{M}_1,\mathcal{N}_1;\mathfrak{F},A)$ is dense in C^r_W, there is a sequence $\{f_l\}_{l\in\mathbb{N}} \subset \pitchfork^r_k(\mathcal{M}_1,\mathcal{N}_1;\mathfrak{F},A)$ which converges to f with respect to the weak topology. If $\mathcal{M}_1$ is compact, the sequence $\{f_l\}_{l\in\mathbb{N}}$ converges also with respect to the strong topology, which implies that $\pitchfork_k(\mathcal{M}_1,\mathcal{N}_1;\mathfrak{F},A)$ is dense in C^r_S.
In the case that $\mathcal{M}_1$ is not compact, we construct another sequence of functions $\{h_l\}_{l\in\mathbb{N}} \subset C^r(\mathcal{M}_1,\mathcal{N}_1)$:

$$h_l(x) := \begin{cases} \exp_{f(x)}\Big(\lambda(x)\exp^{-1}_{f(x)} f_l(x)\Big) & \text{if } x \in K_k \\ f(x) & \text{if } x \in \mathcal{M}_1 \setminus K_k. \end{cases} \tag{4.4}$$

where $\lambda : K_k \to [0,1]$ is a C^r map with compact support, such that $\lambda(x) = 1$ for all x near L_k. Here, $\exp_x : T_x\mathcal{M}_2 \to \mathcal{M}_2$ denotes the Riemannian exponential map. Regarding (4.4), we have to mention that the injectivity radius of the Riemannian exponential map $\exp_p$ is a continuous function in $p \in \mathcal{M}_2$ (cf. [97] p.131). Since K_k is compact, there exists a global injectivity radius of $\exp_{f(x)}$ for all $x \in K_k$. Thus, using the fact that f_l converges to f, we conclude that h_l is well-defined if $l \in \mathbb{N}$ is large enough.

Since f_l converges to f with respect to the weak topology, h_l converges to f with respect to the strong topology. Moreover, we have $h_l|_{L_k} \equiv f_l|_{L_k}$ and therefore $\{h_l\}_{l\in\mathbb{N}} \subset\pitchfork^r_k (\mathcal{M}_1,\mathcal{N}_1;\mathfrak{F},A)$. Since f was given arbitrarily, it follows that $\pitchfork_k (\mathcal{M}_1,\mathcal{N}_1;\mathfrak{F},A)$ is dense in C^r_S.

The *Transversality Theorem* states that the set $\pitchfork^s (\mathcal{M}_2,\mathcal{N}_2,A\cap B_k(\tilde{y}))$ is residual with respect to the strong topology. That is, $\pitchfork^s (\mathcal{M}_2,\mathcal{N}_2,A\cap B_k(\tilde{y}))$ is a countable intersection of open and dense sets in $C^s_S(\mathcal{M}_2,\mathcal{N}_2)$. Since $\mathfrak{F}$ is continuous and

$$\pitchfork^r_k (\mathcal{M}_1,\mathcal{N}_1;\mathfrak{F},A) = \mathfrak{F}^{-1}\big(\pitchfork^s (\mathcal{M}_2,\mathcal{N}_2,A\cap B_k(y))\big)$$

we get that $\pitchfork^r_k (\mathcal{M}_1,\mathcal{N}_1;\mathfrak{F},A)$ is also a countable intersection of open sets with respect to the strong topology. Thus, $\pitchfork^r_k (\mathcal{M}_1,\mathcal{N}_1;\mathfrak{F},A)$ is generic for each $k \in \mathbb{N}$. Since $\mathcal{M}_2$ is connected, we have

$$\pitchfork^r (\mathcal{M}_1,\mathcal{N}_1;\mathfrak{F},A) = \bigcap_{k\in\mathbb{N}} \pitchfork^r_k (\mathcal{M}_1,\mathcal{N}_1;\mathfrak{F},A).$$

Thus, using the fact that $C^r_S(\mathcal{M},\mathcal{N})$ is a Baire space, we conclude that the set $\pitchfork^r (\mathcal{M}_1,\mathcal{N}_1;\mathfrak{F},A)$ is generic as well. □

The following theorem can be seen as an extension of the Morse-Sard Theorem into the field of transversality theory, its proof can be found e.g. in [50].

Theorem 4.5 (Parametric Transversality)

Let $\mathcal{V},\mathcal{M},\mathcal{N}$ be C^r manifolds without boundary and $A \subset \mathcal{N}$ be a C^r submanifold. Let $F : \mathcal{V} \to C^r(\mathcal{M},\mathcal{N})$ satisfy the following conditions:

(a) the evaluation map $F^{\mathrm{ev}} : \mathcal{V}\times\mathcal{M} \to \mathcal{N}, (v,x) \mapsto F_v(x)$ is C^r.

(b) F^{ev} is transverse to A.

(c) $r > max\{0, dim\ \mathcal{M} + dim\ A - dim\ \mathcal{N}\}$.

Then the set

$$\pitchfork (F;A) := \{v \in \mathcal{V} \mid F_v \pitchfork A\}$$

is residual and therefore dense. If A is closed in $\mathcal{N}$ and F is continuous for the strong topology on $C^r(\mathcal{M},\mathcal{N})$, then $\pitchfork (F,A)$ is also open.

We conclude this section by applying of the *Parametric Transversality Theorem* to the so-called *jet-manifolds*: Again, let $\mathcal{M}$ and $\mathcal{N}$ be differentiable manifolds. Let m and n denote the dimensions of $\mathcal{M}$ and $\mathcal{N}$, respectively. In order to define the elements of the jet-manifold from $\mathcal{M}$ to $\mathcal{N}$ we consider triples of the form $[x,f,U]_r$, where $U \subset \mathcal{M}$ is an open subset, $x \in \mathcal{M}$ and $f : U \to \mathcal{N}$ is an C^r map. We say that two triples $[x,f,U]_r$ and $[x',f',U']_r$ are equivalent if

(i) $x = x'$,

(ii) $f(x) = f'(x)$ and

(iii) for some charts $\varphi : V \to \mathbb{R}^m$, $\psi : W \to \mathbb{R}^n$ around $x \in V \subset \mathcal{M}$ and $f(x) \in W \subset N$, (V and W open,) the functions $\psi \circ f \circ \varphi^{-1}$ and $\psi \circ f' \circ \varphi^{-1}$ have the same derivatives in x up to the order $r \in \mathbb{N}_0$.

We call the equivalence-class of $[x, f, U]_r$ the *r-jets* and we denote it as $j^r_x f$, i.e.

$$j^r_x f := [x, f, U]_r. \tag{4.5}$$

The set of all r-jets from $\mathcal{M}$ to $\mathcal{N}$ is denoted by $J^r(\mathcal{M}, \mathcal{N})$ and

$$j^r f : \mathcal{M} \to J^r(\mathcal{M}, \mathcal{N}), \quad x \mapsto j^r_x f$$

defines the so-called *r-prolongation map.*
Moreover, we can equip $J^r(\mathcal{M}, \mathcal{N})$ with a manifold structure. If (φ, U) and (ψ, V) are charts of $\mathcal{M}$ and $\mathcal{N}$, respectively, we construct a map $\theta : J^r(U, V) \to J^r(\varphi(U), \psi(V))$ by

$$\theta : j^r_x f \mapsto j^r_y(\psi \circ f \circ \varphi^{-1}), \quad y = f(x). \tag{4.6}$$

Since $\varphi(U)$ and $\psi(V)$ are open sets in a m-dimensional vector space and in an m-dimensional vector space, respectively, we obtain that

$$J^r(\varphi(U), \psi(V)) \cong \varphi(U) \times \psi(V) \times \prod_{k=1}^{r} \mathbb{L}^k_{\text{sym}}(\mathbb{R}^m, \mathbb{R}^n), \tag{4.7}$$

where $\mathbb{L}^k_{\text{sym}}(\mathbb{R}^m, \mathbb{R}^n)$ denotes the vector space of symmetric, k-linear maps from $\mathbb{R}^m$ to $\mathbb{R}^n$. Therefore, $J^r(\varphi(U), \psi(V))$ is an open set of a vector space and one can easily show that θ defines a chart of $J^r(\mathcal{M}, \mathcal{N})$. Hence, $J^r(\mathcal{M}, \mathcal{N})$ is a manifold and from (4.7) we obtain that it has the dimension

$$\dim J^r(\mathcal{M}, \mathcal{N}) = \dim(\mathcal{M}) + \dim(\mathcal{N}) \cdot \sum_{k=0}^{r} \binom{\dim(\mathcal{M}) + k - 1}{k}. \tag{4.8}$$

The following theorem is an application of the previous *Parametric Transversality Theorem* to the jet manifold. Its proof can be found in [50].

Theorem 4.6 (Jet Transversality Theorem)

Let $\mathcal{M}$, $\mathcal{N}$ be C^∞ manifolds without boundary, and let $A \subset J^r(\mathcal{M}, \mathcal{N})$ be a C^∞ submanifold. Suppose $1 \leqslant r < s \leqslant \infty$. Then

$$\pitchfork^s (\mathcal{M}, \mathcal{N}; j^r, A) := \left\{ f \in C^s(\mathcal{M}, \mathcal{N}) \;\middle|\; j^r f \pitchfork A \right\}$$

is residual and thus dense in $C^s_S(\mathcal{M}, \mathcal{N})$ and open if A is closed.

4.2 Image Appearance Manifolds with Finite Evaluations

In contrast to our approach, where images are differentiable functions of the form (3.6) or of the form (3.8), Baraniuk et al. [48, 98] considered images as Lebesgue integrable functions $R \in L^2(\mathbb{R}^n, \mathbb{R})$. For a given Lie group G acting smoothly on $\mathbb{R}^n$, they studied the G-orbit of R, i.e. the set of all images occurring by a transformation of R

$$R^G := \{R \circ \rho_g \mid g \in G\} \subset L^2(\mathbb{R}^n, \mathbb{R}), \tag{4.9}$$

with $\rho_g : \mathbb{R}^n \to \mathbb{R}^n,\ x \mapsto g \cdot x$. The set R^G is called “Image Appearance Manifold” (IAM), even though there are no known conditions in the literature, such that R^G is a manifold (not even for the simplest case when G denotes the set of all translations; cf [48]).
In [48, 98] it was demonstrated that R^G is in general not differentiable with respect to the L^2-norm. In the given examples, discontinuous functions R yielded such a lack of differentiability. For instance, let $\chi_{B_1(0)}$ denote the characteristic function of a ball $B_1(0) \subset \mathbb{R}^2$ around zero with radius one and let G denote the set of translations in $\mathbb{R}^2$. Then $R(x) = \chi_{B_1(0)}(x)$ is a function representation of an image consisting of a white circle on a black background and R^G is not differentiable. Such discontinuous functions R appear quite naturally if there is no interpolation process applied to the images, as it was presented in Section 3.2.
The missing differentiable structure can be interpreted as another reason for a multiscale approach for the registration problem like it is presented in (3.28). Roughly speaking, to construct and examine image registration algorithms, we are forced to use smooth image representations of the form (3.6).
In this section, we study IAMs of the form R^G with $R \in C^r(\mathbb{R}^n, \mathbb{R})$, $r \geq 1$. In many applications we will not use R^G itself, but a sampled version of it. In order to model this sampled version, we use here the map eval : $C(\mathbb{R}^n, \mathbb{R}) \times \mathbb{R}^{n \times m} \to \mathbb{R}^m$ which evaluates a function f simultaneously in m sample-points

$$\operatorname{eval}(f; \mathcal{X}) := \operatorname{eval}(f; x_1, \ldots, x_m) := (f(x_1), \ldots, f(x_m))^\top. \tag{4.10}$$

For a given image representation R of the form (3.6) and a given tuple of sample-points $\mathcal{X} = (x_1, \ldots, x_m) \in \mathbb{R}^{n \times m}$ we will examine the evaluation of $R \circ \rho_g$ in $\mathcal{X}$ by varying the transformation $g \in G$, i.e. we study the function

$$\phi_{R,\mathcal{X}} : G \to \mathbb{R}^m, \quad \phi_{R,\mathcal{X}}(g) = \operatorname{eval}(R \circ \rho_g, \mathcal{X}). \tag{4.11}$$

Using the notation

$$\cdot\ : G \times \mathbb{R}^{n \times m} \to \mathbb{R}^{n \times m}, \quad g \cdot (x_1, \ldots, x_m) := (g \cdot x_1, \ldots, g \cdot x_m) \tag{4.12}$$

for the simultaneous group action of G on the set of m sample-points, we get an equivalent definition of $\phi_{R,\mathcal{X}}$ by

$$\phi_{R,\mathcal{X}}(g) = \operatorname{eval}(R, g \cdot \mathcal{X}). \tag{4.13}$$

We will see that under mild conditions on $\mathcal{X}$ and for a generic choice of R, the set $\phi_{R,\mathcal{X}}(G)$ is a manifold. Since $\phi_{R,\mathcal{X}}$ is a composition of the evaluation map eval and the simultaneous group action, we study the map (4.12) in the first subsection. In Subsection 4.2.2 we examine the function $\operatorname{eval}\big|_{G \cdot \mathcal{X}}$.

4.2.1 Properties of Special Group Actions

In this subsection we examine the simultaneous group action of the special affine group and its subgroup, the special Euclidean group. Using the $SA(n)$ group action, defined in (2.9), the simultaneous group action (4.12) becomes

$$\cdot : SA(n) \times \mathbb{R}^{n\times m} \to \mathbb{R}^{n\times m}, \quad M \cdot (x_1, \dots, x_m) := (PM\bar{x}_1, \dots, PM\bar{x}_m),$$

with $P = (I_n\ 0) \in \mathbb{R}^{n\times(n+1)}$. Here, $x_1, \dots, x_m \in \mathbb{R}^n$ are given sample-points, and $\bar{x}_1, \dots, \bar{x}_m \in \mathbb{R}^{n+1}$ denotes their homogenous coordinates. Again, we write $\mathcal{X} = (x_1, \dots, x_m) \in \mathbb{R}^{n\times m}$, and define the orbit of $\mathcal{X}$ with respect to $SA(n)$ by

$$SA(n) \cdot \mathcal{X} := \alpha_{\mathcal{X}}(SA(n))$$

with

$$\alpha_{\mathcal{X}} : SA(n) \to \mathbb{R}^{n\times m}, \quad \alpha_{\mathcal{X}}(M) := M \cdot \mathcal{X}. \tag{4.14}$$

The next theorem states that, for $m \geq n$ and under some generic conditions for the sample-points $\mathcal{X}$, the map $\alpha_{\mathcal{X}} : SA(n) \to \mathbb{R}^{n\times m}$ is an embedding. So the image of $\alpha_{\mathcal{X}}$ forms a submanifold, i.e. $SA(n) \cong SA(n) \cdot \mathcal{X}$.

Theorem 4.7
Let the sample-points $(x_1, \dots, x_m) =: \mathcal{X} \in \mathbb{R}^{n\times m}$ be given and let $H \subset \mathbb{R}^n$ denote the smallest affine subspace containing $x_1, \dots, x_m$. The map $\alpha_{\mathcal{X}}$ is a closed embedding if and only if $H = \mathbb{R}^n$.

Recall that a function $f : \mathcal{M} \to \mathcal{N}$ between two metric spaces $\mathcal{M}, \mathcal{N}$ is called proper, if the preimage $f^{-1}(K)$ of each compact subset $K \subset \mathcal{N}$ is compact in $\mathcal{M}$. The function f is called embedding if it is an homeomorphism and an immersion. If it is, additionally, proper, we call f an closed embedding.

Proof
First, assume that the codimension of H is one or greater. Let $(b_1, \dots, b_n)$ be an orthonormal basis of $\mathbb{R}^n$ such that $H \subseteq \langle b_2, \dots, b_n\rangle$. Therefore, we get $(I + b_2 b_1^\top)x = x$ for all $x \in H$. Hence, $\alpha_{\mathcal{X}}$ is not injective, since $\alpha_{\mathcal{X}}(I, 0) = \alpha_{\mathcal{X}}(I + b_2 b_1^\top, 0)$. (Here, we use the representation $SA(n) \cong SL(n) \ltimes \mathbb{R}^n$.)
Now, assume that $H = \mathbb{R}^n$, which in particular implies $m \geq n$. We will construct the inverse map of $\alpha_{\mathcal{X}}$ and show it's continuity. Therefore, we have to solve the equation $(y_1, \dots, y_m) = \alpha_{\mathcal{X}}(A, t)$ with respect to A and t. Since the map $(A, t) \to (A, t + \tau)$ is a diffeomorphism for each $\tau \in \mathbb{R}^n$, we can w.l.o.g. assume that the mean $x^\star := \frac{1}{m}\sum_{i=1}^m x_i$ of $\mathcal{X}$ is 0 and that the first n vectors $\{x_1, \dots, x_n\}$ are linearly independent. Then, the translation t is given by the formula

$$t = A\left(\frac{1}{m}\sum_{i=1}^m x_i\right) + t = \frac{1}{m}\sum_{i=1}^m y_i \tag{4.15}$$

and the distortion A is given by

$$A = (y_1 - t, \dots, y_n - t)\,(x_1, \dots, x_n)^{-1}\,. \tag{4.16}$$

This can be interpreted as a linear, bijective map $\mathbb{R}^{n\times m} \to gl(n) \times \mathbb{R}^n$, $(y_1, \dots, y_m) \mapsto (A, t)$. Therefore, the inverse map $\alpha_X^{-1} : \alpha_X(SA(n)) \to SA(n)$ is continuous with respect to the standard topology on $\mathbb{R}^{n\times m}$.
In order to show that the orbit $SA(n) \cdot \mathcal{X}$ is closed, let $(\mathcal{Y}_k)_{k\in\mathbb{N}} \subset SA(n) \cdot \mathcal{X}$ denote an arbitrary sequence, which converge to $\mathcal{Y} \in \mathbb{R}^{n\times m}$. Moreover, let $(A_k, t_k) \in SA(n)$ denote the corresponding group element, i.e. $(A_k, t_k) \cdot \mathcal{X} = \mathcal{Y}_k$. Thus, (4.15) and (4.16) yields that (A_k, t_k) converges, which implies that $\alpha_{\mathcal{X}}$ is closed. □

Since the Special Euclidean Group is a sub-Lie Group of the Special Affine Group, we can restrict the action of $SA(n)$ on $\mathbb{R}^n$ defined in (2.9) to an action of $SE(n)$ on $\mathbb{R}^n$. In the same way, we can restrict the function $\alpha_{\mathcal{X}}$ defined in (4.14) on $SE(n)$ to examine the simultaneous action of the Lie group on a set of sample-points $\mathcal{X} = (x_1, \dots, x_m) \subset \mathbb{R}^{n\times m}$. The following theorem is the analogue of Theorem 4.7 for the Special Euclidian Group.

Theorem 4.8

Let the sample-points $(x_1, \dots, x_m) =: \mathcal{X} \in \mathbb{R}^{n\times m}$ be given and let $H \subset \mathbb{R}^n$ denote the smallest affine subspace containing $x_1, \dots, x_m$. The map $\alpha_{\mathcal{X}}|_{SE(n)}$ is a closed embedding if and only if the codimension of H is smaller or equal to 1.

Proof

Firstly assume that the sample-points lie in an affine subspace $H \subset \mathbb{R}^d$ with codimension 2 or greater. Then, each rotation around this subspace H is invariant to all sample-points. Hence $\alpha_{\mathcal{X}}|_{SE(n)}$ is not injective.
Next, assume that the subspace $H \subset \mathbb{R}^d$ has codimension 1 or less. Then without loss of generality we can assume that $x_1 = 0$ and that the vectors $\{x_2, \dots, x_n\}$ are linearly independent. We construct another sample-point x^* by the cross product in $\mathbb{R}^n$

$$x^* = x_2 \times \ldots \times x_n.$$

In order to show that $\alpha_{\mathcal{X}}|_{SE(n)}$ is an embedding, it is sufficient to construct the inverse map and to show its continuity. Therefore, we have to solve the equation $(y_1, \dots, y_m) = \alpha_{\mathcal{X}}(A, t)$ with respect to A and t. Using

$$y^* := (y_2 - y_1) \times \ldots \times (y_n - y_1) + y_1 \tag{4.17}$$

we get

$$A = (y_2 - y_1, \dots, y_n - y_1, y^* - y_1)(x_2, \dots, x_n, x^*)^{-1}. \tag{4.18}$$

and

$$t = y_1 \tag{4.19}$$

Since $\{x_2, \ldots, x_n\}$ are linearly independent the matrix $(x_2, \ldots, x_n, x^*)$ is invertible. Each of the three formulas, (4.17), (4.19) and (4.18), can be interpreted as linear maps in the variables $y_1, \ldots, y_m$. Hence, the inverse map $\alpha_{\mathcal{X}}|_{SE(n)}^{-1} : \alpha_{\mathcal{X}}(SE(n)) \to SE(n)$ is continuous with respect to the standard topology on $\mathbb{R}^{n\times m}$. Using the same argumentation as in Theorem 4.7, we can show that $\alpha_{\mathcal{X}}|_{SE(n)}$ is also a closed embedding. □

4.2.2 Generic Properties for Fixed Evaluation Points

Recall that the aim of this section is to find conditions for which the map

$$\phi_{f,\mathcal{X}} : G \to \mathbb{R}^m, \quad g \mapsto \mathrm{eval}(f; g \cdot \mathcal{X})$$

is a closed embedding. Here, $f : \mathbb{R}^n \to \mathbb{R}$ is a given and sufficiently smooth function, G is given Lie group and the evaluation map eval was defined in (4.10). In the previous subsection we already mentioned the necessary condition that the function $\alpha_{\mathcal{X}} : G \to G \cdot \mathcal{X} \subset \mathbb{R}^{n\times m}$ defined in (4.12) has to be a closed embedding. This condition was studied in Theorems 4.7 and 4.8 for the Lie groups $SE(n)$ and $SA(n)$, respectively. Throughout this subsection, we assume that this condition is satisfied, which also implies that the G-orbit $G \cdot \mathcal{X} \subset \mathbb{R}^{n\times m}$ of $\mathcal{X}$ is a submanifold with the same dimension as G.

Theorem 4.9
Let G be a finite-dimensional Lie group acting smoothly on $\mathbb{R}^n$ and let $\mathcal{X} \in \mathbb{R}^{n\times m}$ be a m-tuple of distinct sample-points in $\mathbb{R}^n$ such that $G \to G \cdot \mathcal{X}$, $g \mapsto g \cdot \mathcal{X}$ is an embedding of G in $\mathbb{R}^{n\times m}$. If $m \geq 2 \dim G$, then the set

$$\left\{ f \in C^r(\mathbb{R}^n, \mathbb{R}) \;\middle|\; \mathit{eval}(f;\ .\ )|_{G\cdot\mathcal{X}} : G \cdot \mathcal{X} \to \mathbb{R}^m \text{ is an immersion} \right\}, \quad r \geq 2$$

is generic in $C^r_S(\mathbb{R}^n, \mathbb{R})$.

Before starting the proof, we make some useful notations and remarks. We denote the set of real-valued polynomials in n variables with degree l or less of $\mathcal{P}_{n,l}$. For two given manifolds $\mathcal{M}, \mathcal{N}$ we define the set $\mathcal{A}_s \subset J^1(\mathcal{M}, \mathcal{N})$ consisting of all 1-jets with the property that their derivative part has rank $s \in \mathbb{N}$. According to [96] p.61, $\mathcal{A}_s$ is a submanifold of $J^1(\mathcal{M}, \mathcal{N})$ with

$$\dim \mathcal{A}_s = \dim J^1(\mathcal{M}, \mathcal{N}) - (\dim \mathcal{M} - s)(\dim \mathcal{N} - s)$$

for $s \leq \min\{\dim \mathcal{M}, \dim \mathcal{N}\}$. Hence, assuming that $\dim \mathcal{N} \geq \dim \mathcal{M}$ we can determine an estimate for the dimension of $\mathcal{A}_s$ when $s \leq \dim \mathcal{M} - 1$ as

$$\begin{aligned} \dim \mathcal{A}_s \leq \dim \mathcal{A}_{\dim \mathcal{M}-1} &= \dim \mathcal{M} + \dim \mathcal{N} + \dim \mathcal{M} \dim \mathcal{N} - (\dim \mathcal{N} - \dim \mathcal{M} + 1) \\ &= 2 \dim \mathcal{M} + \dim \mathcal{M} \dim \mathcal{N} - 1. \end{aligned} \tag{4.20}$$

Finally, $\mathrm{pr}^1_{x,y} : J^1(\mathcal{M}, \mathcal{N}) \to \mathbb{L}(T_x\mathcal{M}, T_y\mathcal{N})$ denotes the current "projection" to the derivative-part of the jet space. In the following proof, we will use $\mathcal{A}_s$ and $\mathrm{pr}^1_{x,y}$ for the case $\mathcal{M} = G \cdot \mathcal{X}$ and $\mathcal{N} = \mathbb{R}^m$.

Proof
Let x_i, $1 \leqslant i \leqslant m$ denote the elements of $\mathcal{X}$. Then, there are polynomials $h_{k,b,g}$ in $\mathcal{P}_{n,2m-1}$, for each $1 \leqslant k \leqslant m$, $b \in \mathbb{R}^n$ and $g \in G$, which satisfy

$$\nabla h_{k,b,g}(g \cdot x_i) = \begin{cases} 0 & , \quad k \neq i \\ b & , \quad k = i \end{cases}$$

for all $1 \leqslant i \leqslant m$. For instance, we can set

$$h_{k,b,g}(x) := \frac{b^\top (x - g \cdot x_k) \prod_{i \neq k} \|x - g \cdot x_i\|^2}{\prod_{i \neq k} \|x_k - g \cdot x_i\|^2}.$$

In the following, we will use the abbreviations

$$\mathfrak{F} : C^r(\mathbb{R}^n, \mathbb{R}) \to C^{r-1}(G \cdot \mathcal{X}, J^1(G \cdot \mathcal{X}, \mathbb{R}^m)), \quad f \mapsto j^1 \mathrm{eval}(f, \ . \ )|_{G \cdot \mathcal{X}},$$
$$\mathfrak{F}^{\mathrm{ev}} : C^r(\mathbb{R}^n, \mathbb{R}) \times G \cdot \mathcal{X} \to J^1(G \cdot \mathcal{X}, \mathbb{R}^m), \quad (f, Y) \mapsto j^1_Y \mathrm{eval}(f, \ . \ )|_{G \cdot \mathcal{X}}$$

and

$$\varphi : C^r(\mathbb{R}^n, \mathbb{R}) \times \mathbb{R}^{n \times m} \to \underbrace{J^1(\mathbb{R}^n, \mathbb{R}) \times \ldots \times J^1(\mathbb{R}^n, \mathbb{R})}_{m-\text{times}}, \quad (\tilde{f}; y_1, \ldots, y_m) \mapsto (j^1_{y_1} \tilde{f}, \ldots, j^1_{y_m} \tilde{f})^\top.$$

In a first step, we take an arbitrary $f \in C^r(\mathbb{R}^n, \mathbb{R})$ and define the affine subset $W := f + \mathcal{P}_{n,2m-1}$. Since $\mathcal{P}_{n,2m-1}$ is a finite-dimensional vector space, we can equip W with the Euclidean norm. In the following, we want to apply the *Parametric Transversality Theorem* to the function $\mathfrak{F}^{\mathrm{ev}}|_{W \times G \cdot \mathcal{X}}$.
In order to verify the assumptions of this theorem, we consider the derivative-part of the i^{th} component of φ. It satisfies

$$(\tilde{f} + h_{1,b_1,g} + \ldots + h_{m,b_m,g}; g \cdot x_1, \ldots, g \cdot x_m) \mapsto \nabla \tilde{f}(g \cdot x_i) + b_i \tag{4.21}$$

for each $g \in G$ and each $b_i \in \mathbb{R}^n$. By varying $b_i \in \mathbb{R}^n$, we get that (4.21) is surjective. Hence, the derivative-part of $\varphi\big|_{W \times \mathbb{R}^{n \times m}}(\ . \ , Y)$ is surjective and affine for each $Y \in G \cdot \mathcal{X}$.
Let $F : \mathbb{R}^{n \times m} \to \mathbb{R}^m$ be an arbitrary differentiable function. Since $G \cdot \mathcal{X}$ is a submanifold of $\mathbb{R}^{n \times m}$, we can calculate the tangential-map $T_Y(F\big|_{G \cdot \mathcal{X}}) : T_Y G \cdot \mathcal{Y} \to \mathbb{R}^m$ using $T_Y(F\big|_{G \cdot \mathcal{X}}) = (T_Y F)\big|_{T_Y G \cdot \mathcal{X}}$. In particular, we can use this formula for $F = \tilde{f} \times \ldots \times \tilde{f}$, $(y_1, \ldots, y_m) \mapsto (\tilde{f}(y_1), \ldots, \tilde{f}(y_m))$ and get

$$\mathrm{pr}^1_{Y, \mathrm{eval}(\tilde{f}, Y)}(\mathfrak{F}(\tilde{f})(Y)) = \mathrm{pr}^1_{Y, \mathrm{eval}(\tilde{f}, Y)}(\varphi(\tilde{f}, Y))\big|_{T_Y G \cdot \mathcal{X}}.$$

Therefore, we also get that the derivative-part of $\mathfrak{F}^{\mathrm{ev}}(\tilde{f}, Y) \in J^1(G \cdot \mathcal{X}, \mathbb{R}^m)$ is surjective and affine by varying $\tilde{f} \in W$ for each $Y \in G \cdot \mathcal{X}$. Thus, $\mathfrak{F}^{\mathrm{ev}}$ is transversal to each submanifold $\mathcal{A}_s \subset J^1(G \cdot \mathcal{X}, \mathbb{R}^m)$.
Finally, we verify part (c) of the *Parametric Transversality Theorem* in order to apply the theorem. For $s \leqslant m - 1$, equation (4.20) yields

$$\dim \mathcal{A}_s \leqslant 2 \dim G + m \dim G - 1.$$

Thus

$$\dim \mathcal{A}_s + \dim G \cdot \mathcal{X} \leqslant (\dim G + m + m \dim G) + 2 \dim G - m - 1. \tag{4.22}$$

With equation (4.8) and the condition $m > 2 \dim G - 1$, this leads to

$$\dim \mathcal{A}_s + \dim G \cdot \mathcal{X} - \dim J^1(G \cdot \mathcal{X}, \mathbb{R}^m) < 0.$$

Therefore, since $\mathfrak{F}^{\text{ev}}$ is a C^1 map, we can apply the *Parametric Transversality Theorem* to the function. Hence $\mathfrak{F}^{\text{ev}}(\tilde{f}, \ . \ )$ is transverse to each $\mathcal{A}_s$ for a dense subset of functions $\tilde{f} \in W$. Since $f \in C^r(\mathbb{R}^n, \mathbb{R})$ was arbitrarily chosen, the set

$$\pitchfork^r (\mathbb{R}^n, \mathbb{R}; \mathfrak{F}, \mathcal{A}_s) := \{ f \in C^r(\mathbb{R}^n, \mathbb{R}) \mid \mathfrak{F} \pitchfork \mathcal{A}_s \}$$

is dense in $C^r_W(\mathbb{R}^n, \mathbb{R})$.

To complete the proof, we define the subset $V \subset C^r(\mathbb{R}^n, \mathbb{R})$ containing all functions f for which $\text{eval}(f, \ . \ )|_{G \cdot \mathcal{X}}$ is an immersion. By the definition of $\mathcal{A}_s$, this implies that

$$V = \left\{ f \in C^r(\mathbb{R}^n, \mathbb{R}) \;\middle|\; \mathfrak{F}(f)(G \cdot \mathcal{X}) \cap \mathcal{A}_s = \varnothing \quad \text{for all } s < m \right\}.$$

The theorem is proven, if we show that V is generic in $C^r_S(\mathbb{R}^n, \mathbb{R})$.

Now, suppose $\mathfrak{F}(f) \pitchfork \mathcal{A}_s$ holds for a function $f \in C^r(\mathbb{R}^n, \mathbb{R})$. Hence, from the definition of transversality (4.2) yields that for all $Y \in G \cdot \mathcal{X}$ with $\mathfrak{F}(f)(Y) \cap \mathcal{A}_s \neq \varnothing$ the inequality

$$\dim \mathcal{A}_s + \dim T_Y \mathfrak{F}(f) \geqslant \dim J^1(G \cdot \mathcal{X}, \mathbb{R}^m) \tag{4.23}$$

holds. However, using $\dim T_Y \mathfrak{F}(f) = \dim G$, equation (4.22) leads to

$$\dim \mathcal{A}_s + \dim T_Y \mathfrak{F}(f) \lneqq \dim J^1(G \cdot \mathcal{X}, \mathbb{R}^m),$$

which contradicts (4.23). Hence, $\mathfrak{F}(f) \pitchfork \mathcal{A}_s$ implies $\mathfrak{F}(f)(G \cdot \mathcal{X}) \cap \mathcal{A}_s = \varnothing$, if $s \leqslant m - 1$ and $m > 2 \dim G - 1$. Therefore, since $2 \dim G \cdot \mathcal{X} \leqslant m$ is assumed in the theorem, we have

$$\begin{aligned} V &= \left\{ f \in C^r(\mathbb{R}^n, \mathbb{R}) \;\middle|\; \mathfrak{F}(f) \pitchfork \mathcal{A}_s \quad \text{for all } r < m \right\} \\ &= \bigcap_{s<m} \pitchfork^r (\mathbb{R}^n, \mathbb{R}; \mathfrak{F}, \mathcal{A}_s). \end{aligned}$$

We have already shown that all sets of the form $\pitchfork^r (\mathbb{R}^n, \mathbb{R}; \mathfrak{F}, \mathcal{A}_s)$ are dense with respect to the weak topology. Lemma 4.4 yields that the sets are even generic with respect to the strong topology. Thus, V is generic in C^r_S, which complets the proof. □

Theorem 4.9 can be seen as a variation of the well known fact that immersions are dense in the function space $C^1_S(\mathcal{M}, \mathcal{N})$, if $\dim N \geqslant 2 \dim M$, where $\mathcal{M}, \mathcal{N}$ denote C^1 manifolds (cf. [50] p. 53). However, the same relationship will not be completely satisfied regarding the injectivity of $\text{eval}(f, \ . \ )|_{G \cdot \mathcal{X}}$. (Injective functions are dense in $C_S(\mathcal{M}, \mathcal{N})$, if $\dim N \geqslant 2 \dim M + 1$.) We want to give a short heuristic for that:

Let $\triangle_{\mathbb{R}^k} := \left\{ (x, x) \mid x \in \mathbb{R}^k \right\} \subset \mathbb{R}^k \times \mathbb{R}^k$ denote the diagonal of $\mathbb{R}^k \times \mathbb{R}^k$. Then, like in the proof before, a function $f \in C^1(\mathbb{R}^n, \mathbb{R}^m)$ is injective if $(f \times f) : (\mathbb{R}^n \times \mathbb{R}^n) \setminus \triangle_{\mathbb{R}^n} \to \mathbb{R}^m \times \mathbb{R}^m$

misses $\triangle_{\mathbb{R}^m}$. Comparing the dimensions, like it was done in (4.22) yields that this is equivalent to $(f \times f) \pitchfork \triangle_{\mathbb{R}^m}$ in the case $m > 2n$. Since the map

$$C^1(\mathbb{R}^n, \mathbb{R})^m \times (\mathbb{R}^n \times \mathbb{R}^n) \to \mathbb{R}^m \times \mathbb{R}^m, \quad (f_1, \dots, f_m; x, y) \mapsto \begin{pmatrix} f_1(x), & f_1(y) \\ \vdots & \vdots \\ f_m(x), & f_m(y) \end{pmatrix}$$

is linear and surjective in the first entry $f = (f_1, \dots, f_m)$ for $x \neq y$, the *Parametric Transversality Theorem* yields the transversality for a dense subset of functions f. In comparison, if we want to show the injectivity of $\operatorname{eval}(f, \ . \ ) : \mathbb{R}^{n \times m} \to \mathbb{R}^m$ with $f \in C^1(\mathbb{R}^n, \mathbb{R})$ in the same way, we have to study the map

$$C^1(\mathbb{R}^n, \mathbb{R}) \times \left(\mathbb{R}^{n \times m} \times \mathbb{R}^{n \times m}\right) \to \mathbb{R}^m \times \mathbb{R}^m, \quad (f, \mathcal{X}, \mathcal{Y}) \mapsto \begin{pmatrix} f(x_1), & f(y_1) \\ \vdots & \vdots \\ f(x_m), & f(y_m) \end{pmatrix}. \tag{4.24}$$

Here, the surjectivity in the first entry is not satisfied for all $\mathcal{X} = (x_1, \dots, x_m) \in \mathbb{R}^{n \times m}$ and $\mathcal{Y} = (y_1, \dots, y_m) \in \mathbb{R}^{n \times m}$ with $\mathcal{X} \neq \mathcal{Y}$. In fact, if $x_i = y_i$ for some $1 \leqslant i \leqslant m$, the map cannot be surjective in the first entry. However, we can regain surjectivity if we restrict the image of the function (4.24) to the appropriate components. More precisely, if we consider the case $x_l \neq y_l$ for all $l \in I := \{1, \dots, k\}$, $k < m$, then the function in (4.24) may not be transverse to $\triangle_{\mathbb{R}^m}$, but it is transverse to the linear subspace

$$\mathfrak{D}^I := \left\{(z, \tilde{z}) \in \mathbb{R}^m \times \mathbb{R}^m \;\middle|\; z_l = \tilde{z}_l \ , \quad \text{for all } l \in I\right\}. \tag{4.25}$$

This is stated in the following lemma.

Lemma 4.10
Let G be a finite-dimensional Lie group acting smoothly on $\mathbb{R}^n$ and let $\mathcal{X} = (x_1, \dots, x_m) \in \mathbb{R}^{n \times m}$ be a m-tuple of distinct sample-points in $\mathbb{R}^n$ such that $G \to G \cdot \mathcal{X}$, $g \mapsto g \cdot \mathcal{X}$ is an embedding of G in $\mathbb{R}^{n \times m}$. Moreover, let $k \leqslant m$ and let $I = \{i_1, \dots, i_k\}$, $1 \leqslant i_1 < \dots < i_k \leqslant m$ be given. We use the notation $\mathfrak{M}_I$ for the submanifold of $G \cdot \mathcal{X} \times G \cdot \mathcal{X}$ given by

$$\mathfrak{M}_I := \left\{ \begin{pmatrix} g \cdot x_1, & \tilde{g} \cdot x_1 \\ \vdots & \vdots \\ g \cdot x_m, & \tilde{g} \cdot x_m \end{pmatrix} \in G \cdot \mathcal{X} \times G \cdot \mathcal{X} \;\middle|\; g, \tilde{g} \in G, \ g \cdot x_l \neq \tilde{g} \cdot x_l \text{ for all } l \in I \right\}.$$

and the notation $\mathfrak{D}^I \subset \mathbb{R}^m \times \mathbb{R}^m$ for the linear subspace defined in (4.25). Let $\mathcal{N}$ be a submanifold of $\mathfrak{M}_I$ of dimension $\dim \mathcal{N} < k$. Then, the set of all $f \in C^r(\mathbb{R}^n, \mathbb{R})$, $r \geq 1$ that satisfy

$$\textit{eval}\,(f, \mathcal{N}) \cap \mathfrak{D}^I = \varnothing \tag{4.26}$$

is generic in $C^r_S(\mathbb{R}^n, \mathbb{R})$.
In particular, for $k > 2 \dim G$, we can set $\mathcal{N} = \mathfrak{M}_I$, since $\dim \mathfrak{M}_I \leq 2 \dim G < k$. We get the genericity of all $f \in C^r_S(\mathbb{R}^n, \mathbb{R})$ which satisfies

$$\textit{eval}\,(f, \mathfrak{M}_I) \cap \mathfrak{D}^I = \varnothing.$$

Before starting the proof, we make some remarks. We will use the notation $(y, \tilde{y}) \in \mathcal{N}$ for an element in $\mathcal{N}$. This means that y and $\tilde{y}$ consists of m-tuple of n-dimensional vectors, i.e. $y = (y_1, \ldots, y_m)$ and $\tilde{y} = (\tilde{y}_1, \ldots, \tilde{y}_m)$ with $y_i, \tilde{y}_i \in \mathbb{R}^n$ and $y_i = g \cdot x_i$, $\tilde{y}_i = \tilde{g} \cdot x_i$ for two elements $g, \tilde{g} \in G$. The evaluation map eval was defined in (4.10) as a function of the form $\mathrm{eval} : C(\mathbb{R}^n, \mathbb{R}) \times \mathbb{R}^{n \times m} \to \mathbb{R}^m$. In (4.26) we used the same notation for the function

$$\mathrm{eval} : C(\mathbb{R}^n, \mathbb{R}) \times \mathcal{N} \to \mathbb{R}^m \times \mathbb{R}^m, \quad (\tilde{f}, y, \tilde{y}) \mapsto \begin{pmatrix} \tilde{f}(y_1), & \tilde{f}(\tilde{y}_1) \\ \vdots & \vdots \\ \tilde{f}(y_m), & \tilde{f}(\tilde{y}_m) \end{pmatrix}. \tag{4.27}$$

Proof

Since the statement of the lemma is independent of the ordering of the sample-points $\mathcal{X} = (x_1, \ldots, x_m)$, without loss of generality we take $\{i_1, \ldots, i_k\} = \{1, \ldots, k\}$. We define polynomials $h_{l,g,\tilde{g}}$ in $\mathcal{P}_{n,2m-1}$ for each $(g, \tilde{g}) \in (\alpha_{\mathcal{X}} \times \alpha_{\mathcal{X}})^{-1}(\mathfrak{M}_{(1,\ldots,k)})$ and each $1 \leqslant l \leqslant k$ such that

$$h_{l,g,\tilde{g}}(y) = \begin{cases} 0\,, & y = gx_j,\ j \in \{1, \ldots, m\} \setminus \{l\} \\ 0\,, & y = \tilde{g}x_j,\ j \in \{1, \ldots, m\} \\ 1\,, & y = gx_l. \end{cases} \tag{4.28}$$

For instance, we can set

$$h_{l,g,\tilde{g}}(y) = \left(\prod_{j=1, j \neq l}^{m} a^\top (y - gx_j) \right) \left(\prod_{j=1}^{m} a^\top (y - \tilde{g}x_j) \right)$$

for an appropriate $a \in \mathbb{R}^n$. Of course the condition (4.28) can only be consistent, if $g \cdot x_l \neq \tilde{g} \cdot x_l$ for all $1 \leqslant l \leqslant k$, which is exactly the definition of $\mathfrak{M}_{(1,\ldots,k)}$.

In a first step, we take an arbitrary $f \in C^r(\mathbb{R}^n, \mathbb{R})$ and define the subset $W := f + \mathcal{P}_{n,2m-1}$. We apply the *Parametric Transversality Theorem* for

$$\mathfrak{F}^{\mathrm{ev}} : W \times \mathcal{N} \to \mathbb{R}^m \times \mathbb{R}^m, \quad \mathfrak{F}^{\mathrm{ev}} := \mathrm{eval}\big|_{W \times \mathcal{N}}$$

and the submanifold $\mathfrak{D}^{(1,\ldots,k)} \subset \mathbb{R}^m \times \mathbb{R}^m$.

In order to verify the assumptions of this theorem, suppose there is a point $(\tilde{f}, y, \tilde{y}) \in W \times \mathcal{N}$ such that $\mathfrak{F}^{\mathrm{ev}}(\tilde{f}, y, \tilde{y}) \in \mathfrak{D}^{(1,\ldots,k)}$, i.e. $\tilde{f}(y_l) = \tilde{f}(\tilde{y}_l)$ for all $1 \leqslant l \leqslant k$. Define the affine linear map

$$(\alpha_1, \ldots, \alpha_k) \mapsto \mathfrak{F}^{\mathrm{ev}}(\tilde{f} + \alpha_1 h_{1,g,\tilde{g}} + \ldots + \alpha_k h_{k,g,\tilde{g}}, y, \tilde{y}) = \begin{pmatrix} \tilde{f}(y_1) + \alpha_1, & \tilde{f}(\tilde{y}_1) \\ \vdots & \vdots \\ \tilde{f}(y_k) + \alpha_k, & \tilde{f}(\tilde{y}_k) \\ \tilde{f}(y_{k+1}), & \tilde{f}(\tilde{y}_{k+1}) \\ \vdots & \vdots \\ \tilde{f}(y_m), & \tilde{f}(\tilde{y}_m) \end{pmatrix}.$$

From the definition of $\mathfrak{D}^{(1,\ldots,k)}$, this map is transverse to $\mathfrak{D}^{(1,\ldots,k)}$ in $(\tilde{f}, y, \tilde{y})$. Therefore, $\mathfrak{F}^{\mathrm{ev}}$ is transverse to the entire vector-subsubspace $\mathfrak{D}^{(1,\ldots,k)}$. Finally, we have to verify part (c) of the *Parametric Transversality Theorem* in order to apply the theorem. Since $\dim \mathcal{N} < k$, we get

$$\dim \mathcal{N} + \dim \mathfrak{D}^{(1,\dots,k)} - \dim \mathbb{R}^{2m} < k + (2m-k) - 2m = 0. \tag{4.29}$$

Hence, since $\mathfrak{F}^{\mathrm{ev}}$ is a C^1 map, we can apply the *Parametric Transversality Theorem* to $\mathfrak{F}^{\mathrm{ev}}$. Thus, $\mathfrak{F}^{\mathrm{ev}}(\tilde{f},\ .\ )$ is transversal to $\mathfrak{D}^{(1,\dots,k)}$ for a dense subset of functions $\tilde{f} \in W$. Since $f \in C^r(\mathbb{R}^n, \mathbb{R})$ was arbitrarily chosen, the set

$$\pitchfork^r (\mathbb{R}^n, \mathbb{R}; \mathfrak{F}, \mathfrak{D}^{(1,\dots,k)}) := \Big\{ f \in C^r(\mathbb{R}^n, \mathbb{R}) \mid \mathfrak{F}(f) \pitchfork \mathfrak{D}^{(1,\dots,k)} \Big\}$$

with

$$\mathfrak{F}(f) := \mathrm{eval}(f,\ .\ )|_{\mathcal{N}} = \mathfrak{F}^{\mathrm{ev}}(f,\ .\ )$$

is dense in $C^r_W(\mathbb{R}^n, \mathbb{R})$.
To complete the proof, we define the set

$$V = \Big\{ f \in C^r(\mathbb{R}^n, \mathbb{R}) \;\Big|\; \mathrm{eval}(f, \mathcal{N}) \cap \mathfrak{D}^{(1,\dots,k)} = \varnothing \Big\}.$$

The lemma is proven, if we show that V is generic with respect to the strong topology.
Now, suppose that $\mathfrak{F}(f) \pitchfork \mathfrak{D}^{(1,\dots,k)}$ holds for a function $f \in C^r(\mathbb{R}^n, \mathbb{R})$. Hence, the definition of transversality (4.2) yields that for all $Y \in \mathcal{N}$ with $\mathfrak{F}(f)(Y) \in \mathfrak{D}^{(1,\dots,k)}$ the inequality

$$\dim \mathfrak{D}^{(1,\dots,k)} + \dim T_Y \mathfrak{F}(f) \geqslant \dim (\mathbb{R}^m \times \mathbb{R}^m) \tag{4.30}$$

is satisfied. Since $\dim T_Y \mathfrak{F}(f) = \dim \mathcal{N}$, equation (4.29) leads to

$$\dim \mathfrak{D}^{(1,\dots,k)} + \dim T_Y \mathfrak{F}(f) < \dim \mathbb{R}^{2m}$$

which contradicts (4.30). Therefore, the transversality of $\mathfrak{F}(f)$ to $\mathfrak{D}^{(1,\dots,k)}$ implies $\mathfrak{F}(f)(\mathcal{N}) \cap \mathfrak{D}^{(1,\dots,k)} = \varnothing$ and we have

$$\begin{aligned} V &= \Big\{ f \in C^r(\mathbb{R}^n, \mathbb{R}) \;\Big|\; \mathfrak{F}(f) \cap \mathfrak{D}^{(1,\dots,k)} = \varnothing \Big\} \\ &= \pitchfork^1 (\mathbb{R}^n, \mathbb{R}; \mathfrak{F}, \mathfrak{D}^{(1,\dots,k)}). \end{aligned}$$

Since we have already shown that the set $\pitchfork^1 (\mathbb{R}^n, \mathbb{R}; \mathfrak{F}, \mathfrak{D}^{(1,\dots,k)})$ is dense with respect to the weak topology, Lemma 4.4 yields that V is even generic with respect to the strong topology, which completes the proof. □

We continue to study under which conditions the function $\mathrm{eval}(f,\ .\ ) : G{\cdot}\mathcal{X} \to \mathbb{R}^m$ is injective. Unfortunately, it is not enough to require that $G{\cdot}\mathcal{X}$ is a manifold and that $m \in \mathbb{N}$ is sufficiently large. We want to unterline this with an example. Let $x_1 = (1\ 0\ 0)^\top$ and $x_l = (0\ l\ 0)^\top$, $l = 2, \dots, m$ denote the sample-points of $\mathcal{X}$ and let $G = SE(3)$. Using Theorem 4.8, the set $G \cdot \mathcal{X}$ forms an embedding if $m \geqslant 3$. Now, let $f \in C^1(\mathbb{R}^3, \mathbb{R})$ be arbitrarily given and let $\gamma : S^1 \to SE(3)$ denote the rotation around the axis $\big\langle (0\ 1\ 0)^\top \big\rangle$, i.e.

$$\gamma : e^{it} \mapsto \begin{pmatrix} \mathrm{Re}\ e^{it} & 0 & -\mathrm{Im}\ e^{it} & 0 \\ 0 & 1 & 0 & 0 \\ \mathrm{Im}\ e^{it} & 0 & \mathrm{Re}\ e^{it} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Then, $e^{it} \mapsto \operatorname{eval}(f, \gamma(e^{it}))$ is constant from the second component on. If we consider only the first component of this function, i.e. $e^{it} \mapsto e_1^\top \operatorname{eval}(f, \gamma(e^{it}))$, we get a real-valued function defined on S^1, which can not be injective for any choice of $f \in C^1(\mathbb{R}^3, \mathbb{R})$. Therefore, $\operatorname{eval}(f, \ . \ ) : G \cdot \mathcal{X} \to \mathbb{R}^m$ is not injective for any choice of $f \in C^1(\mathbb{R}^3, \mathbb{R})$. Roughly speaking, in this example the evaluation map $\operatorname{eval}(f, \ . \ ) : G \cdot \mathcal{X} \to \mathbb{R}^m$ cannot be injective, since too many sample-points lie on a line. This example suggests that conditions on the distribution of the sample-points are needed in order to guarantee injectivity of $\operatorname{eval}(f, \ . \ ) : G \cdot \mathcal{X} \to \mathbb{R}^m$. These conditions are summarized in the following.

Condition 4.1

Let G denote a finite-dimensional Lie group acting smoothly on $\mathbb{R}^n$ and let $\mathcal{X} = (x_1, \dots, x_m)$ be an ordered set of exactly $m = 2 \dim G + 1$ distinct sample-points $x_i \in \mathbb{R}^n$, $i = 1, \dots, m$, such that $G \to G \cdot \mathcal{X}$, $g \mapsto g \cdot \mathcal{X}$ is an embedding of G in $\mathbb{R}^{n \times m}$.
We suppose that each ordered subset $\mathcal{X}' \subset \mathcal{X}$ with $|\mathcal{X}'| \leqslant \dim G$ satisfies the inequality

$$\dim Stab_G(\mathcal{X}') < \dim G + 1 - |\mathcal{X}'| \tag{4.31}$$

and each ordered subset $\mathcal{X}' \subset \mathcal{X}$ with $|\mathcal{X}'| > \dim G$ satisfies

$$Stab_G(\mathcal{X}') = \{id\}\,, \tag{4.32}$$

where $Stab_G(\mathcal{X}')$ denotes the stabilizer of $\mathcal{X}'$.

A few remarks are in order. The action $G \times \mathbb{R}^n \to \mathbb{R}^n$, $(g, x) \mapsto g \cdot x$ is smooth. This implies that the stabilizer $\operatorname{Stab}_G(\mathcal{X}')$ is a Lie subgroup for all ordered subsets $\mathcal{X}' \subset \mathcal{X}$ (see e.g. [99], p.13). Thus, condition (4.32) states that the map $\alpha_{\mathcal{X}'} : G \to \mathbb{R}^{n \times |\mathcal{X}'|}$, $\alpha_{\mathcal{X}'}(g) := g \cdot \mathcal{X}'$ is an embedding for each subset $\mathcal{X}'$ with $\dim G < |\mathcal{X}'|$. For us, the condition (4.31) seems to be the most crucial part.
Two special cases of (4.31) can be discussed separately: First, if $\mathcal{X}' = \varnothing$, the inequality becomes $\dim G < \dim G + 1$, which is a trivial statement. Second, if the subset $\mathcal{X}' = \{a\}$ has only one element, condition (4.31) yields

$$\dim \operatorname{Stab}_G(\mathcal{X}') \leqslant \dim G - 1 < \dim G.$$

This implies that $\operatorname{Stab}_G(\{a\})$ is a proper subgroup of G for all $a \in \mathcal{X}$. Hence, we only have to verify (4.31) for all $\mathcal{X}'$ with $|\mathcal{X}'| > 1$. The following proposition summarizes some helpful tools to analyze the condition (4.31) for the special cases of the Euclidean and the volume-preserving affine group.

Proposition 4.11

Let $H \subset \mathbb{R}^n$ denote an affine subspace with $\dim H = d$. We define the stabilizer of H with respect to a group action $G \times \mathbb{R}^n \to \mathbb{R}^n$ as

$$Stab_G(H) := \big\{ g \in G \;\big|\; g \cdot x = x \text{ for all } x \in H \big\}\,. \tag{4.33}$$

Then,

$$\dim Stab_{SE(n)}(H) = \frac{1}{2}(n - d)(n - d - 1) \tag{4.34}$$

and

$$\dim Stab_{SA(n)}(H) = n(n-d) - 1. \tag{4.35}$$

Proof
Because $\mathrm{Stab}_G(g{\cdot}H) = g\mathrm{Stab}_{SE(n)}(H)g^{-1}$ for all $g \in G$, without loss of generality we consider the case $H = \langle e_1, \dots, e_d \rangle$. Then

$$\mathrm{Stab}_{SE(n)}(H) = \{(A,t) \in SE(n) \mid t = 0,\ Ae_k = e_k,\ k = 1, \dots, d\}$$

and therefore

$$\dim \mathrm{Stab}_{SE(n)}(H) = \dim SO(n-d) = \frac{1}{2}(n-d)(n-d-1).$$

In the same manner we get

$$\mathrm{Stab}_{SA(n)}(H) \cong SL(n-d) \times \mathbb{R}^{d\times(n-d)},$$

which yields (4.34). □

Example: Condition 4.1 in the case of the special Euclidean group

We consider the case $G = SE(n)$. Let $H(\mathcal{X}')$ denote the smallest affine subspace containing $\mathcal{X}'$. The equation (4.32) was already discussed in Theorem 4.8. We get the equivalent condition

$$|\mathcal{X}'| > \dim G \quad \Rightarrow \quad \dim H(\mathcal{X}') \geqslant n-1. \tag{4.36}$$

If we want to examine (4.31) we have to consider the stabilizer of $SE(n)$. Since the map $\mathbb{R}^n \to \mathbb{R}^n$, $x \mapsto g{\cdot}x$ is affine for each $g \in SE(n)$, we have $Stab_{SE(n)}(\mathcal{X}') = Stab_{SE(n)}(H(\mathcal{X}'))$, where we have used (4.33) for the definition of the stabilizer of a subspace. Hence, using (4.34) and Theorem 4.8 for $G = SE(n)$ the inequality (4.31) is equivalent to the fact that the inequality

$$\frac{1}{2}\left(n - \dim H(\mathcal{X}')\right)\left(n - \dim H(\mathcal{X}') - 1\right) < \frac{1}{2}(n^2+n) + 1 - |\mathcal{X}'| \tag{4.37}$$

is satisfied for all $\mathcal{X}' \subset \mathcal{X}$ with $0 < \dim H(\mathcal{X}') < n-1$. Using (4.36) and (4.37), we can obtain explicit geometric conditions on the set of sample-points which are equivalent to Condition 4.1. For example:

$n = 2$: Here, we do not have to verify (4.36) and (4.37), since they are always satisfied. Therefore, Condition 4.1 is satisfied for any set of sample-points $\mathcal{X}$ with $m = 7$ pairwise different elements.

$n = 3$: In this case, we have to consider $\mathcal{X} \subset \mathbb{R}^3$ with $m = 13$ elements. Condition (4.36) yields $\dim H(\mathcal{X}') \geq 2$ for $|\mathcal{X}'| > 6$. Moreover, (4.37) has to be verified for all subsets $\mathcal{X}' \subset \mathcal{X}$ with $\dim H(\mathcal{X}') = 1$. In this case (4.37) becomes

$$1 < 7 - |\mathcal{X}'|.$$

Hence, for $n = 3$ and $G = SE(3)$, Condition 4.1 is satisfied if and only if $\mathcal{X}$ has 13 elements and "any 6 of them are not lie on a line".

		$SE(n)$		$SA(n)$	
n	$\dim H(\mathcal{X}')$	m	Conditions on $\mathcal{X}'$	m	Conditions on $\mathcal{X}'$
2	1	7	no cond.	11	$\lvert\mathcal{X}'\rvert < 5$
3	1	13	$\lvert\mathcal{X}'\rvert < 6$	23	$\lvert\mathcal{X}'\rvert < 7$
	2		no cond.		$\lvert\mathcal{X}'\rvert < 10$
4	1	21	$\lvert\mathcal{X}'\rvert < 8$	39	$\lvert\mathcal{X}'\rvert < 9$
	2		$\lvert\mathcal{X}'\rvert < 10$		$\lvert\mathcal{X}'\rvert < 13$
	3		no cond.		$\lvert\mathcal{X}'\rvert < 17$
5	1	31	$\lvert\mathcal{X}'\rvert < 10$	57	$\lvert\mathcal{X}'\rvert < 11$
	2		$\lvert\mathcal{X}'\rvert < 13$		$\lvert\mathcal{X}'\rvert < 16$
	3		$\lvert\mathcal{X}'\rvert < 15$		$\lvert\mathcal{X}'\rvert < 21$
	4		no cond.		$\lvert\mathcal{X}'\rvert < 26$

Table 4.1: Geometric formulation of Condition 4.1 for $SE(n)$ and $SA(n)$. Let $\mathcal{X} \subset \mathbb{R}^n$ be a given set of m sample-points and $\mathcal{X}' \subset \mathcal{X}$. It was shown in the previous examples that condition (4.31) can be rewritten in the form $\dim H(\mathcal{X}') = d \Rightarrow |\mathcal{X}'| < k$. In this table we list d and k for each $n = 2, \ldots, 5$.

Such equivalent formulations of (4.31) and (4.32) can be found for any dimension $n \in \mathbb{N}$. We summarized some of them in Table 4.1.

Example: Condition 4.1 in the case of the special affine group

In the same manner we will tackle the case $G = SA(n)$. Let $H(\mathcal{X}')$ denote the smallest affine subspace containing $\mathcal{X}'$. Using Theorem 4.7, we get that condition (4.32) is equivalent to

$$|\mathcal{X}'| > \dim G \quad \Rightarrow \quad H(\mathcal{X}') = \mathbb{R}^n. \tag{4.38}$$

Moreover, we can use (4.35) in order to get an equivalent formulation of (4.31): The inequality

$$n\left(n - \dim H(\mathcal{X}')\right) - 1 < (n^2 + n - 1) + 1 - |\mathcal{X}'| \tag{4.39}$$

is satisfied for all $\mathcal{X}' \subset \mathcal{X}$ with $0 < \dim H(\mathcal{X}') < n$. Once again we consider some special cases:

- $n = 2$: *Here, we have $m = 11$ distinct sample-points and (4.39) yields that "any 5 of them are not lie on a line".*
- $n = 3$: *In this case, we have $m = 23$, and (4.39) yields that first, "any 7 elements are not lying on a line" and second that "any 10 elements are not lying on a plane".*

Equivalent formulations of (4.31) and (4.32) can be found for any dimension $n \in \mathbb{N}$. We summarized some of them in Table 4.1.

Theorem 4.12
Let G denote a finite-dimensional Lie group, acting smoothly on $\mathbb{R}^n$. Moreover, let $\mathcal{X} \subset \mathbb{R}^n$ denote a finite set of $m \geqslant 2\dim G + 1$ elements. If $\mathcal{X}$ contains a subset $\mathcal{Y} \subset \mathcal{X}$ of exactly $2\dim G + 1$ elements such that the tuple $(\mathcal{Y}, G)$ satisfies Condition 4.1, then the set

$$\left\{ f \in C^r(\mathbb{R}^n, \mathbb{R}) \;\middle|\; \textit{eval}(f;\ .\ )|_{G\cdot\mathcal{X}} : G \cdot \mathcal{X} \to \mathbb{R}^m \textit{ is injective} \right\}, \tag{4.40}$$

contains a generic subset in $C^r_S(\mathbb{R}^n, \mathbb{R})$, $r \geq 1$.
If $m = 2\dim G + 1$ and $(\mathcal{X}, G)$ does not satisfy Condition 4.1, then (4.40) is not dense in $C^r_S(\mathbb{R}^n, \mathbb{R})$.

Proof
Suppose $(\mathcal{Y}, G)$ satisfies Condition 4.1. Without loss of generality, we ignore the sample-points in $\mathcal{X} \setminus \mathcal{Y}$. Thus, we assume $\mathcal{X} = \mathcal{Y}$ and $m = |\mathcal{X}| = 2\dim G + 1$. Again, we denote with $x_1, \ldots, x_m$ the elements of $\mathcal{X}$. For a given space X, we use $\triangle_X = \{(x, x) \mid x \in X\}$ to denote the diagonal in the space $X \times X$ and we set $I = \{1, \ldots, m\}$.
We partition $G \cdot \mathcal{X} \times G \cdot \mathcal{X} \setminus \triangle_{G\cdot\mathcal{X}}$ as a finite union of sets

$$G \cdot \mathcal{X} \times G \cdot \mathcal{X} \setminus \triangle_{G\cdot\mathcal{X}} = \bigcup_{J \subset I,\ J \neq \varnothing} \mathcal{N}_J \tag{4.41}$$

with

$$\mathcal{N}_J := \left\{ \begin{pmatrix} g \cdot x_1, & \tilde{g} \cdot x_1 \\ \vdots & \vdots \\ g \cdot x_m, & \tilde{g} \cdot x_m \end{pmatrix} \in G \cdot \mathcal{X} \times G \cdot \mathcal{X} \;\middle|\; \begin{array}{l} g, \tilde{g} \in G, \\ g \cdot x_i = \tilde{g} \cdot x_i \text{ for all } i \in I \setminus J, \\ g \cdot x_i \neq \tilde{g} \cdot x_i \text{ for all } i \in J \end{array} \right\}.$$

We will use the abbreviation $\mathcal{X}' := (x_i \mid i \in I \setminus J) \in \mathbb{R}^{n \times (m-k)}$, $k = |J|$. First, we want to show that $\mathcal{N}_J$ is a submanifold of the manifold $\mathfrak{M}_J$ defined in Lemma 4.10. Let $\alpha : G \times \mathbb{R}^{n\times(m-k)} \to \mathbb{R}^{n\times(m-k)}$ denote the simultaneous group action of G on $m - k$ vectors in $\mathbb{R}^n$. Hence, we have

$$\alpha(\tilde{g}, \alpha(g, \mathcal{X}')) = \alpha(\tilde{g}g, \mathcal{X}').$$

Differentiating this formula with respect to $\tilde{g}$ at $\tilde{g} = id$ yields

$$D_1\alpha(id, g \cdot \mathcal{X}') = D_1\alpha(g, \mathcal{X}') \cdot T_{id}l(g), \tag{4.42}$$

where $l(g) : G \to G$, $h \mapsto gh$ denotes the left translation in the Lie group G and D_1 denotes the partial derivative with respect to the first component. Now, consider the function $F : G \times G \to \mathbb{R}^{n\times(m-k)}$, $F(g, \tilde{g}) := g \cdot \mathcal{X}' - \tilde{g} \cdot \mathcal{X}'$ with the tangential map

$$T_{(g,\tilde{g})}F(h, \tilde{h}) = \Big(D_1\alpha(g, \mathcal{X}')h, D_1\alpha(\tilde{g}, \mathcal{X}')\tilde{h} \Big).$$

Using (4.42), this map becomes

$$T_{(g,\tilde{g})}F(h, \tilde{h}) = \Big(D_1\alpha(id, g \cdot \mathcal{X}')T_{id}l(g)h, D_1\alpha(id, \tilde{g} \cdot \mathcal{X}')T_{id}l(\tilde{g})\tilde{h} \Big).$$

Thus, in the case $g \cdot \mathcal{X}' = \tilde{g} \cdot \mathcal{X}'$ we have

$$\operatorname{rk}(DF(g,\tilde{g})) = \operatorname{rk}\big(D_1\alpha(id, g\cdot\mathcal{X}')\big) = \dim G - \dim \operatorname{Stab}_G \mathcal{X}'.$$

The *Submersion Theorem* yields that $F^{-1}(0)$ is a submanifold with dimension $\dim G + \operatorname{Stab}_G(\mathcal{X}')$. Hence, since $\mathfrak{M}_J$ is a manifold, $\mathcal{N}_J$ is a submanifold with dimension $\dim G + \operatorname{Stab}_G(\mathcal{X}')$

Now, we will verify that the Lemma 4.10 can be applied for each submanifold $\mathcal{N} = \mathcal{N}_J$. Since the tuple $(\mathcal{X}, G)$ satisfies Condition 4.1, we conclude that

$$\mathcal{N}_J = \varnothing \quad \text{for} \quad m - |J| = |\,(x_i \mid i \in I \setminus J)\,| > \dim G.$$

Therefore, we only have to consider $\mathcal{N}_J$ for $m - |J| \leq \dim G$. To calculate the dimension of $\mathcal{N}_J$, we first consider the case $m - |J| > 0$. Using Condition 4.1, we get

$$\begin{aligned}\dim \mathcal{N}_J &= \dim G + \dim \operatorname{Stab}_G\left((x_i \mid i \in I \setminus J)\right)\\ &< 2\dim G + 1 - |(x_i \mid i \in I \setminus J)|.\\ &= m - (m - |J|)\\ &= |J|.\end{aligned}$$

For $m - |J| = 0$, we have $\mathcal{N}_J = \mathfrak{M}_I$ and therefore

$$\begin{aligned}\dim \mathcal{N}_J &= 2\dim G\\ &= m - 1\\ &< |J|.\end{aligned}$$

Hence, for all $J \subset I$, $J \neq \varnothing$ the set $\mathcal{N}_J$ is a submanifold of $\mathfrak{M}_I$ with $\dim \mathcal{N}_J < |J| = k$ and we can apply Lemma 4.10. For each $J \subset I$, $J \neq \varnothing$, the set of all $f \in C^r(\mathbb{R}^n, \mathbb{R})$ which satisfy

$$\operatorname{eval}(f, \mathcal{N}_J) \cap \mathfrak{D}^J = \varnothing$$

is generic in $C^r_S(\mathbb{R}^n, \mathbb{R})$. Since $\triangle_{\mathbb{R}^m} \subset \mathfrak{D}^J$, the same statement is true if we replace $\mathfrak{D}^J$ with $\triangle_{\mathbb{R}^m}$. Using (4.41), we can conclude that the set of all $f \in C(\mathbb{R}^n, \mathbb{R})$ which satisfy

$$\operatorname{eval}(f, G\cdot\mathcal{X} \times G\cdot\mathcal{X} \setminus \triangle_{G\cdot\mathcal{X}}) \cap \triangle_{\mathbb{R}^m} = \varnothing$$

contains a finite intersection of generic sets. Hence, the set (4.40) contains a generic subset in $C^r_S(\mathbb{R}^n, \mathbb{R})$, which proves the first part of the theorem.

Now, suppose $m = 2\dim G + 1$ and $(\mathcal{X}, G)$ does not satisfy Condition 4.1. Then, we find a subset $\mathcal{X}' \subset \mathcal{X}$ with $|\mathcal{X}'| \leqslant \dim G$ which contradicts (4.31) or with $|\mathcal{X}'| > \dim G$ which contradicts (4.32). In the first case, we get

$$\begin{aligned}\dim \mathcal{N}_J + \dim \mathfrak{D}^J &= \dim G + \dim \operatorname{Stab}_G(\mathcal{X}') + m + |\mathcal{X}'|\\ &\geqslant \dim G + \dim G + 1 - |\mathcal{X}'| + m + |\mathcal{X}'| \qquad (4.43)\\ &= \dim \mathbb{R}^{m\times 2},\end{aligned}$$

and in the second case, we get

$$\begin{aligned} \dim \mathcal{N}_J + \dim \mathfrak{D}^J &= \dim G + m + |\mathcal{X}'| \\ &\geqslant \dim G + m + \dim G + 1 \\ &= \dim \mathbb{R}^{m\times 2}. \end{aligned} \tag{4.44}$$

We will divide the rest of the proof in two steps. First, we will use (4.43) and (4.44) to construct a polynomial $p \in C(\mathbb{R}^n, \mathbb{R})$ such that $\operatorname{eval}(p, \ . \ ) : \mathcal{N}_J \to \mathbb{R}^{m\times 2}$ has a transversal intersection with $\mathfrak{D}^J$. In the second step, we will show that $\operatorname{eval}(f, \ . \ )|_{G\cdot\mathcal{X}}$ is not injective for all $f \in C_s(\mathbb{R}^n, \mathbb{R})$ in an open neighborhood around p.

In order to prove the first step, we assume without loss of generality that $\mathcal{X}' = (x_{k+1}, \dots, x_m)$ and set $J := \{1, \dots, k\}$. Furthermore, we fix $\tilde{g}, \hat{g} \in G$ and construct $(\tilde{y}, \hat{y}) \in \mathcal{N}_J$ by $\tilde{y} = (\tilde{y}_1, \dots, \tilde{y}_m)$ and $\hat{y} = (\hat{y}_1, \dots, \hat{y}_m)$ with $\tilde{y}_i = \tilde{g} \cdot x_i$ and $\hat{y}_i = \hat{g} \cdot x_i$, $i = 1, \dots, k$. We define the subset V of polynomials

$$V := \left\{ p \in \mathcal{P}_{n,2k-1} \mid p(\tilde{y}_j) = p(\hat{y}_j) = 0, \quad j \in J \right\}. \tag{4.45}$$

In particular, we simultaneously generate any derivative of p in $\tilde{y}_j$ and $\hat{y}_j$ by varying $p \in V$. For instance, we can use

$$p(x) = a^\top (x - \tilde{y}_1) \left(\prod_{i=2}^{k} \|x - \tilde{y}_i\|^2 \right) \left(\prod_{i=1}^{k} \|x - \hat{y}_i\|^2 \right)$$

to generate an arbitrary derivative in the first sample $\tilde{y}_1$.

Moreover, for each $p \in V$ we define the map

$$\Xi_p \in \underbrace{\mathbb{L}(\mathbb{R}^n, \mathbb{R}) \times \ldots \times \mathbb{L}(\mathbb{R}^n, \mathbb{R})}_{m\times-\text{times}} =: \mathcal{L},$$

$$\Xi_p(u_1, \dots, u_m, v_1, \dots, v_m) := \begin{pmatrix} \langle \nabla p(\tilde{y}_1), u_1 \rangle & \langle \nabla p(\hat{y}_1), v_1 \rangle \\ \vdots & \vdots \\ \langle \nabla p(\tilde{y}_m), u_m \rangle & \langle \nabla p(\hat{y}_m), v_m \rangle \end{pmatrix}$$

Since we can generate any gradient $\nabla p(\tilde{y}_i)$ and $\nabla p(\hat{y}_j)$, the map $V \to \mathcal{L}$, $p \mapsto \Xi_p$ is surjective. Therefore, using (4.43) or (4.44), we find an element $p \in V$ with

$$\Xi_p \left(T_{(\tilde{y}, \hat{y})} \mathcal{N}_J \right) + \mathfrak{D}^J = \mathbb{R}^{m\times 2}. \tag{4.46}$$

By definition of $\mathcal{N}_J$ and V, we have additionally $p(\tilde{y}_i) = p(\hat{y}_i) = 0$ for all $i = 1, \dots, m$. Thus, $\operatorname{eval}(p, \ . \ )|_{G\cdot\mathcal{X}}$ is not injective. Moreover, (4.46) and (4.45) yield

$$\operatorname{eval}(p, \ . \ )|_{\mathcal{N}_J} \pitchfork \mathfrak{D}^J \quad \text{and} \quad \operatorname{eval}(p, (\tilde{y}, \hat{y})) \in \mathfrak{D}^J,$$

which completes the first step of our proof.

In order to complet the second step of our proof, we apply the Proposition 4.2: There is an open neighborhood $U \subset C^r_S(\mathbb{R}^n, \mathbb{R})$ of p with

$$\operatorname{eval}(f, \mathcal{N}_J) \cap \mathfrak{D}^J \neq \varnothing \quad \text{for all } f \in U.$$

Recall that we have $f(\tilde{y}_i) = f(\hat{y}_i)$ for all $i = k+1, \dots, m$, by the definition of $\mathcal{N}_J$. Therefore, we conclude

$$\operatorname{eval}(f, \mathcal{N}_J) \cap \triangle_{\mathbb{R}^m} \neq \varnothing \quad \text{for all } f \in U.$$

Since $\mathcal{N}_J$ is a subset of $G \cdot \mathcal{X} \times G \cdot \mathcal{X}$, we finally get

$$\operatorname{eval}(f, G \cdot \mathcal{X} \times G \cdot \mathcal{X}) \cap \triangle_{\mathbb{R}^m} \neq \varnothing \quad \text{for all } f \in U.$$

Hence, for all $f \in U$ the map $\operatorname{eval}(f,\ .\ )|_{G \cdot \mathcal{X}}$ is not injective. □

We end this subsection with an examination of conditions under which the function $\operatorname{eval}(f,\ .\ )|_{G \cdot \mathcal{X}}$ additionally is homeomorphic. For this purpose, we introduce some notations:

$$\begin{aligned}
\operatorname{Prop}^r_S(\mathcal{M}, \mathcal{N}) &:= \big\{ f \in C^r_S(\mathcal{M}, \mathcal{N}) \ \big|\ f \text{ is proper} \big\} \\
\operatorname{Imm}^r_S(\mathcal{M}, \mathcal{N}) &:= \big\{ f \in C^r_S(\mathcal{M}, \mathcal{N}) \ \big|\ f \text{ is an injective immersion} \big\} \\
\operatorname{CEmb}^r_S(\mathcal{M}, \mathcal{N}) &:= \big\{ f \in C^r_S(\mathcal{M}, \mathcal{N}) \ \big|\ f \text{ is a closed embedding} \big\}
\end{aligned}$$

Theorem 4.13
Let G be a finite-dimensional Lie group acting smoothly on $\mathbb{R}^n$ and let $\mathcal{X} \subset \mathbb{R}^n$ be a set of $m \in \mathbb{N}$ different sample-points. If $\mathcal{X}$ contains a set $\mathcal{Y} \subset \mathcal{X}$ of exactly $2 \dim G + 1$ elements such that that $(G, \mathcal{Y})$ satisfies Condition 4.1, then, for $r \geq 2$, the set

$$\Gamma_1 := \big\{ f \in C^r_S(\mathbb{R}^n, \mathbb{R}) \ \big|\ eval(f,\ .\ )|_{G \cdot \mathcal{X}} \in Imm^r_S(G \cdot \mathcal{X}, \mathbb{R}^m) \big\}$$

is generic in $C^r_S(\mathbb{R}^n, \mathbb{R})$ and the set

$$\Gamma_2 := \big\{ f \in Prop^r_S(\mathbb{R}^n, \mathbb{R}) \ \big|\ eval(f,\ .\ )|_{G \cdot \mathcal{X}} \in CEmb^r_S(G \cdot \mathcal{X}, \mathbb{R}^m) \big\}$$

is generic in $Prop^r_S(\mathbb{R}^n, \mathbb{R})$.

Proof
According to Theorem 4.9, the set of all $f \in C^r_S(\mathbb{R}^n, \mathbb{R})$ for which $\operatorname{eval}(f,\ .\ )|_{G \cdot \mathcal{X}}$ is an immersion is a countable intersection of open and dense sets in $C^r_S(\mathbb{R}^n, \mathbb{R})$. Additionally, we get from Theorem 4.12 that the set of all f, for which $\operatorname{eval}(f,\ .\ )|_{G \cdot \mathcal{X}}$ is injective, is a countable intersection of open and dense sets in $C^r_S(\mathbb{R}^n, \mathbb{R})$. Hence, Γ_1 is generic in $C^r_S(\mathbb{R}^n, \mathbb{R})$.
Now, suppose that $f \in \operatorname{Prop}^r_S(\mathbb{R}^n, \mathbb{R})$. Then, the definition of eval yields

$$\operatorname{eval}(f,\ .\ ) \in \operatorname{Prop}^r_S(\mathbb{R}^{n \times m}, \mathbb{R}^m)$$

and therefore

$$\operatorname{eval}(f,\ .\ )|_{G \cdot \mathcal{X}} \in \operatorname{Prop}^r_S(G \cdot \mathcal{X}, \mathbb{R}^m).$$

Hence we have

$$\Gamma_2 = \Gamma_1 \cap \operatorname{Prop}^r_S(\mathbb{R}^n, \mathbb{R}).$$

From [50] p.38 we get that $\mathrm{Prop}_S^r(\mathbb{R}^n,\mathbb{R})$ is open in $C_S^r(\mathbb{R}^n,\mathbb{R})$. Since we already showed that Γ_1 is generic in $C_S^r(\mathbb{R}^n,\mathbb{R})$ we get that $\Gamma_1 \cap \mathrm{Prop}_S^r(\mathbb{R}^n,\mathbb{R})$ is even generic in $\mathrm{Prop}_S^r(\mathbb{R}^n,\mathbb{R})$, which completes the proof. □

We want to point out that Theorem 4.13 tackles the abstract task whether or not the evaluation map $\mathrm{eval}(f,\ .\ )$ for a given function f is a closed embedding. In image processing, f has always compact support, which implies that $\mathrm{eval}(f,\ .\ )|_{SE(n)\cdot\mathcal{X}}$ is never injective on the entire group $SE(n)$. However, if the support $T = \mathrm{supp}(f)$ of $f \in C^r(\mathbb{R}^n,\mathbb{R})$ is bounded and given a priori, it is advisable to shrink the deformation group to an open subset $U \subset G$ (not necessarily a subgroup), satisfying

$$\overline{U} \subset \alpha_{\mathcal{X}}^{-}(T \times \ldots \times T) = \alpha_{\mathcal{X}}^{-}\left(\mathrm{supp}(\mathrm{eval}(f,\ .\ )|_{G\cdot\mathcal{X}})\right).$$

A similar argument as for Theorem 4.13 yields that the set

$$\left\{f \in C_S^r(T,\mathbb{R}) \;\middle|\; \mathrm{eval}(f,\ .\ )|_{U\cdot\mathcal{X}} \in \mathrm{CEmb}_S^r(U\cdot\mathcal{X},\mathbb{R}^m)\right\}$$

is generic in $C_S^r(T,\mathbb{R})$. This might be more relevant to medical imaging, where T is a priori given in form of the region of interest.

4.2.3 Generic Properties for Fixed Images

The previous subsection focused on generic statements on the set of all images. However, such examinations stand outside of the classical task of image registration, where two images are given (and especially fixed). Up to this point, no conditions are given which ensure that the evaluation map $\mathrm{eval}(f,\ .\ )|_{G\cdot\mathcal{X}}$ is an embedding for a specific image $f \in C^1(\mathbb{R}^n,\mathbb{R})$. In fact, it is very easy to construct images f for which $\mathrm{eval}(f,\ .\ )$ is not even injective. For example, $f(x) = \exp(-\|x\|^2)$ is rotationally invariant which implies $\mathrm{eval}(f,g) = \mathrm{eval}(f,id)$ for all $g \in SO(n)$. Moreover, if an image has domains of constant gray values, i.e. $f^{-1}(a) := \{x \in \mathbb{R}^n \mid f(x) = a\}$ has interior points, it strongly depends on the distribution of the sample-points whether the evaluation map is injective or not[1]. In this subsection, we derive sufficient conditions for an image f such that

$$\phi_{f,\mathcal{X}} : G \to \mathbb{R}^m, \quad g \mapsto \mathrm{eval}(f, G\cdot\mathcal{X})$$

is an immersion (which implies at least local injectivity) for almost all choices of sampling-points $\mathcal{X} = (x_1 \ldots, x_m) \in \mathbb{R}^{n\times m}$.

Assumption 4.1
Let $f \in C^1(\mathbb{R}^n,\mathbb{R})$ be given and let G be a finite-dimensional Lie group acting smoothly on $\mathbb{R}^n$. The set

$$\mathfrak{M} := \left\{(x_1,\ldots,x_m) \in \mathbb{R}^{n\times m} \mid \mathit{rank}\; T_{id}\phi_{f,x_1,\ldots,x_m} < \dim G\right\}$$

is assumed to be a finite union of closed submanifolds with dimensions $nm - \dim G - 1$ or less.

[1] To avoid confusion: One might believe that images with domains of constant gray values are common in medical imaging. For example, the first picture in Fig. 3.1 suggested that all pixels which correspond to the bones have equal brightness. However, a more precise examination contradicts this cognition. As one can see in the second picture of Fig. 3.1 the gray-values vary extremely. This effect occurs for almost all sorts of tissue.

Theorem 4.14
Let Assumption 4.1 be satisfied, then the set of all $\mathcal{X} = (x_1, \dots, x_m) \in \mathbb{R}^{n\times m}$ for which $\phi_{f,\mathcal{X}}$ is an immersion, is open and dense in $\mathbb{R}^{n\times m}$.

Proof
We consider the simultaneous group action

$$\begin{aligned} &\alpha : G \times \mathbb{R}^{n\times m} \to \mathbb{R}^{n\times m}, \quad (g, x_1, \dots, x_m) \mapsto (g \cdot x_1, \dots, g \cdot x_m) \\ &\alpha_{\mathcal{X}} : G \to \mathbb{R}^{n\times m}, \quad g \mapsto (g \cdot x_1, \dots, g \cdot x_m). \end{aligned}$$

Since $x \mapsto g \cdot x$ is a submersion for all $g \in G$, the map α is also a submersion and therefore transversal to $\mathfrak{M}$. The *Parametric Transversality Theorem* implies $\alpha_{\mathcal{X}} \pitchfork \mathfrak{M}$ for all $\mathcal{X} = (x_1, \dots, x_m)$ in an open, dense subset $U \subset \mathbb{R}^{n\times m}$. Condition 4.1 yields

$$\dim G + \dim \mathfrak{M} < \dim \mathbb{R}^{n\times m}.$$

Therefore, $\alpha_{\mathcal{X}} \pitchfork \mathfrak{M}$ implies $\alpha_{\mathcal{X}}(G) \cap \mathfrak{M} = \emptyset$. This means that $T_{id}\phi_{f,g\cdot\mathcal{X}}$ is injective for all $\mathcal{X} \in U$ and all $g \in G$. The theorem is now proven, since $T_{id}\phi_{f,g\cdot\mathcal{X}}$ is injective if, and only if $T_g\phi_{f,\mathcal{X}}$ is injective. □

We want to point out that Assumption 4.1 restricts the set of admissible images, since $\mathfrak{M}$ depends on f. In comparison, Theorem 4.9 restricts the set of admissible sample-points, but it yields the same immersion statement as Theorem 4.14. However, testing whether Assumption 4.1 is satisfied or not is not practicable, since $\mathfrak{M}$ is embedded in the quite high dimensional vector space $\mathbb{R}^{n\times m}$. But for specific Lie groups, we can develop explicit assumptions for the images which implies Assumption 4.1.

Assumption 4.2
Let $f \in C^2(\mathbb{R}^2, \mathbb{R})$ be given. The Lie group G is the Euclidean Transformation Group $SE(2)$, the set of all critical points of f has no accumulation point and the set

$$\left\{ x \in \mathbb{R}^2 \;\middle|\; detH_f(x) = 0 \quad \vee \quad 0 = \nabla f(x)^\top H_f(x)^{-1} \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \nabla f(x) \right\} \tag{4.47}$$

is a finite union of closed submanifolds with dimension 1 or less.

Lemma 4.15
Let Assumption 4.2 be satisfied for a given function $f \in C^2(\mathbb{R}^2, \mathbb{R})$ and let $(b, c) \in \mathbb{R}^2 \times \mathbb{R} \backslash (0, 0)$ be arbitrary. Then, the set of zeros of the function

$$\mathbb{R}^2 \to \mathbb{R}, \quad x \mapsto \nabla f(x)^\top \cdot \left[b + c \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} x \right] \tag{4.48}$$

lie in a finite union of closed submanifolds of dimension 1 or less.

Proof
We denote the set defined in (4.47) with $\mathfrak{C}$. First, consider the case $c = 0$. Here, the function defined in (4.48) has the derivative $x \mapsto b^T H_f(x)$. Since $H_f(x)$ is regular for all $x \in \mathbb{R}^2 \setminus \mathfrak{C}$,

we can use the *Regular Value Theorem*. Hence, the set of zeros of (4.48) is a one-dimensional manifold in $\mathbb{R}^2 \setminus \mathfrak{C}$.
Now let $c \in \mathbb{R} \setminus \{0\}$. Again it is sufficient to show that zero is a regular value of the function (4.48) restricted to $\mathbb{R}^2 \setminus \mathfrak{C}$. Assuming that there is a critical point $x \in \mathbb{R}^2 \setminus \mathfrak{C}$ of (4.48), then the function and its derivative vanishes in x, i.e.

$$\nabla f(x)^\top \cdot \left[b + c \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} x \right] = 0 \tag{4.49}$$

and

$$\left[b + c \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} x \right]^\top H_f(x) + c \nabla f(x)^\top \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} = 0. \tag{4.50}$$

Since $\det H_f(x) \neq 0$ we can solve equation (4.50) for b. Then, equation (4.49) becomes

$$0 = c \cdot \nabla f(x)^\top H_f(x)^{-1} \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \nabla f(x),$$

which is a contradiction to $x \in \mathbb{R}^2 \setminus \mathfrak{C}$. □

Theorem 4.16

Assumption 4.2 implies Assumption 4.1.

Proof

We equip the vector space $T_{id}SE(2)$ with the basis

$$\begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}, \quad \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

Then, the linear map $T_{id}\phi_{f;x_1,\dots,x_m}$ has the representation matrix

$$X = \begin{pmatrix} \nabla f(x_1)^\top, & \nabla f(x_1)^\top \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} x_1 \\ \vdots & \vdots \\ \nabla f(x_m)^\top, & \nabla f(x_n)^\top \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} x_m \end{pmatrix} \tag{4.51}$$

and we have $(x_1, \dots, x_m) \in \mathfrak{M}$ if and only if rank $X < 3$. In the following, we split $\mathfrak{M}$ in $\mathfrak{M} = \mathfrak{M}_0 \cup \mathfrak{M}_1 \cup \mathfrak{M}_2$, where $\mathfrak{M}_i$ involves all elements of $\mathfrak{M}$ for which X has rank i. Subsequently, we prove separately for all $i = 0, 1, 2$ that $\mathfrak{M}_i$ is a finite union of closed submanifolds of dimension $2m - 4$ or less.
The case $i = 0$: The matrix X is identical zero if and only if $\nabla f(x_k) = 0$ for all $k = 1, \dots, m$. Using Condition 4.2 this causes $\mathfrak{M}_0$ to become discrete.

The case $i = 1$: Here, each column of X is a multiple of another column of X. Suppose that the third column of X is zero. Then all $\nabla f(x_i)$ lie in a one-dimensional linear subspace. Due to

$$\nabla f(x_i)^\top \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} x_i = 0 \quad \text{for all } i = 1, \dots m$$

all x_i lie in a one-dimensional linear subspace. Hence, in this case, the set of all $(x_1, \dots, x_m) \in \mathbb{R}^{2m}$ form a $m+1$ dimensional submanifold.
Now, supposs that the third column of X is not zero. Then, we partition the matrix X as

$$X = \begin{pmatrix} A & B \\ C & D \end{pmatrix}$$

with

$$A = \nabla f(x_1)^\top, \qquad B = \nabla f(x_1)^\top \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} x_1,$$

$$C = \begin{pmatrix} \nabla f(x_2)^\top \\ \vdots \\ \nabla f(x_m)^\top \end{pmatrix}, \qquad D = \begin{pmatrix} \nabla f(x_2)^\top \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} x_2 \\ \vdots \\ \nabla f(x_m)^\top \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} x_m \end{pmatrix}.$$

We will, without loss of generality, assume $B \neq 0$ and investigate the dimension of the set of all $(x_1, \dots, x_m) \in \mathbb{R}^{2m}$ of which X has rank 1. This occurs if, and only if there is a vector $b \in \mathbb{R}^2$ with $A = Bb^\top$ and $Db^\top = C$. This is equivalent to $DAB^{-1} = C$. Hence, for $B \neq 0$, the matrix X has rank 1 if, and only if the function

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix} \mapsto C - D \cdot B^{-1} \cdot A$$

becomes zero. Thus, we have to investigate the set of zeros of

$$(x_1, \dots, x_m) \mapsto \begin{pmatrix} \nabla f(x_2)^\top \left[\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} x_2 b^\top \right] \\ \vdots \\ \nabla f(x_m)^\top \left[\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} x_m b^\top \right] \end{pmatrix}, \quad b^\top = B^{-1} \cdot A \tag{4.52}$$

For a fixed $b \in \mathbb{R}^2$ these $m-1$ equations are not connected. Hence, we are able to reduce our investigation to the solutions of the equation

$$\rho(x) = 0 , \quad \text{with} \quad \rho : x \mapsto \left[\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - bx^\top \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \right] \nabla f(x).$$

A necessary condition for $x \in \mathbb{R}^2$ being a zero of ρ is either that $\nabla f(x) = 0$ or that the matrix in square brackets is irregular. Since Assumption 4.2 is valid, $\nabla f(x) = 0$ is only satisfied

for a discrete set. It is easy to show that the matrix in square brackets in the function term of ρ is irregular, if and only if x is in an appropriate one-dimensional affine subspace of $\mathbb{R}^2$. Therefore, the set of all zeros of ρ are in a one-dimensional manifold of the $\mathbb{R}^2$ and the set of all zeros of (4.52) lie in a manifold of dimension $m+1$. The case in which the first column of X is zero (an the second or third is not) can be handled in the same way.
<u>The case i=2</u>: Suppose that the first and the second column of X are linear independent. Then, we can assume without loss of generality that $\nabla f(x_1)$ and $\nabla f(x_2)$ are linear independent. We partition the matrix X as

$$X = \begin{pmatrix} A & B \\ C & D \end{pmatrix}$$

with

$$A = \begin{pmatrix} \nabla f(x_1)^\top \\ \nabla f(x_2)^\top \end{pmatrix}, \qquad B = \begin{pmatrix} \nabla f(x_1)^\top \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} x_1 \\ \nabla f(x_2)^\top \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} x_2 \end{pmatrix}$$

$$C = \begin{pmatrix} \nabla f(x_3)^\top \\ \vdots \\ \nabla f(x_m)^\top \end{pmatrix}, \qquad D = \begin{pmatrix} \nabla f(x_3)^\top \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} x_3 \\ \vdots \\ \nabla f(x_m)^\top \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} x_m \end{pmatrix}$$

Thus, we have $\det A \neq 0$ and we investigate the dimension of the set of all $(x_1, \ldots, x_m) \in \mathbb{R}^{2\times m}$ for which X has rank 2. This occurs if, and only if there is a vector $b \in \mathbb{R}^2$ with $Ab = B$ and $Cb = D$. This is equivalent to $CA^{-1}B = D$. Hence, for $det(A) \neq 0$, the matrix X has rank 2 if, and only if the function

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix} \mapsto D - C \cdot A^{-1} \cdot B \tag{4.53}$$

becomes zero. As in the case $i = 1$, each component of (4.53) has the form

$$x \mapsto \nabla f(x)^\top \cdot \left[\begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} x - b \right] \ , \quad b = A^{-1} \cdot B. \tag{4.54}$$

Due to Lemma 4.15, the set of zeros of (4.54) lies in a union of one-dimensional submanifolds for fixed b. Therefore, the set of zeros of (4.53) lies in a union of submanifolds of dimension $m+2$, which completes the proof under the assumption that the first and second column of X are linear independent.
We carry out the procedure above in the same fashion for the cases that the first and the third column of X are linear independent and the second and the third respectively. This yields two more forms of function (4.54):

$$x \mapsto \nabla f(x)^\top \cdot \left[\begin{pmatrix} -b_2 \\ 1 \end{pmatrix} + b_1 \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} x \right] \ , \quad b = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}.$$

$$x \mapsto \nabla f(x)^\top \cdot \left[\begin{pmatrix} 1 \\ -b_2 \end{pmatrix} + b_1 \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} x \right] \quad , \quad b = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}.$$

As shown in Lemma 4.15, the set of zeros of these functions lies always in a union of one-dimensional submanifolds. Therefore, the set of zeros of (4.53) lies in a union of submanifolds of dimension $m + 2$, which completes the proof. □

Theorem 4.16 confirms the intuition that images are not allowed to have domains of constant gray values if we want to reduce the registration task to a small amount of arbitrarily chosen sample-points. (Even domains with constant increasing or decreasing gray values are forbidden.) The next theorem states that Assumption 4.2 is generically satisfied.

Assumption 4.3

Let $\mathfrak{V} \subset C^3(\mathbb{R}^n, \mathbb{R})$ be a Banach space such that for any chosen $(x, v, A) \in \mathbb{R}^2 \times \mathbb{R}^2 \times Sym(2)$ there is a $f \in \mathfrak{V}$ with

$$\nabla f(x) = v \quad \wedge \quad H_f(x) = A.$$

Theorem 4.17

Let $\mathfrak{V}$ satisfies Assumption 4.3, then the subset of all $f \in \mathfrak{V}$, which satisfies Assumption 4.2 is open and dense in $\mathfrak{V}$.

Proof

Consider the jet-space $J^2(\mathbb{R}^2, \mathbb{R})$ and the subset

$$\mathfrak{D} := \left\{ [x, v, A] \in J^2(\mathbb{R}^2, \mathbb{R}) \;\middle|\; \det A = 0 \quad \vee \quad 0 = v^\top A^{-1} \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} v \right\}.$$

This subset is a finite union of submanifolds with codimension 1. Supposing we have $j^2 f \pitchfork \mathfrak{D}$ for a certain $f \in \mathfrak{V}$, then $(j^2 f)^-(\mathfrak{D})$ is a finite union of submanifolds with codimension 1. Hence, all points of $(j^2 f)^-(\mathfrak{D})$ are in the set defined in (4.47) and f satisfies Assumption 4.2. Therefore, it is enough to show that $j^2 f \pitchfork \mathfrak{D}$ holds for an open and dense subset of all $f \in \mathfrak{V}$. Consider the function

$$F^{\text{ev}} : \mathfrak{V} \times \mathbb{R}^2 \to J^2(\mathbb{R}^2, \mathbb{R}), \quad F^{\text{ev}}(f, x) = j^2_x f.$$

Since $j^2_x f$ is linear in f and Assumption 4.3 is satisfied, F^{ev} is a submersion. Therefore, F^{ev} is transversal to $\mathfrak{D}$. Hence, we can apply the *Parametric Transversality Theorem*, which completes the proof. □

We finish this section with two examples of spaces $\mathfrak{V}$ which satisfy Assumption 4.3. The first one is obviously the set of quadratic functions on $\mathbb{R}^2$

$$\mathfrak{V} := \left\{ f \in C(\mathbb{R}^2, \mathbb{R}) \mid f(x) = x^\top A x + b^\top x + c,\; A \in \mathbb{R}^{2\times 2},\; b \in \mathbb{R}^2,\; c \in \mathbb{R} \right\}.$$

More important, in the sense of image processing, is that fact, that spline function space of order 4 or greater satisfy Assumption 4.3, i.e. $\mathfrak{V} = \mathcal{S}^k_\lambda$ for any $\lambda > 0$ and $k \geq 4$. Thus, Assumption 4.2 is a generic property for images of the form (3.8).

Chapter 5

Image Registration using an Approximate-Newton Algorithm

In this chapter we take a closer look at the task of rigid image registration as an optimization problem on a Lie group. Although standard formulations of the problem focus on two- or three-dimensional images, the subsequent mathematical analysis is valid for any dimension. Thus, any gray-scale image I is identified as its associated intensity function $I : \mathbb{R}^n \to \mathbb{R}$ of the form (3.6) or (3.8), which we assume to be at least a three times continuously differentiable function with compact support. The rigid image registration task for two such functions R, T, then, amounts to find the Euclidean transformation (A, t) that minimizes the SSD-measure introduced in Section 3.3.1.

$$\int_{\mathbb{R}^n} \big(R(Ax+t) - T(x)\big)^2 dx. \tag{5.1}$$

Since the set of rigid body or affine transformations $\rho_{A,t} : \mathbb{R}^n \to \mathbb{R}^n, x \mapsto Ax + t$, forms a Lie group, i.e. the Euclidean transformation group $SE(n)$ and the Special Affine Group $SA(n)$ respectively, we obtain a least squares optimization problem on $SE(n)$ or $SA(n)$.

In this chapter, we apply the algorithms developed in Section 2.2 (Table 2.1 and Table 2.2) to the least square problem (5.1).

A bottleneck in implementing such algorithms lies in the difficulty of effectively evaluating the higher dimensional integrals, which may suffer from the curse of dimensionality. In this chapter we will present and discuss two different strategies to circumvent this problem. First, we use Quasi-Monte Carlo methods to approximate the integrals by taking samplings of the functions at suitable random points. This leads to an easily implementable algorithm in the case of image registration. Second, we used B-spline approximations of the images to get an approximation of the integral. The second method actually works better in practice as will be shown by examining concrete medical imaging tasks. Extensions to multimodal registration suing the mutual information measure are given too.

Although an overwhelming number of publications focuses on non-rigid registration where the task is to find a non-linear diffeomorphism, rigid registration is still of considerable interest and can be a good basis for a subsequent non-rigid registration phase; see e.g. [19], [20], [21], [22], and [23].

5.1 Registration on Smooth Function Spaces

In the case of the image-registration problem, G is either the Euclidean transformation group $SE(n)$ or the Special Affine Group $SA(n)$. The cost function which we want to maximize on the motion groups $G = SA(n), SE(n)$ is

$$\Phi : G \to \mathbb{R}, \quad \Phi(A,t) := \int_{\mathbb{R}^n} R(Ax+t)T(x)dx. \tag{5.2}$$

By invariance of the integral under volume-preserving maps, this function differs from the least squares term (5.1) in a constant. In particular, maximization of Φ is equivalent to minimization of (5.1).

Throughout this section, we have to assume that R is 3 times continuously differentiable, T is continuous and R and T have compact support. Since this ensures that Φ is 3 times continuously differentiable, we can apply the (μ, ν)-Newton algorithm, which was discussed in Section 2.2.

5.1.1 The Quasi-Monte Carlo Newton Algorithm

A direct comparison yields that the cost function Φ in (5.2) is of the form (2.17), if we set $f(x,y) = R(x)T(y)$ and $\varphi(x) = 1$. Thus, it is possible to apply the algorithms in the Tables 2.1 and 2.2 to the cost function Φ. Even though, a few specifications have to be made in order to apply it to the image registration task.

First, we note that the most time-consuming part of the algorithms is the calculation of the integrals appearing in the coefficients $\alpha, \beta, \gamma, \delta$ and ϵ in (2.27) – if they can be calculated at all. One way to overcome this difficulty is to approximate the integrals via Quasi-Monte Carlo methods. (See Appendix A for an introduction.) More precisely: Instead of maximizing the cost function Φ, we maximize

$$\Psi(A,t) := \frac{1}{N}\sum_{r=1}^{N} R(Ax_r+t)T(x_r). \tag{5.3}$$

Again, this cost function is of the form (2.17) and the coefficients $\alpha, \beta, \gamma, \delta$ and ϵ in (2.27) become the average of sampled function-values via

$$\begin{aligned}
\alpha_i &= \frac{1}{N}\sum_{r=1}^{N} T(x_r)\frac{\partial R}{\partial x_i}(Ax_r+t),\\
\beta_{i,j} &= \frac{1}{N}\sum_{r=1}^{N} T(x_r)\frac{\partial R}{\partial x_i}(Ax_r+t)(Ax_r+t)_j,\\
\gamma_{i,j,k} &= \frac{1}{N}\sum_{r=1}^{N} T(x_r)(Ax_r+t)_i\frac{\partial^2 R}{\partial x_j \partial x_k}(Ax_r+t),\\
\delta_{i,j,k,l} &= \frac{1}{N}\sum_{r=1}^{N} T(x_r)(Ax_r+t)_i(Ax_r+t)_j\frac{\partial^2 R}{\partial x_k \partial x_l}(Ax_r+t),
\end{aligned} \tag{5.4}$$

$$\epsilon_{i,j} = \frac{1}{N}\sum_{r=1}^{N} T(x_r)\frac{\partial^2 R}{\partial x_i \partial x_j}(Ax_r + t).$$

In contrast to a classical Monte–Carlo method, the sampling-points $x_i \in Q$ are chosen to be uniformly distributed in Q. The so-called Halton-sequence [100] is a good example of such uniformly distributed sampling points. It is shown in the literature (see e.g. [101]) that for such well chosen sampling points, the approximation error of (5.3) is bounded by $O((\log N)^n/N)$. (In contrast, the error of the Monte Carlo method tends to zero with the order $O(1/\sqrt{N})$.) We will give a definition of the Halton-sequence in Appendix A. We call the final methods the **QMC-Newton algorithms**. These algorithms are presented in Table 5.1 on $SE(n)$ and in Table 5.2 on $SA(n)$ respectively.

Before stating our main convergence result, a few remarks are in order. We use a standard Gauss elimination to solve the linear system in Step 3 for each algorithm. If this system is not solvable, a standard approach in optimization theory is to search for the least squares solution. Moreover, as is the case for all Newton methods, convergence of the algorithm is not guaranteed for an arbitrary initial condition $M_0 \in G$. Even if the algorithm converges, the limit needs not be a local maximum. In order to overcome this, one can either adapt a Gauss-Newton step, or first test if the Newton-direction (Ω, v) is an ascent direction. Alternatively, we can use the gradient of the objective function instead. Furthermore, one can make a line-search in the ascent direction, e.g. by using the Armijo-rule. We skip the straightforward details.

Theorem 5.1

Suppose that $T \in C(\mathbb{R}^n, \mathbb{R})$ and $R \in C^3(\mathbb{R}^n, \mathbb{R})$. Then the QMC-Newton algorithms described in Table 5.1 and Table 5.2, applied to the two following cost functions

(a) $\Phi : G \to \mathbb{R}$ defined by $\Phi(M) := \int_{\mathbb{R}^n} \varphi(x) f(M\cdot x, x)$, with $\varphi(x) := 1$, $f(x,y) := R(x)T(y)$ and with coefficients (2.27) and

(b) $\Psi : G \to \mathbb{R}$ defined by (5.3) with coefficients (5.4),

are quadratically convergent in a sufficiently small neighborhood of each nondegenerate critical point.

To avoid confusion: We mentioned before that we will use the Halton-sequence for the sample-points $(x_r)_{r=1}^N$ in (5.3). However, the statement of the theorem is valid for any given sequence $(x_r)_{r=1}^N$.

Proof

Let us first focus on the cost function Φ. We observe that the algorithm in Table 5.1 describes a (μ, ν)-Newton algorithm on $G = SE(n)$, where μ and ν are defined in (2.10) and (2.11). Respectively, Table 5.2 applies a $(\mu, \nu)-$Newton algorithm on $G = SA(n)$ where μ and ν are defined in (2.6) and (2.8). In both cases the parameterizations satisfy

$$D\mu_M(0) = D\nu_M(0), \tag{5.5}$$

for all $M \in G$. Moreover, the cost function Φ is in $C^3(G, \mathbb{R})$ since $T \in C(\mathbb{R}^n, \mathbb{R})$ and $R \in C^3(\mathbb{R}^n, \mathbb{R})$. Thus, we can apply Theorem 2.3. There exists an open neighborhood

Table 5.1: QMC-Newton Registration-Algorithm on SE(n)

Step 1.
Make an initial guess $M_0 \in SE(n)$ and set $m = 0$.

Step 2.
Calculate $\alpha_i,\ \beta_{i,j},\ \gamma_{i,j,k},\ \delta_{i,j,k,l}$ and $\epsilon_{i,j}$ for all $1 \leqslant i, j, k, l \leqslant n$ as defined in equation (5.4).

Step 3.
Solve the linear system which consists of the equations

$$\sum_{k>l}(\gamma_{l,k,i} - \gamma_{k,l,i})\Omega_{k,l} + \sum_{k} \epsilon_{i,k} v_k = -\alpha_i \quad \text{for all } 1 \leqslant i \leqslant n$$

and

$$\begin{aligned}\sum_{k>j}\frac{1}{2}(\beta_{i,k} + \beta_{k,i})\Omega_{k,j} - \sum_{k<j}\frac{1}{2}(\beta_{i,k} + \beta_{k,i})\Omega_{j,k} - \sum_{k>i}\frac{1}{2}(\beta_{j,k} + \beta_{k,j})\Omega_{k,i}& \\ + \sum_{k<i}\frac{1}{2}(\beta_{j,k} + \beta_{k,j})\Omega_{i,k} - \sum_{k>l}(\delta_{i,k,l,j} - \delta_{j,l,k,i} + \delta_{i,l,k,j} - \delta_{i,k,l,j})\Omega_{k,l}& \\ - \sum_{k}(\gamma_{j,k,i} - \gamma_{i,k,j})v_k = \beta_{i,j} - \beta_{j,i} \quad \text{for all } 1 \leqslant i < j \leqslant n&\end{aligned}$$

with the unknowns v_i and $\Omega_{i,j}$, $j > i$.

Step 4.
Construct the $n \times n$ matrix Ω with the entries $\Omega_{i,j}$. In the case of $j > i$ use the solution of step 3. Else, set

$$\Omega_{i,j} = \begin{cases} -\Omega_{j,i} & \text{for } j < i \\ 0 & \text{for } j = i. \end{cases}$$

Compute

$$M_{m+1} := \nu^{QR}_{M_m}(\Omega, v),$$

where ν^{QR} is defined in (2.11).

Step 5.
Set $m = m + 1$ and goto Step 2.

Table 5.2: QMC-Newton Registration-Algorithm on SA(n)

Step 1.
Make an initial guess $M_0 \in SA(n)$ and set $m = 0$.

Step 2.
Calculate α_i, $\beta_{i,j}$, $\gamma_{i,j,k}$, $\delta_{i,j,k,l}$ and $\epsilon_{i,j}$ for all $1 \leqslant i, j, k, l \leqslant n$ as defined in equation (5.4).

Step 3.
Solve the linear system described in (2.32) - (2.36) with the unknowns v_i and $\Omega_{i,j}$, $(i,j) \neq (n,n)$.

Step 4.
Construct the $n \times n$ matrix Ω with the entries $\Omega_{i,j}$. In the case of $(j,i) \neq (n,n)$ use the solution of Step 3. Else, set

$$\Omega_{n,n} = -\sum_{k \neq n} \Omega_{k,k}$$

and compute

$$M_{m+1} := \nu^{QR}_{M_m}(\Omega, v),$$

where ν^{QR} is defined in (2.8).

Step 5.
Set $m = m + 1$ and goto Step 2.

$V \subset G$ of each critical point $M^* \in G$ such that the point sequence $(M_k)_{k \in \mathbb{N}_0}$ generated by the algorithms in Table 5.1 or Table 5.2, converges quadratically to M^*, if $M_0 \in V$, which proves (a).
In order to prove (b), we exploit the fact that Ψ is also a function of type (2.15). Therefore, using (5.4) in (2.28)-(2.29) and (2.32)-(2.36), respectively, yields again a (μ, ν)-Newton step, and the local quadratic convergence is obtained by the same argument which was used for the function Φ. □

A few remarks are in order: Firstly, the condition that R be three times continuously differentiable is only needed to ensure the local quadratic convergence behavior. The QMC-Newton algorithms need only evaluations of the first and second derivative of R. Moreover, since the raw image data are usually given in a discretized form, a previous interpolation step is needed to extend the function continuously. If the interpolation order is high enough, then $R \in C^3(\mathbb{R}^n, \mathbb{R})$ can always be guaranteed (see Section 3.2).
In [45], sufficient conditions for local quadratic convergence are: a C^3 cost function, condition

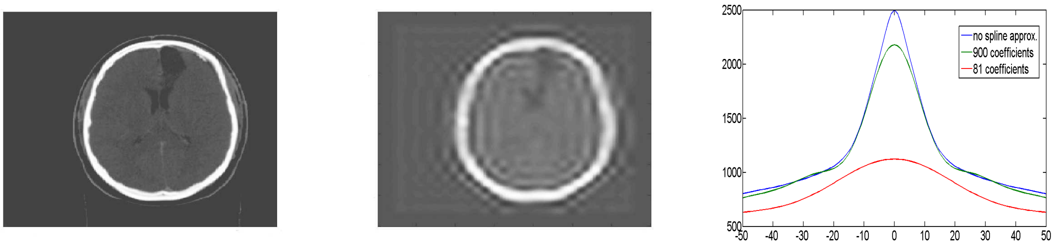


Figure 5.1: The first image shows a 2D-slide of a CT data-set of a head with 350 × 350 pixels. The second image is a spline-approximation with 900 basis-functions. The appearing artefacts may be caused by the Gibbs phenomena. The blue graph in the third image describes the cost function (5.2) by translating the image on the left against itself. For the green and red graph we replace the image with its spline-approximation with 900 and 81 basis-functions respectively.

(5.5) and the non-degenericity of the critical points. We will show in Section 5.1.2 that the last condition is fulfilled for a generic choice of the images.

Note that the cost function (5.3) is only an approximation of (5.2); a more sensible choice would be the discretization of the least squares as

$$\sum_{r=1}^{N}(R(Ax_r + t) - T(x_r))^2. \tag{5.6}$$

The local quadratic convergence result above would hold as well for suitably adapted choices of coefficients (5.4).

The numerical experiments in Section 5.3.1 will support the local quadratic convergence of the algorithms. This property is very useful if the region in which the algorithms converge quadratically is large. However, in Fig.5.1 it is demonstrated that this region can be very small if R and T are interpolations of raw medical images. (The region of quadratic convergence is a subset of the region in which the cost function is convex.) One way to overcome this is to smooth the image up to a certain level and use the smoothed image data instead. The result can then be used as the initial guess for the algorithms and applied to less smoothed images etc. Throughout this work, we perform the smoothing by projecting the images onto a finite-dimensional function-space, namely a spline-basis (see Section 3.2) for details. Fig.5.1 shows the change of the objective function caused by varying the degree of smoothness. In all the examples which we have been considered in this work, the smoothing of the images yields an increase of the domain of quadratic convergence.

5.1.2 A Generic Property of the Cost Function

In Theorem 5.1 we proved the local quadratic convergence of the QMC-algorithm under the condition that the critical points of the cost functions are nondegenerate. In this subsection we will show that this condition is generically fulfilled in image registration, i.e. on very mild conditions on the reference image $T \in C(\mathbb{R}^n, \mathbb{R})$, the set of all template images $R \in C^3(\mathbb{R}^n, \mathbb{R})$ for which the algorithm converges locally quadratically is open and dense in $C^3(\mathbb{R}^n, \mathbb{R})$ in terms of the strong topology.

In the jet-manifold $J^2(G,\mathbb{R})$ the subset

$$\mathfrak{U} = \left\{ j_x^2 h \;\middle|\; h \in C^2(G,\mathbb{R}),\ x \in G,\ \nabla h(x) = 0,\ \det \mathrm{Hess}_h(x) = 0 \right\} \tag{5.7}$$

contains all degenerate critical points.

Proposition 5.2
The set $\mathfrak{U}$ in (5.7) is a finite union of submanifolds $\mathfrak{U} = \mathfrak{U}_1 \cup \ldots \cup \mathfrak{U}_m$ of $J^2(G,\mathbb{R})$ with

$$\dim \mathfrak{U}_1 \leqslant \ldots \leqslant \dim \mathfrak{U}_m = \frac{3}{2} \dim G + \frac{1}{2} (\dim G)^2.$$

Proof
We will use the abbreviation $V = \mathbb{R} \times \mathbb{L}(\mathfrak{g},\mathbb{R}) \times \mathbb{L}^2_{\mathrm{Sym}}(\mathfrak{g},\mathbb{R})$, where $\mathfrak{g}$ denotes the Lie algebra of G. Since $\mathfrak{g}$ is also a vector space, we can write the elements of V in the form $j_0^2 v$ with $j_0^2 v \in J^2(\mathfrak{g},\mathbb{R})$. Moreover, we write $l(g) : G \to G,\ h \mapsto gh$ for the left-translation in the Lie group G and $\exp : \mathfrak{g} \to G$ for the exponential map.
Since the function

$$J^2(G,\mathbb{R}) \to G \times V, \quad j_g^2 f \mapsto (g, j_0^2 (f \circ l(g) \circ \exp)$$

is smooth and has the smooth inverse map

$$G \times V \to J^2(G,\mathbb{R}), \quad (g, j_0^2 v) \mapsto j_g^2 (v \circ \exp^{-1} \circ l(g^{-1})),$$

we get

$$J^2(G,\mathbb{R}) \cong G \times \mathbb{R} \times \mathbb{L}(\mathfrak{g},\mathbb{R}) \times \mathbb{L}^2_{\mathrm{Sym}}(\mathfrak{g},\mathbb{R}). \tag{5.8}$$

Thus, the diffeomorphic image of $\mathfrak{U}$ is

$$\left\{ (g,a,0,A) \in G \times \mathbb{R} \times \mathbb{L}(\mathfrak{g},\mathbb{R}) \times \mathbb{L}^2_{\mathrm{Sym}}(\mathfrak{g},\mathbb{R}) \;\middle|\; \det A = 0 \right\}.$$

According to [102], $\left\{ A \in \mathbb{L}^2_{\mathrm{Sym}}(\mathfrak{g},\mathbb{R}) \mid \det A = 0 \right\}$ is a finite union of submanifolds in $\mathbb{L}(\mathfrak{g},\mathbb{R})$ with codimension larger than 0. Thus, we get $\mathfrak{U} = \mathfrak{U}_1 \cup \ldots \cup \mathfrak{U}_m$ for appropriate submanifolds of $J^2(G,\mathbb{R})$. Assuming without loss of generality that $\mathfrak{U}_m$ has the largest dimension, we get

$$\begin{aligned} \dim \mathfrak{U}_m &= \dim G + \dim \mathbb{R} + \dim \mathbb{L}^2_{\mathrm{Sym}}(\mathfrak{g},\mathbb{R}) - 1 \\ &= \dim G + \frac{1}{2} \dim G (\dim G + 1). \end{aligned}$$

□

Once again, we consider the Lie groups $G = SA(n)$ and $G = SE(n)$ respectively. We wish to study the influence of the images R and T on the cost functions Ψ and Φ of formula (5.2) and (5.3). Therefore, we introduce the linear maps $\hat{\Psi}, \hat{\Phi} : C^3(\mathbb{R}^n,\mathbb{R}) \to C^3(G,\mathbb{R})$ which are defined by

$$\hat{\Psi}(R)(A,t) := \frac{1}{N} \sum_{i=1}^{N} T(x_i) R(Ax_i + t)$$

and, respectively,

$$\hat{\Phi}(R)(A,t) := \int_{\mathbb{R}^n} T(x)R(Ax+t)dx.$$

For a given function $R \in C^3(\mathbb{R}^n, \mathbb{R})$ the cost function $\hat{\Psi}(R)$ has no degenerate critical points if, and only if $j^2\hat{\Psi}(R)$ misses $\mathfrak{U}$. Since (4.8) yields

$$\dim J^2(G,\mathbb{R}) = \frac{5}{2}\dim G + \frac{1}{2}(\dim G)^2 + 1$$

we get

$$\dim G + \dim \mathfrak{U}_i < \dim J^2(G,\mathbb{R}), \quad i = 1, \ldots, m.$$

By means of Theorem 4.1 we get that $j^2\hat{\Psi}(R)$ misses $\mathfrak{U}$ if, and only if $j^2\hat{\Psi}(R)$ is transverse to every submanifold $\mathfrak{U}_i$, $i = 1, \ldots, m$. The same statement applies to the function $\hat{\Phi}$. We will make use of these equivalences in the proof of Theorem 5.4. In the following Lemma 5.3, we formulate a useful approximation statement.

Lemma 5.3
Let $\mathcal{P}_{n,k}$, $k \in \mathbb{N}$ denote the set of polynomials in n variables with degree less than or equal to k. For $f \in L^2(K,\mathbb{R})$, $K \subset \mathbb{R}^n$ compact, let $p_{f,k} \in \mathcal{P}_{n,k}$ denote the best-approximated polynomial of f in $\mathcal{P}_{n,k}$ with respect to the L^2-norm. If $f_1, \ldots, f_m \in L^2(K,\mathbb{R})$ are linearly independent, then there exists a $k \in \mathbb{N}$ such that $p_{f_1,k}, \ldots, p_{f_m,k}$ are also linearly independent.

Proof
Since $f_1, \ldots, f_m \in L^2(K,\mathbb{R})$ are linear independent, we get

$$\|\lambda_1 f_1 + \ldots + \lambda_m f_m\|_{L^2}^2 \neq 0 \tag{5.9}$$

for all $(\lambda_1, \ldots, \lambda_m) \in S^{m-1}$. Regard the left hand side of (5.9) as a function of the form $S^{m-1} \to \mathbb{R}$ in the variables λ_i. Since S^{m-1} is compact, this function has a minimum $\mu \in \mathbb{R}$ and with (5.9) we get

$$0 < \mu \leq \|\lambda_1 f_1 + \ldots + \lambda_m f_m\|_{L^2}^2 \tag{5.10}$$

for all $(\lambda_1, \ldots, \lambda_m) \in S^{m-1}$. Now, we choose $k \in \mathbb{N}$ large enough such that $\|p_{f_i,k} - f_i\|^2 < \frac{\mu}{m}$ is satisfied for all $i = 1, \ldots, m$. Assuming that $p_{f_1,k}, \ldots, p_{f_m,k}$ are linear dependent, there is $(\lambda_1, \ldots, \lambda_m) \in S^{m-1}$ with

$$\lambda_1 p_{f_1,k} + \ldots + \lambda_m p_{f_m,k} = 0$$

and therefore

$$\begin{aligned}\|\lambda_1 f_1 + \ldots + \lambda_m f_m\|_{L^2}^2 &= \|\lambda_1(f_1 - p_{f_1,k}) + \ldots + \lambda_m(f_m - p_{f_m,k})\|_{L^2}^2 \\ &\leq \sum_{i=1}^{m} \lambda_i \|p_{f_i,k} - f_i\|^2\end{aligned}$$

$$< \mu.$$

Since this contradicts (5.10), the polynomials $p_{f_1,k}, \ldots, p_{f_m,k}$ must be linear independent.

□

Theorem 5.4
Suppose that the template image $T \in C^3(\mathbb{R}^n, \mathbb{R})$ has compact support and satisfies the following conditions:

a) Regarding optimization problem (5.2), T is not identical to zero.

b) Regarding optimization problem (5.3), there are $k = \frac{1}{2}(n+1)(n+2)$ elements of the sequence $(x_i)_{i=1}^N$ which do not lie on a quadric hypersurface and $T(x_i) \neq 0$ for all these k elements.

Then, the cost functions Φ and Ψ have no degenerate critical points, for a generic set of reference images $R \in C^3_S(\mathbb{R}^n, \mathbb{R})$.

Proof
Following the argument above, we define

$$A_{\Psi,i} := \pitchfork^3 (\mathbb{R}^n, \mathbb{R}; j^2\hat{\Psi}, \mathfrak{U}_i) := \left\{ R \in C^3(\mathbb{R}^n, \mathbb{R}) \;\middle|\; j^2\hat{\Psi}(R) \pitchfork \mathfrak{U}_i \right\}$$

and $A_{\Phi,i} := \pitchfork^3 (\mathbb{R}^n, \mathbb{R}; j^2\hat{\Phi}, \mathfrak{U}_i)$ respectively. We have to show that $A_{\Psi,i}$ and $A_{\Phi,i}$ are open and dense in $C^3_S(\mathbb{R}^n, \mathbb{R})$.
The *Jet Transversality Theorem* 4.6 yields the openess of $\pitchfork^3 (G, \mathbb{R}; j^2, \mathfrak{U}_i)$. Since $\hat{\Psi}$ and $\hat{\Phi}$ are continuous maps, we conclude that the sets $A_{\Psi,i}$ and $A_{\Phi,i}$ are open. Hence, we only have to show the denseness of both sets with respect to the strong topology. Applying Lemma 4.4 to $\mathfrak{F} = j^2\hat{\Psi}$ and $\mathfrak{F} = j^2\hat{\Phi}$ respectively yields that it is enough to show the denseness of both sets in $C^3_W(\mathbb{R}^n, \mathbb{R})$.
With this preliminaries at hand, let us complete the proof for the function $\hat{\Psi}$. Consider the vector space $V = \mathbb{R} \times \mathbb{R}^n \times \mathrm{Sym}(n)$, endowed with the standard scalar-product

$$\langle\ .\ \rangle : V \times V \to \mathbb{R}, \quad \langle (\alpha, a, A), (\beta, b, B) \rangle = \alpha\beta + a^\top b + tr(AB) \tag{5.11}$$

and the map

$$\varphi : \mathbb{R}^n \to V\ , \quad x \mapsto (1, x, xx^\top)^\top.$$

Since a quadric hypersurface Q is defined via

$$Q_{\alpha,a,A} = \left\{ x \in \mathbb{R}^n \mid x^\top A x + a^\top x + \alpha = 0 \right\},$$

for an arbitrary element $(\alpha, a, A) \in V \setminus \{0\}$ we get the equivalence

$$x \in Q_{\alpha,a,A} \;\Leftrightarrow\; \langle (\alpha, a, A), \varphi(x) \rangle = 0.$$

Hence, exploiting the condition that $k = \frac{1}{2}(n+1)(n+2)$ elements of the sequence $(x_i)_{i=1}^N$ do not lie on a quadric hypersurface, yields

$$\operatorname{span}\{\varphi(x_1), \dots, \varphi(x_N)\} = V.$$

Moreover, $\{x_i\}_{i=1}^N$ lie on a quadric hyperface if, and only if $\{Ax_i + t\}_{i=1}^N$ lie on a quadric hyperface for any $(A, t) \in G$. Thus, we get

$$\operatorname{span}\{\varphi(Ax_1 + t), \dots, \varphi(Ax_N + t)\} = V.$$

Exploiting the condition that $T(x_i) \neq 0$ for those elements, we get

$$\operatorname{span}\{T(x_1)\varphi(Ax_1 + t), \dots, T(x_N)\varphi(Ax_N + t)\} = V \tag{5.12}$$

for all $(A, t) \in G$. Hence, the linear map generated by the matrix

$$M = \Big(T(x_1)\varphi(Ax_1 + t), \dots, T(x_N)\varphi(Ax_N + t)\Big) \tag{5.13}$$

is surjective for each $(A, t) \in G$.
Now, consider formula (5.4) as a linear map of the form

$$(j^2_{Ax_1+t}R, \dots, j^2_{Ax_N+t}R) \mapsto (\alpha, \beta, \gamma, \epsilon, \delta). \tag{5.14}$$

Let $m_1, \dots, m_k$ denote the rows of M, i.e. $M^\top = (m_1^\top, \dots, m_k^\top)$. Then, $m_1, \dots, m_k$ can be used to build the rows of the representation matrix of (5.14) in the following way:

$$\begin{aligned}
\alpha_i &= m_1 \cdot \left(\frac{\partial R}{\partial x_i}(Ax_1 + t), \dots, \frac{\partial R}{\partial x_i}(Ax_N + t)\right)^\top, \\
\beta_{i,1} &= m_2 \cdot \left(\frac{\partial R}{\partial x_i}(Ax_1 + t), \dots, \frac{\partial R}{\partial x_i}(Ax_N + t)\right)^\top, \\
&\ \vdots \\
\delta_{n,n,i,j} &= m_k \cdot \left(\frac{\partial^2 R}{\partial x_i \partial x_j}(Ax_1 + t), \dots, \frac{\partial^2 R}{\partial x_i \partial x_j}(Ax_N + t)\right)^\top
\end{aligned}$$

Since the linear map generated by M is surjective, the rows $m_1, \dots, m_k$ are linear independent. Therefore, the map (5.14) is also surjective. Hence, with the formulas (2.28) and (2.29) for $G = SE(n)$, and (2.32)-(2.36) for $G = SA(n)$ respectively, the map

$$(j^2_{Ax_1+t}R, \dots, j^2_{Ax_N+t}R) \mapsto j^2_{(A,t)}\hat{\Psi}(R). \tag{5.15}$$

is surjective for each chosen $(A, t) \in G$. Moreover, the map (5.15) is transverse to every submanifold of $J^2(G, \mathbb{R})$.
Now, let $R \in C^3(\mathbb{R}^n, \mathbb{R})$ be arbitrarily chosen. In order to show the denseness of $A_{\Psi,i}$ in terms of the weak topology, it is enough to show that $A_{\Psi,i} \cap [R + \mathcal{P}_{n,2N}]$ is dense. In the case of the map

$$F^{\text{ev}} : [R + \mathcal{P}_{n,2N}] \times G \to J^2(G, \mathbb{R})$$

$$(R+p, A, t) \mapsto j^2_{(A,t)}\hat{\Psi}(R+p)$$

we have already shown the subjectivity, if $(A,t) \in G$ is fixed. Hence, using the *Parametric Transversality Theorem* 4.5 we get that $F^{\mathrm{ev}}(R+p,\ .\ )$ is transverse to $\mathfrak{U}_\mathfrak{i}$ for a dense subset of $\mathcal{P}_{n,2N}$. Therefore, $A_{\Psi,i}$ is dense in $C^3_W(\mathbb{R}^n, \mathbb{R})$ which completes the proof in the case of the cost function Ψ.

Let us now consider the cost function Φ. Following the argument above, it is enough to prove that $A_{\Phi,i}$ is dense in $C^3_W(\mathbb{R}^n, \mathbb{R})$. Due to the conditions for the template image T, we can find a sequence of $N = \frac{1}{2}(n+1)(n+2)$ sample-points $(x_i)_{i=1}^N \subset \operatorname{supp}(T)$ which do not lie on a quadratic hypersurface. Hence, the equation (5.12) is again satisfied. Therefore, the set of functions

$$\mathcal{H}_{A,t} = \Big\{ T(x),\ T(x)(Ax+t)_k,\ T(x)(Ax+t)_k(Ax+t)_l \quad \Big| \quad 1 \leqslant k \leqslant l \leqslant n \Big\}$$

is linearly independent. In order to simplify the notation, we take an arbitrary order of $\mathcal{H}_{A,t}$ and write $h_i(x)$ for its elements, $i = 1, \ldots, N$. Now, let $Q \subset \mathbb{R}^n$ be a cube containing $\operatorname{supp}(T)$. Then, define a set of orthonormal polynomials $(b_j(x))_{j\in\mathbb{N}}$ with respect to the L^2-norm on Q, which is in ascending order with respect to the degree. Moreover, define $P_m := \operatorname{span}(b_1(x), \ldots, b_m(x))$. Thus, the best L^2 approximation of $h_i(x) \in \mathcal{H}_{A,t}$ with polynomials up to a certain degree is given by

$$h_{i,m} = \sum_{j=1}^{m} a_{i,j} b_j(x)$$

with

$$a_{i,j} = \int_Q b_j(x) h_i(x) dx.$$

Now, consider the matrix

$$\tilde{M} = \begin{pmatrix} a_{1,1} & \cdots & a_{1,m} \\ \vdots & & \vdots \\ a_{N,1} & \cdots & a_{N,m} \end{pmatrix}.$$

According to Lemma 5.3 the fact that $\mathcal{H}_{A,t}$ is linearly independent yields the existence of a real number $m \in \mathbb{N}$ such that all rows of $\tilde{M}$ are also linear independent. Due to the fact that $\langle\, \mathcal{H}_{A,t} \,\rangle = \langle\, \mathcal{H}_{\tilde{A},\tilde{t}} \,\rangle$ for all $(A,t), (\tilde{A},\tilde{t}) \in G$, this number m is independent from the choice of (A,t). Let $\tilde{m}_1, \ldots, \tilde{m}_N$ denote the rows of $\tilde{M}$ and let $\hat{p} \in \mathbb{R}^n$ denote the representation of a polynomial $p \in P_m$ with respect to the basis $b_1(x), \ldots, b_m(x)$. Then, the formulas in (2.27) become

$$\alpha_i = m_1 \cdot \widehat{\frac{\partial R}{\partial x_i}}(Ax+t),\ \beta_i = m_2 \cdot \widehat{\frac{\partial R}{\partial x_i}}(Ax+t), \ldots,\ \delta_{n,n,i,j} = m_N \cdot \widehat{\frac{\partial^2 R}{\partial x_i \partial x_j}}(Ax+t)$$

if $\frac{\partial R}{\partial x_i}(Ax+t), \frac{\partial^2 R}{\partial x_i^2}(Ax+t), \frac{\partial^2 R}{\partial x_i x_j}(Ax+t) \in P_m$ for all $i, j = 1, \ldots, n$. Since $\tilde{m}_1, \ldots, \tilde{m}_N$ are linear independent, we get that the map

$$\mathcal{P}_{n,m} \to \mathbb{R}^N, \quad R \mapsto (\alpha, \beta, \gamma, \delta, \epsilon)$$

is surjective for each $(A,t) \in G$, if m is large enough. Hence, using the formulas (2.28) and (2.29) for $G = SE(n)$, and (2.32)-(2.36) for $G = SA(n)$ respectively, one can easily verify that $j^2_{(A,t)}\hat{\Phi}\big|_{P_m}$ is surjective for each chosen $(A,t) \in G$.
Now, let $R \in C^3(\mathbb{R}^n, \mathbb{R})$ be arbitrarily chosen. Like in the case of the cost function Ψ before, we apply the *Parametric Transversality Theorem* to the map

$$\begin{aligned} F^{\mathrm{ev}} &: [R + P_m] \times G \to J^2(G, \mathbb{R}) \\ (R+p, A, t) &\mapsto j^2_{(A,t)}\hat{\Phi}(R+p). \end{aligned}$$

Since $F^{\mathrm{ev}}(\ .\ , A, t)$ is surjective for all $(A,t) \in G$, the map $F^{\mathrm{ev}}(R+p,\ .\ )$ is transverse to $\mathfrak{U}_\mathfrak{i}$ for a dense subset of P_m. Therefore, $A_{\Phi,i}$ is dense in $C^3_W(\mathbb{R}^n, \mathbb{R})$ which completes the proof in the case of the cost function Φ. □

Theorem 5.4 yields a condition for the sequence of sample-points $(x_i)_{i=1}^N$ such that the cost function $\hat{\Psi}(f) : SE(n) \to \mathbb{R}$ has generically no degenerate critical points. However, one would like to know if this condition is typically satisfied, if we choose the sample-points randomly. This question will be answered in the following lemma.

Lemma 5.5

Set $k = \frac{1}{2}(n+1)(n+2)$, $n \in \mathbb{N}$. The set

$$M := \Big\{(x_1, \dots, x_k) \in \mathbb{R}^{n\times k} \;\Big|\; x_1, \dots, x_k \textit{ do not lie on a quadric hypersurface}\Big\}$$

is open and dense in $\mathbb{R}^{n\times k}$.

Proof
Let V and φ be defined like in the proof of the previous theorem. Then, $x \in \mathbb{R}^n$ lies on the quadric hyperface, defined by $(\alpha, a, A) \in V$ if, and only if

$$\langle(\alpha, a, A), \varphi(x)\rangle = 0$$

is satisfied. Therefore, using

$$\chi : \mathbb{R}^{n\times k} \to \mathbb{R}, \quad \chi(x_1, \dots, x_k) = \det\left(\varphi(x_1), \dots, \varphi(x_k)\right)$$

we get $M = \mathbb{R}^{n\times k} \setminus \chi^{-}(0)$.
Since φ is a polynomial in each component, the same holds for χ. Consequently, $\chi^{-}(0)$ is an affine variety and we only have to show that M is not empty to complete the proof. For this purpose, we consider the points

$$(x_1, \dots, x_k) = (0, e_1, \dots, e_n, -e_1, \dots, -e_n, e_1 + e_2, \dots, e_{n-1} + e_n)$$

where the e_i's form a basis of the $\mathbb{R}^n$. We will show that $\operatorname{span}(\varphi(x_1), \dots, \varphi(x_k)) = V$, which is equivalent to $(x_1, \dots, x_k) \in M$. Firstly, we have $\varphi(0) = (1, 0, 0)$ and therefore $\operatorname{span}((1,0,0)) \subset \operatorname{span}(\varphi(x_1), \dots, \varphi(x_k))$. Moreover, using $\frac{1}{2}\left[\varphi(e_i) - \varphi(-e_i)\right] = (0, e_i, 0)$ we get $\operatorname{span}((0,x,0) \mid x \in \mathbb{R}^n) \subset \operatorname{span}(\varphi(x_1), \dots, \varphi(x_k))$. Finally, we have for $i < j$

$$\varphi(e_i + e_j) - \varphi(e_i) - \varphi(e_j) + \varphi(0) = (0, 0, e_i e_j^\top + e_j e_i^\top)$$

and

$$\frac{1}{2}\left[\varphi(e_i)+\varphi(-e_i)\right]-\varphi(0)=(0,0,e_ie_i^\top).$$

Therefore, we conclude that $\operatorname{span}((0,0,xx^\top)\mid x\in\mathbb{R}^n)\subset\operatorname{span}(\varphi(x_1),\dots,\varphi(x_k))$. To sum up, we have $\operatorname{span}(\varphi(x_1),\dots,\varphi(x_k))=V$ which completes the proof. □

In the QMC-Newton algorithm described, the sequence $(x_i)_{i=1}^N$ is not generated randomly, but by a deterministic method like the Halton-sequence mentioned. We will not discuss how many iterations of the Halton-sequence are at the most necessary such that the condition in part *b)* of Theorem 5.4 is satisfied, since in applications we will need a large number of iterations of the Halton-sequence to be robust against noise in the images. However, the Halton-sequence is dense in the region of interest. Therefore, there is a number $N\in\mathbb{N}$ such that not all points of $(x_i)_{i=1}^N$ lie on a quadric hyperface. With the following lemma at hand, we conclude that the condition in part *b)* of Theorem 5.4 is also satisfied.

Lemma 5.6
Let $(x_i)_{i=1}^N\subset\mathbb{R}^n$ be a finite sequence of $N\in\mathbb{N}$ elements which does not lie on a quadric hypersurface. Then, there is a subsequence $\left(x_{i_j}\right)_{j=1}^k$ of $k=\frac{1}{2}(n+1)(n+2)$ elements which also does not lie on a quadric hypersurface.

Proof
Following the proof of the previous theorem, we get that a given set of points $\{x_i\}_{i=1}^N\subset\mathbb{R}^n$ does not lie on a quadric hyperface if, and only if

$$\operatorname{span}(\varphi(x_1),\dots,\varphi(x_N))=\mathbb{R}\times\mathbb{R}^n\times\operatorname{Sym}(n)=:V.$$

Suppose that this equation is satisfied. Since $\dim V=k$, we can find k linearly independent vectors in the set $\{\varphi(x_1),\dots,\varphi(x_N)\}$. Since these k vectors also span the whole vector space, the lemma is proven. □

We want to note that the condition for the image T in part *b)* of Theorem 5.4 is not very restrictive. Since the region of interest Q is typically bounded, there is a minimum value $m_T\in\mathbb{R}$ of T. Hence, we can consider the registration-problem

$$\min_{(A,t)\in G}\int_Q\left(\tilde{T}(x)-\tilde{R}(Ax+t)\right)^2dx$$

with $\tilde{T}(x)=T(x)+m_T+1$ and $\tilde{R}(x)=R(x)+m_T+1$ instead of (5.1). For this new registration problem we have $\tilde{T}(x)\neq 0$ for all $x\in Q$.
With this result we can guarantee with probability one that the QMC-algorithm locally converges quadratically to a local maximum in the case of non-artificial generated images. However, Theorem 5.4 will not yield any result in the ideal case, in which the reference and template image are identical. Moreover, the case in which both images are elements of a spline function space, or smoothed by a Gaussian kernel, is not applicable to this theorem. Even though these restrictions are more relevant to applications, we believe that a further discussion in this direction would be beyond the scope of this work.

Finally, we want to mention that a similar problem arises in the pose-estimation task (cf. Section 3.1). The cost function we want to maximize here on $SE(3)$ is

$$\Phi : SE(3) \to \mathbb{R}, \quad \Phi(M) := \int\limits_{\mathbb{R}^3} p_{M\cdot\mathcal{X}}(z) p_{\mathcal{Y}}(z) dz, \tag{5.16}$$

where $\mathcal{X} = (x_1, \dots, x_m), \mathcal{Y} = (y_1, \dots, y_m) \in \mathbb{R}^{3\times m}$ are given samples and $p_{M\cdot\mathcal{X}}, p_{\mathcal{Y}}$ are the corresponding density-functions. By invariance of the integral under volume-preserving maps, the cost function in (5.16) differs from the least squares term in (3.4) only in a constant. Thus, maximization of Φ is equivalent the minimization (3.4). Using the approximation (3.3) of the densities we get

$$\begin{aligned} \Phi(M) &\approx (\pi\sigma^2)^{-n} \sum_{i,j=1}^{m} \int\limits_{\mathbb{R}^3} \exp\left(-\frac{1}{2\sigma^2}\|z - M\cdot x_i\|^2\right) \exp\left(-\frac{1}{2\sigma^2}\|z - y_j\|^2\right) dz \\ &= (2\pi\sigma^2)^{-n/2} \sum_{i,j=1}^{m} \exp\left(-\frac{1}{4\sigma^2}\|M\cdot x_i - y_j\|^2\right). \end{aligned}$$

This is a cost function of the form (2.17). Hence, we can apply the algorithm in Table 2.1. Moreover, Theorem 2.3 yields that this algorithm converges locally quadratically around each nondegenerate critical point. Thus, the question which arises is whether the set

$$\left\{\mathcal{Y} \in \mathbb{R}^{3\times m} \mid \Phi \text{ has no degenerate critical points}\right\}$$

is generic in $\mathbb{R}^{3\times m}$ or not.

Up to the best of our knowledge, this problem is still unsolved. Until now, we were not able to prove a similar statement to Theorem 5.4 for the cost function Φ in (5.16). The main problem might be that Φ in (5.16) depends non-linear on $\mathcal{Y}$, whereas the cost function Ψ in (5.3) is linear in the image R.

5.2 Registration on Spline Function Spaces

We already pointed out in the Sections 5.1.1 and 3.4 that image smoothing is often an essential step for image registration algorithm. If we consider an image as a real-valued function with compact support, smoothing can be regarded as a projection to a suitable space of smooth functions, cf. Section 3.2, such as e.g. spaces of splines $\mathcal{S}_\lambda^k$. We will show that this interpretation, compared to standard smoothing methods like Gaussian filters, has the advantage that we can exploit the reduction of information in the previously discussed Newton method while maintaining a high degree of accuracy. Yet, a difficulty with this approach is that the space of splines is usually not invariant under rotations, as it is the case for tensor product splines. Thus, a group action $G \times \mathbb{R}^n \to \mathbb{R}^n$ can be extended to a group action on $C(\mathbb{R}^n, \mathbb{R})$ but not necessarily to a group action on $\mathcal{S}_\lambda^k$. Nevertheless, in this section we present a registration method which exploits the image representations of the form $R, T \in \mathcal{S}_\lambda^k$.

5.2.1 Monomodal Registration

We begin with a reformulation of the registration problem. Let $\mathcal{S}_\lambda^2$ denote the spline function space as defined in (3.10). For two given images $R \in \mathcal{S}_\lambda^2$ and $T \in \mathcal{S}_\lambda^1$ with coefficients $c^R, c^T \in l_2$ the cost function (5.2) becomes

$$\Phi(A,t) = \sum_{r,s\in\frac{1}{\lambda}\mathbb{Z}^n} c_s^R c_r^T \int\limits_{\mathbb{R}^n} B_{s,\lambda}^2(Ax+t)B_{r,\lambda}^1(x)dx,$$

with $B_{r,\lambda}^k(x) := B^k\left(\frac{x}{\lambda} - \lambda r\right)$. Since $B_{r,\lambda}^k(x)$ is a translation of $B_{0,\lambda}^k$, we obtain

$$\begin{aligned} \Phi(A,t) &= \sum_{s,r\in\frac{1}{\lambda}\mathbb{Z}^n} c_s^R c_r^T \int\limits_{\mathbb{R}^n} B_{0,\lambda}^1(x)B_{0,\lambda}^2(A(x+s)+t-r)dx \\ &\approx \sum_{s,r\in\frac{1}{\lambda}\mathbb{Z}^n} c_s^R c_r^T \int\limits_{\mathbb{R}^n} B_{0,\lambda}^1(x)B_{0,\lambda}^2(x-A^{-1}(r-t)+s)dx. \end{aligned} \tag{5.17}$$

The approximation in the last line needs some explanations: Of course, the tensor products (3.9) of B-spines are not invariant under rotation. Assuming they are, we obtain approximations, by rotating the argument of the second spline around the barycenter $t-r$ in such a way that the argument becomes a simple translation in the direction $A^{-1}(r-t)-s$. In the cases where A is close to the identity (which is true for most of the medical image problems) the approximation error tends to zero. Our examples in Section 5.3 will show that this simplification does hardly influence the solution of the optimization problem.
Therefore, the convolution of two splines

$$\begin{aligned} F(s) &:= \int\limits_{\mathbb{R}^n} B_{0,\lambda}^1(x)B_{0,\lambda}^2(x-s)dx, \\ &= B_{0,\lambda}^4(s_1)\cdot\ldots\cdot B_{0,\lambda}^4(s_n) \end{aligned} \tag{5.18}$$

is related to the optimization problem. Note that $F(s)$ as the tensor product of B-splines of order four has already been well studied in the literature [71]. For us, the most important fact of such B-splines is that they are three times continuously differentiable, which makes (5.17) applicable to the Newton-algorithm (2.12). Furthermore, $F(s)$ is piecewise polynomial with degree four, so the values and the derivatives of the function can be calculated very quickly and without additional numerical approximations. Therefore, the modified image registration problem becomes

$$\max_{M\in G} \sum_{s,r\in\frac{1}{\lambda}\mathbb{Z}^n} c_s^R c_r^T F(PM^{-1}\bar{r}-s), \quad P = (I_n\ 0) \in \mathbb{R}^{n\times(n+1)}, \quad M = \begin{pmatrix} A & t \\ 0 & 1 \end{pmatrix}.$$

To calculate the Newton-step for this optimization problem, we apply the setting of Section 2.2 to the cost function

$$\Gamma : G \to \mathbb{R}, \quad \Gamma(M) := \sum_{s,r\in\frac{1}{\lambda}\mathbb{Z}^n} c_s^R c_r^T F(PM\bar{r}-s). \tag{5.19}$$

In order to verify that Γ is again of the form $M \mapsto \varphi(f \circ \varrho_M)$ as it is required in (2.15), we define $\varphi : \mathcal{D}(\mathbb{R}^n) \to \mathbb{R}$ and $f : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$ via

$$\varphi(h) = \sum_{r \in \frac{1}{\lambda}\mathbb{Z}^n} h(r) \quad \text{and} \quad f(x,y) = \sum_{s \in \frac{1}{\lambda}\mathbb{Z}^n} c_s^R c^T(y) F(x-s).$$

Here, $c^T(y)$ is an arbitrary function in $C(\mathbb{R}^n, \mathbb{R})$ satisfying $c^T(r) = c_r^T$ for all $r \in \frac{1}{\lambda}\mathbb{Z}^n$.
Using the same calculations as in Section 5.1 we end up with two new **Spline-based Newton Registration-Algorithms** (SB), one is acting on $SE(n)$ and one on $SA(n)$. Both algorithms have almost the same steps as their continuous counterparts, the QMC-Newton on $SE(n)$ and on $SA(n)$ respectively, which is the reason why we do not present them here again. The only difference between the SB- and the QMC-Newton algorithms is the calculation of the coefficients $\alpha, \beta, \gamma, \delta$ and ϵ which is done in Step 2 for each algorithm. (See Table 5.1 or Table 5.2)
In the case of the spline-based algorithms, these coefficients have the following form:

$$\begin{aligned}
\alpha_i &= \sum_{s,r \in \frac{1}{\lambda}\mathbb{Z}^n} c_s^R c_r^T \frac{\partial F}{\partial x_i}(Ar + t - s), \\
\beta_{i,j} &= \sum_{s,r \in \frac{1}{\lambda}\mathbb{Z}^n} c_s^R c_r^T \frac{\partial F}{\partial x_i}(Ar + t - s)(Ar + t)_j, \\
\gamma_{i,j,k} &= \sum_{s,r \in \frac{1}{\lambda}\mathbb{Z}^n} c_s^R c_r^T (Ar + t)_i \frac{\partial^2 F}{\partial x_j \partial x_k}(Ar + t - s), \qquad (5.20) \\
\delta_{i,j,k,l} &= \sum_{s,r \in \frac{1}{\lambda}\mathbb{Z}^n} c_s^R c_r^T (Ar + t)_i (Ar + t)_j \frac{\partial^2 F}{\partial x_k \partial x_l}(Ar + t - s), \\
\epsilon_{i,j} &= \sum_{r,s \in \frac{1}{\lambda}\mathbb{Z}^n} c_s^R c_r^T \frac{\partial^2 F}{\partial x_i \partial x_j}(Ar + t - s).
\end{aligned}$$

One may argue that the SB-Newton algorithm presented above requires a high degree of differentiability of the images, as it is assumed in the QMC-Newton algorithm. Just as in Theorem 5.1 the only condition on the images R and T to show the local quadratic convergence of the SB-Newton algorithms (Table 5.1 or Table 5.2) is that the cost function Γ in (5.19) is in $C^3(G, \mathbb{R})$. However, in this case, the construction of Φ via splines reduces the requirements of differentiability on R and T: For $R \in \mathcal{S}_\lambda^p$ and $T \in \mathcal{S}_\lambda^q$ this condition is satisfied if, and only if $p + q \geqslant 3$. Thus, for $p = 2$ and $q = 1$ we have $R \in C^1(\mathbb{R}^n, \mathbb{R})$, $T \in C(\mathbb{R}^n, \mathbb{R})$ and get the local quadratic convergence of the SB-Newton algorithm in the same way as was shown in the QMC-Newton algorithms (where $R \in C^3(\mathbb{R}^n, \mathbb{R})$ is needed). This fact is quite favorable in image registration: Since images are usually disturbed by noise, the evaluation of higher derivatives of the images seems to be sensless. Therefore, one would like to choose p and q as small as possible. Moreover, the SB-Newton algorithms need no evaluations of the derivatives of R or T. They work directly on the given coefficients, given by the spline representation of the images.
We end this subsection with an analogon of Theorem 5.4 for the cost function $\Gamma : G \to \mathbb{R}$ as defined in (5.19) for $G = SE(n)$ or $G = SA(n)$.

Theorem 5.7
Let $p, q \in \mathbb{N}$ with $p + q \geq 3$. Suppose that all coefficients c_r^T of the template image

$$T(x) = \sum_{r \in \frac{1}{\lambda}\mathbb{Z}^n} c_r^T B_{r,\lambda}^q(x) \in \mathcal{S}_\lambda^q$$

are not zero. Moreover, suppose that the series

$$\sum_{r \in \frac{1}{\lambda}\mathbb{Z}^n} c_r^T r r^\top$$

converge absolutely. Then, the cost functions Γ in (5.19) has no degenerate critical points, for an open and dense set of reference images $R \in \mathcal{S}_\lambda^p$ with respect to the ℓ_2-norm.

Proof
Since the image $T \in \mathcal{S}_\lambda^q$ is fixed, we study the influence of the image $R \in \mathcal{S}_\lambda^p$ to the cost function Γ. Similar to in the proof of the previous Theorem 5.4, we introduce the linear map $\hat{\Gamma} : \mathcal{S}_\lambda^p \to C^3(G, \mathbb{R})$ defined by

$$\hat{\Gamma}\left(\sum_{s \in \frac{1}{\lambda}\mathbb{Z}^n} c_s^R B_{r,\lambda}^p(x)\right)(A,t) := \sum_{r,s \in \frac{1}{\lambda}\mathbb{Z}^n} c_r^T c_s^R B_\lambda^{p+q+1}(Ar + t - s)$$

and consider the set

$$A_{\Gamma,i} := \left\{ R \in \mathcal{S}_\lambda^p \mid j^2\hat{\Gamma}(R) \pitchfork \mathfrak{U}_i \right\},$$

where $\mathfrak{U}_i$ is defined in (5.7). We have to show that $A_{\Gamma,i}$ is open and dense in $\mathcal{S}_\lambda^p$ with respect to the ℓ_2-norm.
First, we define for a given sequence $\left\{c_s^R\right\}_{s \in \frac{1}{\lambda}\mathbb{Z}^n}$ the function

$$H(x) := \sum_{s \in \frac{1}{\lambda}\mathbb{Z}^n} c_s^R B_{s,\lambda}^{p+q+1}(x) \in \mathcal{S}_\lambda^{p+q+1}.$$

Thus, we can rewrite (5.20) in terms of the ℓ_2-scalar product

$$\begin{aligned}
\alpha_i &= \left\langle (c_r^T)_r, \left(\frac{\partial H}{\partial x_i}(Ar+t)\right)_r \right\rangle, \\
\beta_{i,1} &= \left\langle (c_r^T(Ar+t)_1)_r, \left(\frac{\partial H}{\partial x_i}(Ar+t)\right)_r \right\rangle, \\
&\vdots \\
\delta_{n,n,i,j} &= \left\langle (c_r^T)_r, \left(\frac{\partial^2 H}{\partial x_i \partial x_j}(Ar+t)\right)_r \right\rangle.
\end{aligned}$$

Let $(A,t) \in G$ be fixed. We can find $k = \frac{1}{2}(n+1)(n+2)$ points r_l in the grid $\frac{1}{\lambda}\mathbb{Z}^n$, which do not lie on a quadric hypersurface. Without loss of generality we can assume that the map

$(c_s^R)_s \mapsto (j^2H(Ar_1+t),\dots,j^2H(Ar_k+t))$ is surjective. Following the same argumentation from (5.11) to (5.13) yields that the map $(c_s^R)_s \mapsto (\alpha,\dots,\epsilon)$ is surjective for each $(A,t)\in G$. Hence, using the formulas (2.28) and (2.29) for $G=SE(n)$, and (2.32)-(2.36) for $G=SA(n)$ respectively, one can easily verify that $j^2_{(A,t)}\hat{\gamma}$ is surjective for each chosen $(A,t)\in G$. Therefore, we can apply the *Parametric Transversality Theorem* to the map

$$\begin{aligned} F^{\mathrm{ev}} : \mathcal{S}^2_\lambda \times G &\to J^2(G,\mathbb{R}) \\ (R,A,t) &\mapsto j^2_{(A,t)}\hat{\Gamma}(R). \end{aligned}$$

Since $F^{\mathrm{ev}}(\ .\ ,A,t)$ is surjective for all $(A,t)\in G$, the map $F^{\mathrm{ev}}(R,\ .\ )$ is transverse to $\mathfrak{U}_\mathfrak{i}$ for a dense and open subset. Therefore, $A_{\Gamma,i}$ is dense and open in $\mathcal{S}^p_\lambda$. □

5.2.2 Multimodal Registration

Following the approach of Viola [86], the calculation of the mutual information of two images R and T is done in two steps. First, one has to calculate $R(x_l)-T(x_l)$ for a couple of supporting points $\{x_l\}_{l\in I}\subset\mathbb{R}^n$ to give an estimation of the joint density $\rho_{R,T}$. Afterwards, the integral $\int \rho_{R,T}\log\rho_{R,T}$ is approximated numerically - usually by a Monte Carlo method. Let two images $R\in\mathcal{S}^2_\lambda$ and $T\in\mathcal{S}^1_\lambda$ be given. We estimate the joint density by using the coefficients c_u^R and c_u^T of R and T:

$$\rho_{R,T}(x) \approx \sum_{u\in\frac{1}{\lambda}\mathbb{Z}^n} \sigma B^3\left(\frac{x-(c_u^R-c_u^T)}{\sigma}\right). \tag{5.21}$$

Here, $\sigma\in\mathbb{R}$ controls the approximation error of $\rho_{R,T}$ (cf. [68] p. 88-95), and B^3 denotes the cubic B-Spline in one dimension (cf. Fig. 3.3). In image registration we are especially interested in the case in which T is deformed by an element $M\in G$. Therefore, we have to replace c_u^T in (5.21) by the coefficients $\hat{c}_u^T$ of $T(PM\bar{x})$; cf. (2.9) for the notation of P and $\bar{x}$. However, in general, $T(PM\bar{x})$ is not in $\mathcal{S}^2_\lambda$ for an arbitrary $M\in G$ and an additional projection step is needed to get $\hat{c}_u^T$: Let $P^k_\lambda : L^2(\mathbb{R}^n,\mathbb{R})\to\mathcal{S}^k_\lambda$ denote the projection on the spline function space. Then, we get the representation

$$P^k_\lambda(f) = \sum_{u,r\in\frac{1}{\lambda}\mathbb{Z}^n} \gamma_{u,r}\left\langle f, B^k\left(\frac{x}{\lambda}-r\right)\right\rangle_{L^2} B^k\left(\frac{x}{\lambda}-u\right)$$

with suitable $\gamma_{u,r}\in\mathbb{R}$. The weights $\gamma_{u,r}$ are necessary, since the translations of the splines do not form an orthonormal system. They can be calculated explicitly and independently of the given images R and T. We make the same simplification as in the previous section and get

$$\begin{aligned} \hat{c}_u^T\begin{pmatrix} A & t \\ 0 & 1 \end{pmatrix} &= \sum_{s,r\in\frac{1}{\lambda}\mathbb{Z}^n} c_s^T\gamma_{u,r}\int\limits_{\mathbb{R}^n} B^1\left(\frac{Ax+t}{\lambda}-r\right)B^2\left(\frac{x}{\lambda}-s\right)dx \\ &\approx \sum_{s,r\in\frac{1}{\lambda}\mathbb{Z}^n} c_s^T\gamma_{u,r}F(A^{-1}(r-t)+s). \end{aligned} \tag{5.22}$$

Substituting c_u^T in (5.21) by (5.22) provides an explicit formula for the cost function $\Gamma : G \to \mathbb{R}$:

$$\Gamma(M) := \int_{\mathbb{R}} \varrho \left(\sum_{u \in \frac{1}{\lambda}\mathbb{Z}^n} \sigma B^3 \left(\frac{x - (c_u^R - \hat{c}_u^T(M))}{\sigma} \right) \right) dx, \quad \varrho(x) := -x \log x. \tag{5.23}$$

In the sequel, we use the Simpson-rule to approximate the entropy $\int \rho_{R,T} \log \rho_{R,T}$. Following [86], we have to search for the maximum of Γ. We have to take into account that Γ is not of the form (2.15), which was discussed in Section 2.2.1. However, (5.23) is an three times differentiable function and we can apply the same techniques as in Section 2.2.1. We get:

$$\Gamma(M) = \frac{1}{p} \sum_{k \in \mathbb{Z}} \varrho \left(\sigma \sum_{u \in \frac{1}{\lambda}\mathbb{Z}^n} B^3 \left(\frac{k}{\sigma p} - \frac{1}{\sigma}(c_u^R - \hat{c}_u^T(M)) \right) \right) \tag{5.24}$$

with

$$\begin{aligned} \nabla(\Gamma \circ \mu_M)(0) = \frac{1}{p} \sum_{k \in \mathbb{Z}} \Bigg[& \varrho' \left(\sigma \sum_{u \in \frac{1}{\lambda}\mathbb{Z}^n} B^3 \left(\frac{k}{\sigma p} - \frac{1}{\sigma}(c_u^R - \hat{c}_u^T(M)) \right) \right) \\ & \cdot \sum_{u \in \frac{1}{\lambda}\mathbb{Z}^n} B^{3\prime} \left(\frac{k}{\sigma p} - \frac{1}{\sigma}(c_u^R - \hat{c}_u^T(M)) \right) \nabla(\hat{c}_u^T \circ \mu_M)(0) \Bigg] \end{aligned}$$

and

$$\begin{aligned} & \mathrm{Hess}_{\Gamma \circ \mu_M}(0)(\Omega, v) \\ & = \frac{1}{p} \sum_{k \in \mathbb{Z}} \Bigg[\varrho' \left(\sigma \sum_{u \in \frac{1}{\lambda}\mathbb{Z}^n} B^3 \left(\frac{k}{\sigma p} - \frac{1}{\sigma}(c_u^R - \hat{c}_u^T(M)) \right) \right) \\ & \cdot \sum_{u \in \frac{1}{\lambda}\mathbb{Z}^n} B^{3\prime} \left(\frac{k}{\sigma p} - \frac{1}{\sigma}(c_u^R - \hat{c}_u^T(M)) \right) \mathrm{Hess}\ _{\hat{c}_u^T \circ \mu_M}(0)(\Omega, v) \Bigg] \\ & + \frac{1}{p} \sum_{k \in \mathbb{Z}} \Bigg[\varrho' \left(\sigma \sum_{u \in \frac{1}{\lambda}\mathbb{Z}^n} B^3 \left(\frac{k}{\sigma p} - \frac{1}{\sigma}(c_u^R - \hat{c}_u^T(M)) \right) \right) \\ & \cdot \sum_{u \in \frac{1}{\lambda}\mathbb{Z}^n} B^{3\prime\prime} \left(\frac{k}{\sigma p} - \frac{1}{\sigma}(c_u^R - \hat{c}_u^T(M)) \right) \left\langle \nabla(\hat{c}_u^T \circ \mu_M)(0), (\Omega, v) \right\rangle \nabla(\hat{c}_u^g \circ \mu_M)(0) \Bigg] \\ & + \frac{1}{p} \sum_{k \in \mathbb{Z}} \Bigg[\varrho'' \left(\sigma \sum_{u \in \frac{1}{\lambda}\mathbb{Z}^n} B^3 \left(\frac{k}{\sigma p} - \frac{1}{\sigma}(c_u^R - \hat{c}_u^T(M)) \right) \right) \end{aligned}$$

$$\cdot \sum_{u\in\frac{1}{\lambda}\mathbb{Z}^n} \left(B^{3'}\left(\frac{k}{\sigma p} - \frac{1}{\sigma}(c_u^R - \hat{c}_u^T(M))\right)\right)^2 \left\langle \nabla(\hat{c}_u^T \circ \mu_M)(0), (\Omega, v)\right\rangle \nabla(\hat{c}_u^T \circ \mu_M)(0)\Bigg] .$$

Where $B^{3'}$ and $B^{3''}$ denote the first and second derivative of B^3 (and the same holds for ϱ' and ϱ'').

Before writing the corresponding Newton step in components, we need some substitutions:

$$\begin{aligned}
b_{u,k} =& B^3\left(\frac{k}{\sigma p} - \frac{1}{\sigma}(c_u^R - \hat{c}_u^T(M))\right)\\
{b_{u,k}}' =& B^{3'}\left(\frac{k}{\sigma p} - \frac{1}{\sigma}(c_u^R - \hat{c}_u^T(M))\right)\\
{b_{u,k}}'' =& B^{3''}\left(\frac{k}{\sigma p} - \frac{1}{\sigma}(c_u^R - \hat{c}_u^T(M))\right)\\
\varrho_k' =& \varrho'\left(\sigma \sum_{u\in\frac{1}{\lambda}\mathbb{Z}^n} B^3\left(\frac{k}{\sigma p} - \frac{1}{\sigma}(c_u^R - \hat{c}_u^T(M))\right)\right)\\
\varrho_k'' =& \varrho''\left(\sigma \sum_{u\in\frac{1}{\lambda}\mathbb{Z}^n} B^3\left(\frac{k}{\sigma p} - \frac{1}{\sigma}(c_u^R - \hat{c}_u^T(M))\right)\right)
\end{aligned} \tag{5.25}$$

and

$$\begin{aligned}
\zeta_{r,s,i} =& \sum_{k,u} \left(\varrho_k' {b_{u,k}}'' + \varrho_k'' {b_{u,k}}'^2\right)(\tilde{v}_u)_i(\tilde{\Omega}_u)_{r,s}\\
\eta_{r,i} =& \sum_{k,u} \left(\varrho_k' {b_{u,k}}'' + \varrho_k'' {b_{u,k}}'^2\right)(\tilde{v}_u)_i(\tilde{v}_u)_r\\
\vartheta_{r,s,i,j} =& \sum_{k,u} \left(\varrho_k' {b_{u,k}}'' + \varrho_k'' {b_{u,k}}'^2\right)(\tilde{\Omega}_u)_{i,j}(\tilde{\Omega}_u)_{r,s}\\
\theta_{k,u,i} =& \varrho_k' {b_{u,k}}'(\tilde{v}_u)_i\\
\iota_{k,u,i,j} =& \varrho_k' {b_{u,k}}'(\tilde{\Omega}_u)_{i,j}.
\end{aligned} \tag{5.26}$$

Here, $(\tilde{v}_u, \tilde{\Omega}_u)$ denotes the gradient of the function $\hat{c}_u^T \circ \mu_M$ in zero. Note that $\hat{c}_u^T \circ \mu_M$ is a function of the form (5.19). Hence, the gradient and the Hessian in zero are already calculated in (2.18) and (2.19). Therefore, the i^{th} component of the vector part of the Newton equation becomes

$$\sum_{k,u} \varrho_k' {b_{u,k}}' \left(\mathrm{Hess}_{\hat{c}_u^T\circ\mu_M}(0)(\Omega, v)\right)_{vector,i} + \sum_{r,s} \zeta_{r,s,i}\Omega_{r,s} + \sum_r \eta_{r,i} v_r = -\sum_{k,u} \theta_{k,u,i}$$

and the (i, j) component of the corresponding matrix-part is

$$\sum_{k,u} \varrho_k' {b_{u,k}}' \left(\mathrm{Hess}_{\hat{c}_u^T\circ\mu_M}(0)(\Omega, v)\right)_{matrix,i,j} + \sum_{r,s} \vartheta_{r,s,i,j}\Omega_{r,s} + \sum_r \zeta_{i,j,r} v_r = -\sum_{k,u} \iota_{k,u,i,j}.$$

Table 5.3: Mutual Information-based Registration-Algorithm on SE(n)

Step 1.
Pick an initial guess $M_0 \in SE(n)$ and set $m = 0$.

Step 2.
Calculate all coefficients in (5.25), (5.26) and (5.27).

Step 3.
Solve the linear system described in (5.28) and (5.29) with the unknowns v_i, $1 \leqslant i \leqslant n$ and $\Omega_{i,j}$, $1 \leqslant i < j \leqslant n$.

Step 4.
Construct the $n \times n$ matrix Ω with entries $\Omega_{i,j}$. If $j > i$ use the solution of Step 3 or else set

$$\Omega_{i,j} = \begin{cases} -\Omega_{j,i} & \text{for } j < i \\ 0 & \text{for } j = i. \end{cases}$$

Compute

$$M_{m+1} := \nu^{QR}_{M_m}(\Omega, v),$$

where ν^{QR} is defined in (2.11).

Step 5.
Set $m = m + 1$ and goto Step 2.

To expand these two equations in full detail, we make the same calculations as in the algorithms before and set

$$\begin{aligned}
\hat{\alpha}_i &= \sum_{u,v} \varrho'_v b_{u,v}{}' \sum_{s,r \in \frac{1}{\lambda}\mathbb{Z}^n} c_s^T \gamma_{u,r} \frac{\partial F}{\partial x_i}(Ar + t - s), \\
\hat{\beta}_{i,j} &= \sum_{u,v} \varrho'_v b_{u,v}{}' \sum_{s,r \in \frac{1}{\lambda}\mathbb{Z}^n} c_s^T \gamma_{u,r} \frac{\partial F}{\partial x_i}(Ar + t - s)(Ar + t)_j, \\
\hat{\gamma}_{i,j,k} &= \sum_{u,v} \varrho'_v b_{u,v}{}' \sum_{s,r \in \frac{1}{\lambda}\mathbb{Z}^n} c_s^T \gamma_{u,r} (Ar + t)_i \frac{\partial^2 F}{\partial x_j \partial x_k}(Ar + t - s), \qquad (5.27) \\
\hat{\delta}_{i,j,k,l} &= \sum_{u,v} \varrho'_v b_{u,v}{}' \sum_{s,r \in \frac{1}{\lambda}\mathbb{Z}^n} c_s^T \gamma_{u,r} (Ar + t)_i (Ar + t)_j \frac{\partial^2 F}{\partial x_l \partial x_k}(Ar + t - s), \\
\hat{\epsilon}_{i,j} &= \sum_{u,v} \varrho'_v b_{u,v}{}' \sum_{r,s \in \frac{1}{\lambda}\mathbb{Z}^n} c_s^T \gamma_{u,r} \frac{\partial^2 F}{\partial x_i \partial x_j}(Ar + t - s).
\end{aligned}$$

Table 5.4: Mutual Information-based Registration-Algorithm on SA(n)

Step 1.
Pick an initial guess $M_0 \in SA(n)$ and set $m = 0$.

Step 2.
Calculate all coefficients in (5.25), (5.26) and (5.27).

Step 3.
Solve the linear system described in (5.30) - (5.34) with the unknowns v_i, $1 \leqslant i \leqslant n$ and $\Omega_{i,j}$, $1 \leqslant i,j \leqslant n$ and $(i,j) \neq (n,n)$.

Step 4.
Construct the $n \times n$ matrix Ω with entries $\Omega_{i,j}$. If $(j,i) \neq (n,n)$ use the solution of Step 3 or else set

$$\Omega_{n,n} = -\sum_{k \neq n} \Omega_{k,k}$$

and compute

$$M_{m+1} := \nu^{QR}_{M_m}(\Omega, v),$$

where ν^{QR} is defined in (2.8).

Step 5.
Set $m = m + 1$ and goto Step 2.

Now we have all necessary instruments to present the Newton equation. In the case of the rigid registration we end up with the following analogue to the linear system (2.28)-(2.29):

Lemma 5.8

Let $(\Omega, v) \in so(n) \times \mathbb{R}^n$ be the Newton-direction for the cost function (5.24) in a certain point $M \in SE(n)$. Then the components $\Omega_{k,l}$,$1 \leqslant k,l \leqslant n$, of Ω and v,k $1 \leqslant k \leqslant n$, of v satisfy

$$\sum_{k>l} (\hat{\gamma}_{l,k,i} - \hat{\gamma}_{k,l,i} + \zeta_{k,l,i} - \zeta_{l,k,i})\Omega_{k,l} + \sum_{k} (\hat{\epsilon}_{i,k} + \eta_{k,i}) v_k = -\sum_{k,u} \vartheta_{k,u,i} \tag{5.28}$$

for all $1 \leqslant i \leqslant n$ and

$$\frac{1}{2}\sum_{k>j}(\hat{\beta}_{i,k} + \hat{\beta}_{k,i})\Omega_{k,j} - \frac{1}{2}\sum_{k<j}(\hat{\beta}_{i,k} + \hat{\beta}_{k,i})\Omega_{j,k} - \frac{1}{2}\sum_{k>i}(\hat{\beta}_{j,k} + \hat{\beta}_{k,j})\Omega_{k,i}$$
$$+ \frac{1}{2}\sum_{k<i}(\hat{\beta}_{j,k} + \hat{\beta}_{k,j})\Omega_{i,k} - \sum_{k>l}(\hat{\delta}_{i,k,l,j} - \hat{\delta}_{j,l,k,i} + \hat{\delta}_{i,l,k,j} - \hat{\delta}_{i,k,l,j} - \vartheta_{k,l,i,j} + \vartheta_{l,k,i,j})\Omega_{k,l}$$

$$-\sum_k (\hat{\gamma}_{j,k,i} - \hat{\gamma}_{i,k,j} - \zeta_{i,j,k}) v_k = -\sum_{k,u} \iota_{k,u,i,j}. \tag{5.29}$$

for all $1 \leqslant i < j \leqslant m$.

We finish this section with the corresponding lemma in the case of volume-preserving transformations.

Lemma 5.9
Let $(\Omega, v) \in sl(n) \times \mathbb{R}^n$ *be the Newton-direction for the cost function (5.24) in a certain point* $M \in SA(n)$*. Then the components* $\Omega_{k,l}$, $1 \leqslant k, l \leqslant n$, $(k,l) \neq (n,n)$ *of* Ω *and* v_k $1 \leqslant k \leqslant n$ *of* v *satisfy for each* $1 \leqslant i \leqslant n$

$$\begin{aligned}\sum_{k \neq l} (\hat{\gamma}_{l,k,i} + \zeta_{k,l,i}) \Omega_{k,l} + \sum_{k \neq n} (\hat{\gamma}_{k,k,i} - \hat{\gamma}_{n,n,i} + \zeta_{k,k,i} - \zeta_{n,n,i}) \Omega_{k,k} \\ + \sum_k (\hat{\epsilon}_{i,k} - \eta_{k,i}) v_k = -\sum_{k,l} \theta_{k,l,i}.\end{aligned} \tag{5.30}$$

and for all $1 \leqslant i, j \leqslant n$, $(i,j) \neq (n,n)$ *the following equations:*

$$\begin{aligned}\sum_k \hat{\beta}_{i,k} \Omega_{j,k} + \sum_k \hat{\beta}_{k,j} \Omega_{k,i} + \sum_{(k,l) \neq (n,n)} (\hat{\delta}_{j,l,k,i} + \vartheta_{l,k,i,j}) \Omega_{l,k} \\ -\sum_{k \neq n} (\hat{\delta}_{j,k,k,i} - \vartheta_{k,k,i,j}) \Omega_{k,k} + \sum_k (\hat{\gamma}_{j,k,i} - \zeta_{i,j,k}) v_k = -\sum_{k,l} \iota_{k,l,i,j}\end{aligned} \tag{5.31}$$

for $i \neq n, \ j \neq n, \ i \neq j$,

$$\begin{aligned}\sum_k \hat{\beta}_{n,k} \Omega_{j,k} + \sum_{k \neq n} \hat{\beta}_{k,j} \Omega_{k,n} + \sum_{(k,l) \neq (n,n)} (\hat{\delta}_{j,l,k,n} + \vartheta_{l,k,n,j}) \Omega_{l,k} \\ -\sum_{k \neq n} (\hat{\delta}_{j,k,k,i} - \vartheta_{k,k,n,j} + \frac{1}{2} \hat{\beta}_{n,j}) \Omega_{k,k} + \sum_k (\hat{\gamma}_{j,k,n} - \zeta_{n,j,k}) v_k = -\sum_{k,l} \iota_{k,l,n,j}\end{aligned} \tag{5.32}$$

for $i = n, \ j \neq n$,

$$\begin{aligned}\sum_{k \neq n} \hat{\beta}_{i,k} \Omega_{n,k} + \sum_k \hat{\beta}_{k,n} \Omega_{k,i} + \sum_{(k,l) \neq (n,n)} (\hat{\delta}_{n,l,k,i} + \vartheta_{l,k,i,n}) \Omega_{l,k} \\ -\sum_{k \neq n} (\hat{\delta}_{n,k,k,i} - \vartheta_{k,k,i,n} + \frac{1}{2} \hat{\beta}_{i,n}) \Omega_{k,k} + \sum_k (\hat{\gamma}_{n,k,i} - \zeta_{i,n,k}) v_k = -\sum_{k,l} \iota_{k,l,i,n}\end{aligned} \tag{5.33}$$

for $j = n, \ i \neq n$ *and*

$$\begin{aligned}\sum_k \hat{\beta}_{i,k}\Omega_{j,k} + \sum_k \hat{\beta}_{k,j}\Omega_{k,i} + \sum_k \left(\hat{\gamma}_{j,k,i} - \zeta_{i,j,k} - \frac{1}{n}\sum_l \hat{\gamma}_{l,l,k}\right) v_k \\ + \sum_{(k,l)\neq(n,n)} \left(\hat{\delta}_{j,l,k,i} + \vartheta_{l,k,i,j} - \frac{1}{2}\left(\hat{\beta}_{l,k} + \sum_m \hat{\delta}_{k,m,l,m}\right)\right)\Omega_{l,k} \\ - \sum_{k\neq n} \left(\hat{\delta}_{j,k,k,i} - \vartheta_{k,k,i,j} - \frac{1}{n}\left(\hat{\beta}_{k,k} + \sum_m \hat{\delta}_{k,m,k,m}\right)\right)\Omega_{k,k} \\ = -\sum_{k,l}\left(\iota_{k,l,i,j} - \frac{1}{n}\sum_m \iota_{k,l,m,m}\right)\end{aligned} \tag{5.34}$$

for $i = j,\ i, j \neq n$.

With this lemmas at hand, one can easily check as before that the algorithms presented in Table 5.3 and Table 5.4 are locally quadratically convergent.

5.3 Experimental Results

In the previous Sections 5.1 and 5.2.1 we developed two different algorithms for the image registration task, the QMC-Newton and the SB-Newton algorithm. In order to demonstrate their use with real medical data, we implement the codes in MATLAB. In this subsection we compare the algorithms with other approaches by considering some numerical examples. For a rigorous experimental examination of the algorithms we have to treat several, slightly different tasks: First, we have to verify the quadratic convergence and the robustness of the algorithms which is done in Subsection 5.3.1. Moreover, we have to compare our manifold approach to classical optimization techniques, which usually treat the problem as a constraint optimization problem on a vector space. And finally, we have to integrate our optimization method into an image registration algorithm.

All computations in this section are performed by using MATLAB R2008a on a 1.9 GHz laptop with 2 GB RAM with an AMD Turion 64X2 TL-64 processor.

5.3.1 Verification of the Convergence Rate

In our first example, we demonstrate the local quadratic convergence rate of the described algorithms. The image at top of the left of Fig. 5.2 shows a 2D cross-section (250 × 250 pixels) of a CT data set. The reference image of our artificial problem is the corresponding spline-approximation with 289 coefficients. The template is identical to the reference. In this example we always start with the initial transformation

$$M_0 = \begin{pmatrix} 1.2 & 0.5 & -87.5 \\ 0 & 0.8333 & 20.8 \\ 0 & 0 & 1 \end{pmatrix}. \tag{5.35}$$

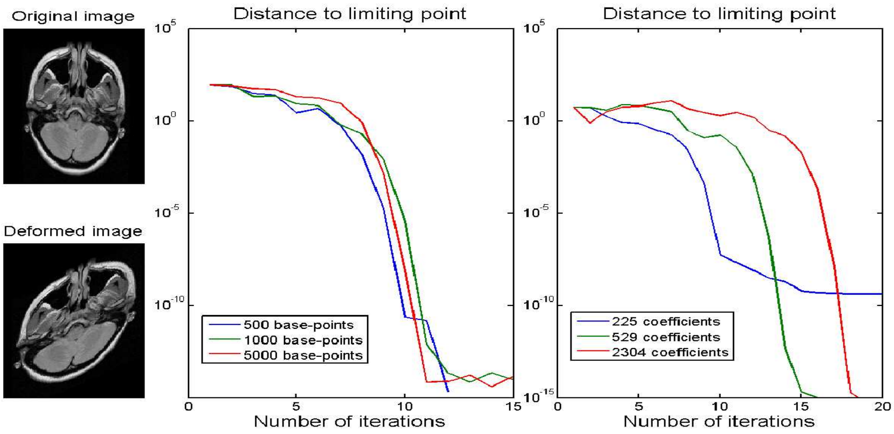


Figure 5.2: Consideration of the convergence speed for a $SA(2)$ problem. Center: QMC-Newton algorithm for several number of sample-points. Right: SB-Newton algorithm for several spline-approximations of the images.

Fig. 5.2 bottom left shows the original image transformed by this matrix. Thus, the optimal solution is $M = I_3$. We take this specific registration task for an examination of the QMC-Newton and the SB-Newton algorithms.

- In the middle of Fig. 5.2, the distance of M_k, $k \in \mathbb{N}_0$ the limit is plotted, using the QMC-Newton algorithm on $SA(2)$. In order to calculate the coefficients (2.27), we take the Quasi-Monte Carlo approximation with the first 500, 1000 and 5000 elements of the Halton-sequence [100]. In all three cases we see a local quadratic convergence and in only 12 steps we achieve an accuracy $< 10^{-12}$. Since the objective function (5.3) is a discretized and approximated version of the correlation-measure (5.2), there is a discrepancy between the exact and the computed transformation. That is, the limit of a particular iteration in Fig. 5.2 is close, but not equal to the identity matrix. For example, for 5000 sample-points, the algorithm ends with

$$M_{20} = \begin{pmatrix} 1.009 & 0.001 & -2.3 \\ 0.008 & 0.991 & 0.24 \\ 0 & 0 & 1 \end{pmatrix},$$

  which is not very close to the optimal solution $M = I_3$. However, this gap would not appear in a discretized version of the “sum of squared difference”.

- The graph on the right of Fig. 5.2 shows the speed of convergence of the spline-based registration algorithm. Here, the reference image is the spline-approximation of the original image with 225, 529 and 2304 coefficients respectively. The template is again identical to the reference and the initial transformation is M_0 from eq. (5.35) for each experiment. In view of the particular registration task, we choose the same order $k = 2$ for the reference R and the template T, i.e. $R, T \in \mathcal{S}^2_\lambda$. Once more, we obtain a local quadratic convergence. In contrast to the Monte Carlo version of the registration algorithm, the limit of this SB-Newton registration is much closer to the identity matrix:

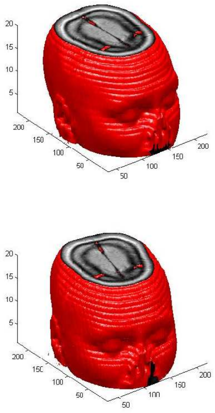


Figure 5.3: Comparison of the SB-Newton with the QMC-Newton algorithm. We register the distance from the requested transformation to the result of the particular algorithm. We perform the QMC-Newton algorithm with the first $N = 2000, 6000$ and 10000 elements of the Halton-sequence.

Number of Coefficients	SB-Newton Algorithm	QMC-Newton Algorithm		
		$N = 2000$	$N = 6000$	$N = 10000$
$11 \times 11 \times 18$	0.0045 0.45	0.075 9.5	0.023 2.3	0.0056 1.5
$16 \times 16 \times 18$	0.0016 0.15	0.013 4.5	0.017 1.7	0.014 2.1
$24 \times 24 \times 18$	$8.8 \cdot 10^{-5}$ 0.47	0.042 5.6	0.0042 0.49	0.0036 0.7
$50 \times 50 \times 18$	$1.1 \cdot 10^{-4}$ 0.4	0.39 52	0.0026 0.3	0.0027 0.19

In the case of 225 spline-coefficients the algorithm ends with

$$M_{20} = \begin{pmatrix} 1.0008 & -0.0004 & -0.043 \\ -0.0002 & 0.9992 & 0.12 \\ 0 & 0 & 1 \end{pmatrix}.$$

In case of 2304 coefficients we achieve a distance to the identity matrix of $3.7 \cdot 10^{-4}$.

In our next example, we examine the convergence of the $SE(3)$ algorithms. We consider a $250 \times 250 \times 20$ CT data set of a head (see Fig. 5.3). In contrast to the first example, we first transform the image and then make use of the spline-approximation to achieve the template (which is much closer to a natural registration problem). The transformation used is

$$\begin{pmatrix} 0.9553 & -0.2955 & 0 & -32.1 \\ -0.2955 & 0.9553 & 0 & 43.5 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \tag{5.36}$$

which is consistent with a rotation of 17.2 degrees around the central principal axis of inertia. In Fig. 5.3 we note the difference between the detected and the exact transformation: The first number in each box shows the distance of the detected to the real rotation matrix in the Frobenius norm, the second number the Euclidean distance of the detected to the real translation in pixel length. Note that a translation error smaller than 1, which is the size of one voxel, can be seen as perfect. We vary the number of coefficients used for the spline-approximation and the number N of elements of the Halton-sequence used for the Quasi-Monte Carlo method.

We recapitulate: For the algorithms defined in Section 5.1.1 we need two approximations: a spline-approximation to smooth the image and a couple of sample-points to approximate the integrals (2.27). Also two approximations are made in the algorithms of Section 5.2.1: the spline-based image smoothing and the approximation of the objective function (5.17). In case of the SB-Newton algorithm we achieve an overwhelming accuracy even for very strongly smoothed images. For a given smoothing level, the operation on the spline function space

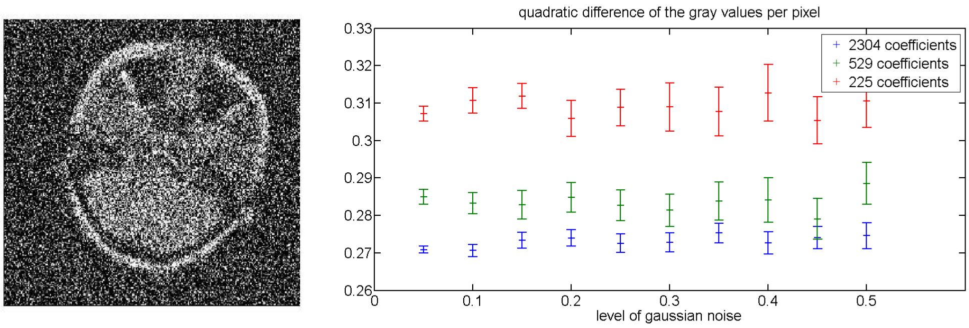


Figure 5.4: Left: Template image with 50% Gaussian noise. Right: For the Gaussian noise from 5% to 50% we plot the mean and the 90% confidence interval of the error (5.37), detected by the SB-algorithm.

seems to be superior to the Monte Carlo approximation. On the other hand, the speed of the Monte Carlo based algorithms is hardly influenced by the level of image smoothing. It depends nearly completely on the chosen number of sample-points for the integral approximation. Therefore, a comparison of the algorithms focused on the rows of Fig. 5.3 is limited. When comparing the columns of Fig.5.3 it could lead to the impression that we achieve better results by increasing the number of spline-coefficients. This is not generally true: For example, by reducing the smoothing local extrema appear, which leads to wrong results. This happens, for example, in the last row of Fig. 5.3 for 2000 sample-points.

In the next experiment, we study the influence of noise on the result of the SB-Newton algorithms. As before, we take the upper left image in Fig. 5.2 as the reference R with 250×250 pixels and gray values in the range $[0, 932]$. In order to construct the template image T, we perform a rotation of the reference around its center with 11.5° and add Gaussian noise varying from 5% to 50%. That is, the variance of the noise lies between 0.05 and 0.5 after rescaling the range of the image to the interval $[0, 1]$. The left of Fig. 5.4 shows the template disturbed by the highest noise level. For each noise level we project the reference and the template image to a spline function space and use the SB-algorithm to detect the transformation. For the detected and the exact transformation M_{detect}, M_{exact}, we measure the discrepancy with

$$\frac{1}{250^2}\sum_i \left(f(PM_{\text{detect}}\bar{x}_i) - f(PM_{\text{exact}}\bar{x}_i)\right)^2, \tag{5.37}$$

where the sum is taken over all pixels of the image. This can be seen as the averaged quadratic difference of the gray values. For each noise level we consider three different cases of spline function spaces, with 15×15, 23×23 and 48×48 coefficients. We repeat each experiment 50 times and evaluate (5.37) for each detected transformation M_{detect}. On the right of Fig. 5.4 we plot the particular mean value and the 90% confidence interval. As in the previous experiments, we observe a systematical error caused by the spline-approximation. In comparison to this, the additional error caused by the Gaussian noise is negligibly small, even for large variances. This is not surprising since the spline approximation of the image is performed with respect to the SSD-norm, known to be the unbiased estimator in the case

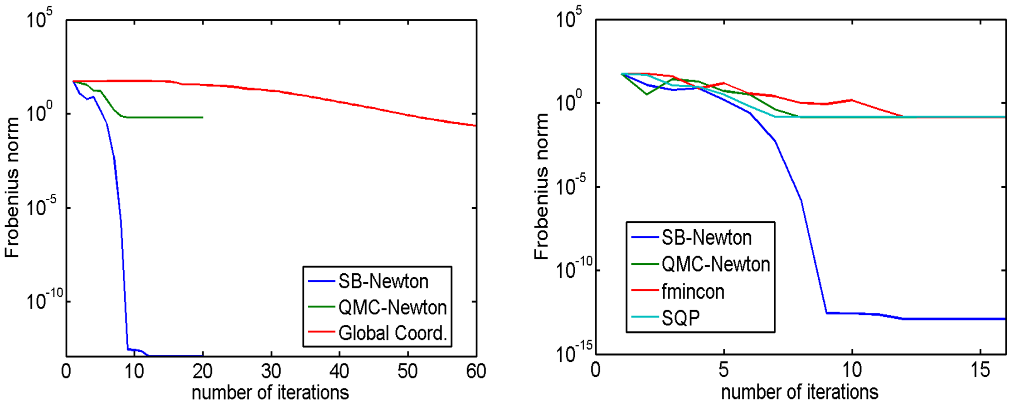


Figure 5.5: The distance from the current iteration matrix to the identity is drawn. We run several algorithms for the the same registration task starting at (5.39). Reference and template are identical. Left: comparison of unconstrained optimization (Trust-region method red) and manifold-types (SB- and QMC-Newton). Right: comparison of constrained optimization (standard SQP-method of MATLAB red and a self-made SQP method turquoise) and the manifold-types (SB- and QMC-Newton).

of Gaussian noise with zero mean. Another result of this experiment is that the systematic error of the spline-approximation is very small (the range of the images is $[0, 932]$), as we have already pointed out in Fig. 5.3. This underlines a well-known fact in image processing, namely that we loose less information of an image by using a spline approximation than using a standard interpolation method (cf. [71]). Moreover, if we want to detect the exact transformation perfectly, we have to incorporate the algorithms in a pyramidal approach in which we gradually increase the dimension of the spline function space. This procedure is quite common and already implemented in various registration algorithms (see e.g. [2] and the references therein). We will discuss this pyramidal approach more detailed in Section 5.3.3.

5.3.2 Comparison with Optimization Techniques on Vector Spaces

In this subsection we cast a critical eye over our approach which tackles the registration task as an optimization problem on manifolds. Of course, there are other approaches for solving problem (3.27) known from classical optimization theory and the question about the benefits of our algorithms arises. In the following, we will introduce two alternative optimization strategies and compare them to our QMC- and SB-Newton algorithms. Throughout this subsection, we will focus on rigid transformations in the plane (i.e. on $SE(2)$).

For instance, one can parameterize the transformation group with only one global chart and treat the problem (3.27) as an unconstrained optimization problem. For our first alternative strategy, we can define

$$\psi_{\alpha,t_1,t_2} : \mathbb{R}^3 \to SE(2), \quad (\alpha, t_1, t_2) \mapsto \psi_{\alpha,t_1,t_2} = \begin{pmatrix} \cos\alpha & -\sin\alpha & t_1 \\ \sin\alpha & \cos\alpha & t_2 \\ 0 & 0 & 1 \end{pmatrix}$$

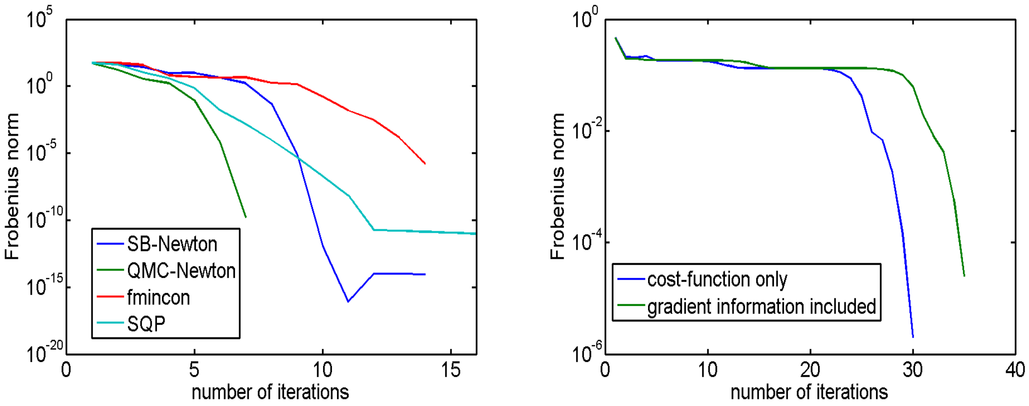


Figure 5.6: We consider the same registration task as in Fig. 5.5. We draw the distance from the current iteration matrix to the particular limit. Left: the same algorithms as in Fig. 5.5. Right: `snopt` solver with no and full gradient information.

and solve the optimization problem

$$\max_{\alpha,t_1,t_2\in\mathbb{R}} \frac{1}{N}\sum_{i=1}^{N} R(P\psi_{\alpha,t_1,t_2}x_i)T(x_i). \tag{5.38}$$

Once again, we choose the sample-points $\{x_i\}_{i=1}^N$ as the elements of the Halton-sequence. This is a quite analogous approach to the cost-function in (5.3) for the QMC-Newton method. In literature, there is already a huge variety of implemented Newton algorithms which can handle problems of the form (5.38), see e.g. [103] for an overview. Here, we chose the `fminunc`-command of the optimization toolbox of MATLAB, which is a Trust-Region method[1] based on [104].

In our first comparison, we took a 2D CT slide of a head as the reference, and the template image (left image at the top on the Fig. 5.2). Afterwards, we construct the functions R and T by a spline-approximation of the reference and the template using 2500 coefficients. In the case of the QMC- and the SB-Newton algorithms, we start with the initial matrix

$$M_0 := \begin{pmatrix} 0.9553 & -0.2955 & -32.1 \\ -0.2955 & 0.9553 & 43.5 \\ 0 & 0 & 1 \end{pmatrix} \tag{5.39}$$

which is the representation matrix of the 17.2° rotation around the center of a 256×256 image. In the case of the `fmincon`-solver we start with $(\alpha_0, t_{1,0}, t_{2,0}) = (17.2^\circ, -32.1, 43.5)$. Afterwards, we plot $\|M_k - I\|_{\mathrm{Fr}}$ for the QMC- and SB-Newton method and $\|\psi_{\alpha_k,t_{1,k},t_{2,k}} - I\|_{\mathrm{Fr}}$ for the unconstrained optimization method, where k denotes the particular iteration. The result is shown in Fig. 5.5.

One can see that the global chart approach needs a lot of iterations to provide the same accuracy as the QMC-Newton method. This is not surprising: The global chart ψ describes

[1] See `http://www.mathworks.com/help/toolbox/optim/ug/fminunc.html` for a documentation.

a rotation around the origin with an additional translation. However, the origin is not the center of the image, but the corner at the top on the left. Therefore, in terms of ψ, the algorithms have to produce a sequence ranging from $(17.2°, -32.1, 43.5)$ to $(0°, 0, 0)$. If we had defined ψ in such a way that the center of rotation is the center of the image, we would have started with the vector $(17.2°, 0, 0)$ instead, which is much closer to the starting point $(0°, 0, 0)$. Hence, the speed of the algorithm depends on a good choice of the global chart, which depends on the result of the algorithm. In comparison, the manifold approach does not have such drawbacks. The speed of this method only depends on the geodesic distance from the starting point to the limit (which is, of course, independent from a particular representation).

Another approach to the registration problem (3.27) is to consider it as a constraints optimization problem. For the special case of a rigid 2D registration, we can rewrite (3.27) to

$$\begin{aligned}\min_{a_1,\ldots,a_6\in\mathbb{R}} \quad & \frac{1}{N}\sum_{i=1}^{N} R\left(\begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix} x_i + \begin{pmatrix} a_5 \\ a_6 \end{pmatrix}\right) T(x_i) \\ \text{st.} \quad & a_1^2 + a_2^2 = 1 \\ & a_3^2 + a_4^2 = 1 \\ & a_1 a_3 + a_2 a_4 = 0. \end{aligned} \tag{5.40}$$

The constrains guarantee that the 2×2 matrix in the cost function is orthogonal. Once again, there is a huge variety of algorithms for solving (5.40). Here, we choose the SQP method since it is known to converge also quadratically and it is quite often implemented in optimization packages. In this method, the objective function is approximated quadratically in every step, while the constraints are linearized. The resulting Newton-step is then performed in a vector space of larger dimension. For an introduction to the SQP-method, we refer to [103] chapter 12.4. In the following example, we use the `fmincon`-command[2] of the optimization toolbox of MATLAB and a self-implemented version of the SQP-method, based on the book of Fletcher[103] pp. 304 with a step-size condition described in [105] pp. 249.

On the right of Fig. 5.5 we consider the same registration task as in the comparison on the left. We plot again the distance from the current iteration M_k to the sought deformation I_3. Here, the SB-Newton is the only algorithm which really detects I_3; the other algorithms converge to a different limit. As already mentioned in the previous subsection, this is caused by the fact that the SB-Newton optimizes a slightly different cost-function than the other methods. Hence, in order to compare the convergence rate of the algorithms, we also calculate $\|M_k - M_\infty\|$, which is the distance from the current iteration to the limit of the particular algorithm. The corresponding graphs are shown in Fig. 5.6.

One result of this experiment is the superior convergence rate of the SB-Newton and the QMC-Newton algorithms when compared to the SQP-type methods (self-made and `fmincon`-solver). This stresses the point of view that intrinsic algorithms dominate extrinsic algorithms if the constraints form a differentiable manifold. By embedding the set of admissible points in a vector space, which is done in an extrinsic algorithm like the SQP, the dimension of the optimization problem may explode. In order to offer an example, the SQP method for

[2] See `http://www.mathworks.com/help/toolbox/optim/ug/fmincon.html.` for a documentation,

registration problems in $SE(3)$ searches in a space of 18 dimensions (9 according to $A \in gl(3)$ instead of $A \in SO(3)$, 3 according to $t \in \mathbb{R}^3$ and 6 parameters are needed for the Lagrange multipliers), whereas the SB-Newton and QMC-Newton methods optimize over a space of 6 dimensions (3 according to $A \in SO(3)$ and 3 according to $t \in \mathbb{R}^3$). Hence, an extrinsic algorithm may require unnecessarily many steps, a higher complexity, and finally, a projection step is needed to make sure that the result is an admissible point, since it is only guaranteed that the limit of an extrinsic algorithm is admissible.

One may argue that the MATLAB optimization toolbox is not the "state of the art" for constrained optimization tasks. Therefore, we also involve the TomLab optimization package[3]. We run the same numeric experiment as before but this time with the provided `snopt`-solver. Additionally, we perform this experiment in two variations: one with explicit given gradient of the cost function and one with a gradient-estimation, done by the solver during the optimization in some minor iterations. The convergence rate is shown in the right graphic of Fig. 5.6 (the minor iterations are not counted). Once again, the speed of convergence is quite low compared to the manifold-type algorithms shown on the left. Even the self-made SQP-algorithm surpasses the `snopt`-solver. One reason for this might be that, in contrast to the self-made SQP-method, neither `snopt` nor `fmincon` uses the Hessian information, even if they are available. One also should mention that the self-made SQP is, like the QMC- and SB-Newton algorithms, made exclusively for the optimization on $SE(n)$, whereas the TomLab solvers are rather general and can be used for different kinds of optimization problems. It is not a surprise that you get faster algorithms when you specify the problem. Therefore, comparing the manifold-methods with the self-made SQP is perhaps the fairest numeric experiment.

Counting the number of iterations is only one aspect of comparing different algorithms. Another point is the entire computation time. However, one has to keep in mind that we test self-made (and partly naively implemented) algorithms against commercial programs. In case of the `snopt`- and `fmincon`-solver, we stopped how long it took for the default stopping criteria to be satisfied (which is the point where the corresponding graphs in Fig.5.6 break off). In case of the other algorithms, we took the time until the values of the cost function in the iteration-points differ only between $\pm 10^{-10}$. The particular computation time is summarized in the following table:

Algorithm	SB-Newton	QMC-Newton	`snopt`	`snopt` (gradient approx.)	SQP	`fmincon`
Iteration	11	8	35	30	12	14
Time	4.71s	11.4s	35.26s	96.18s	52.04s	17.91s

These computations are done on a HP Proliant DL385, 64GB RAM, AMD Opteron 2435.

It turns out that the most time-consuming step in the QMC- or SB-Newton algorithm is Step 2 of Table 5.1, i.e. the one for calculating the function values or derivatives of the images (not the cost-function itself). The other steps operate on a low dimensional matrix space, and are therefore negligible for counting the floating-point operations. For instance, if the images are two-dimensional, the linear equation in Step 3 has dimension three (if the image is three-dimensional, the equation is six-dimensional), which can be handled very

[3] See `http://tomopt.com/tomlab/download/manuals.php` for a documentation.

quickly. However, the calculation of the coefficients in Step 2 depends on the length of the chosen Halton-sequence N for the QMC-Newton (cf. (5.4)) and the number of sample-points for the SB-Newton (cf. (5.20)) respectively. If it is a high-resoluted image, this operation dominates. Roughly speaking, if the dimension of the deformation group is small, then the computation time depends almost completely on the image resolution. Therefore, it seems inappropriate for us to count floating point operations, but image evaluations (ie) of R and T (and their derivatives). Hence, a second order iteration step, like the one of the QMC-Newton, needs $N(n + \frac{1}{2}n(n+1))$ ie, with image dimension n and N sample-points, Nn for the first derivatives and $\frac{1}{2}n(n+1)N$ for the second ones. In contrast to this, an algorithm which generates the step with the first derivative, like steepest ascend or conjugate gradient methods, needs only Nn ie. Here, we neglected minor iterations like line-searches in both cases. Hence, in the case of two dimensional images, switching from a gradient-type method to a Newton-type method rises the computational costs from $2N$ ie to $5N$ ie. In case of three-dimensional images we get an increase from $3N$ ie to $9N$ ie. However, we don't see this increase on the table. This might be caused by the fact that we didn't incorporate minor iterations in this study.

A comparison of the numerical cost of QMC-Newton and SB-Newton is a more difficult task. Since the sums in (5.4) and (5.20) have approximately the same length, the only difference lies in the accessibility of the matching coefficients for the SB-Newton (since F has compact support, the summands in (5.20) are zero for the most part) compared to the image evaluations for the QMC-Newton. Both computations depend extremely on the particular implementation and we do not dare to decide which one is the quickest method.

Nevertheless, for us the most interesting result while working with the TomLab package is that the `snopt`-solver works slightly better (in iteration steps - not in time) if we don't provide gradient-information. One might interpret this as an additional drawback of the constraint optimization approach.

We also tested the `knitro`-solver of the TomLab package. But since we were not able to extract the iteration points, (it only yield the cost function value at each iteration,) we leave out a graphical comparison. Using the same stopping criteria as for the `snopt`-solver, `knitro` needs 34 iterations for the optimization, which is quite a similar behavior as the `snopt`-solver shows.

5.3.3 Comparison with Previous Registration Algorithms

We already mentioned in Chapter 3 that the general registration task (3.27) is usually embedded in a multi-resolution process (3.28). Until now, we have focused on the optimization procedure for one particular smoothing level and neglected that the full registration process embeds the QMC-Newton and the SB-Newton algorithms in iterations where the high frequency parts of both images are added step by step.

We proceed with a comparison with two established intensity-based registration algorithms. We choose the Levenberg-Marquardt-like (LM) algorithm as described by Thévenaz et al. [24], since the same framework requirements are chosen to formulate the problem. Above all, they also make use of the spline-approximation of the images several times. In their approach, they use the substitution-rule in (5.6) to get a good approximation of the derivative of the discretization (5.3) and thus achieve low computational costs. We also compare our algo-

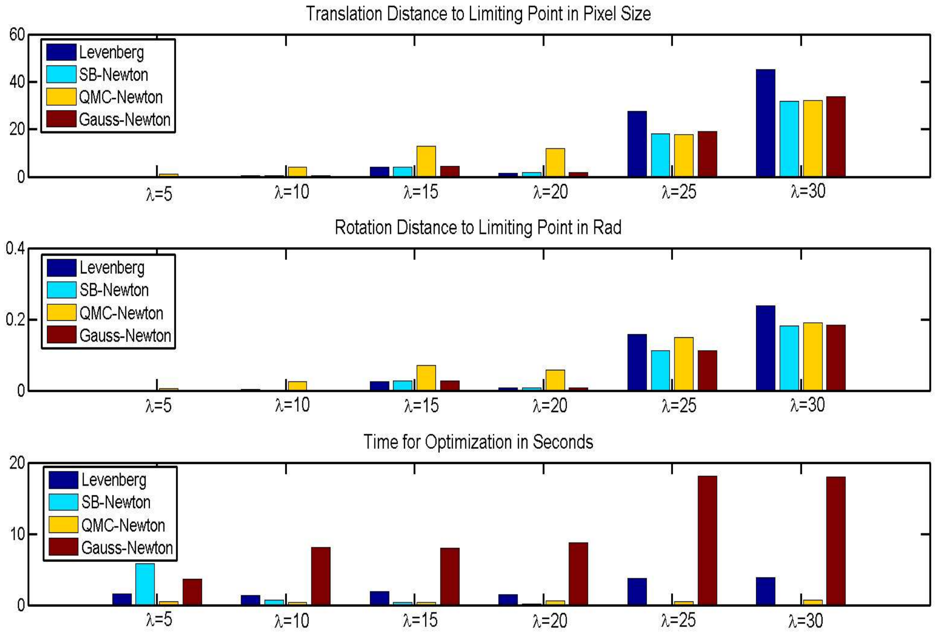


Figure 5.7: Comparison of different registration techniques in a multi-resolution process: the reference is a 256 × 256 CT slide of a head. The template is generated by rotating the reference 17.2° around the center. Both images are projected on $\mathcal{S}_\lambda^k$, $\lambda = 30, 25, 20, 15, 10, 5, 1$, beginning at $\lambda = 30$. At each smoothing level we visualized the time for the optimization in $\mathcal{S}_\lambda^k$ and the distance of result from the the optimization in $\mathcal{S}_\lambda^k$ to the result of the optimization in in $\mathcal{S}_1^k$.

rithm with a method proposed in [10, 89]. There, the optimization process at the particular smoothing level is done by a Gauss-Newton (GN) method, combined with an Armijo line-search.

Just like in the previous subsection, we take the previous 2D-CT slide as reference R and a translated and rotated version of it for the template T (17.2° around the center). In Fig. 5.7 we demonstrate the convergence-behavior of the SB-Newton, the QMC-Newton, the LM and the GN registration. We run the QMC algorithms with $N = 1000$ sample-points, whereas the LM and GN algorithms incorporating all pixels $N = 256^2$, like it was published in the respective papers. We started at $\lambda = 30$ and projected both images on $\mathcal{S}_\lambda^k$. (The choice of $k \in \mathbb{N}$ depends on the particular algorithm. For LM and GN we choose $k = 4$ and for SB and QMC we refer to Section 5.1.1 and 5.2.1 respectively.) Then, we register the smoothed images with the respective algorithm. Afterwards we choose iteratively a finer smoothing level $\lambda = 25, 20, 15, 10, 5, 1$ and take the result of the previous optimization as the initial guess for the current optimization. In the third graphic of Fig. 5.7, we plot the computation time for each optimization process (we left out the time needed for the image smoothing). For all four algorithms we used the same stopping criteria: the translation part of the stepsize had to be smaller than 10^{-4} pixels and the rotation part of the stepsize had to be smaller than 10^{-6}. The distance from the optimization result in $\mathcal{S}_\lambda^k$ to the final optimization result in $\mathcal{S}_1^k$ is visualized in the first and second diagram of Fig. 5.7 - separated into rotation and translation parts.

Requested Transformation	Detected Transformations			
	SB-Newton	QMC-Newton	LM	GN
$0°, \binom{0}{-30}$	$0.006°, \binom{0}{-29.9}$	$0.063°, \binom{-0.1}{-29.7}$	$0.012°, \binom{0.0}{-29.8}$	$0.006°, \binom{0.0}{-29.9}$
$0°, \binom{0}{-30}$	$0.094°, \binom{0.0}{-30.4}$	$0.126°, \binom{-0.2}{-30.5}$	$2.895°, \binom{-7.3}{-20.8}$	$-0.155°, \binom{-0.8}{-29.5}$
$0°, \binom{0}{-40}$	$0.052°, \binom{-0.1}{-39.8}$	$0.086°, \binom{-0.13}{-39.7}$	$21.77°, \binom{-41.6}{51.8}$	$21.3°, \binom{-50.7}{28.1}$
$17.189°, \binom{43.5}{-32.1}$	$17.188°, \binom{41.8}{-30.9}$	$16.93°, \binom{42.9}{-31.8}$	$17.194°, \binom{43.7}{-31.9}$	$17.189°, \binom{43.5}{-32.1}$

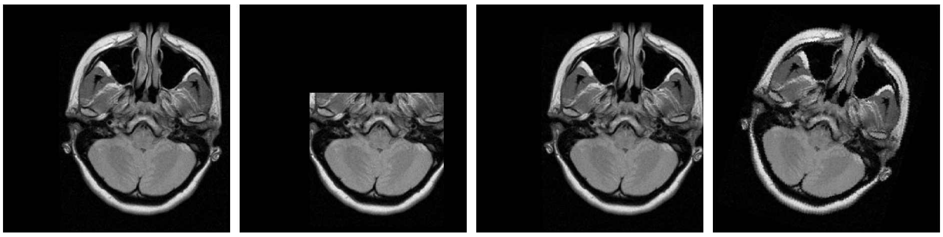


Figure 5.8: Comparison of the SB-Newton, QMC-Newton, LM and GN registration algorithms. The first column presents the requested transformation and the following columns the calculated transformations with the particular algorithms. The first number gives the rotation around the center of the image, the following vector gives the translation in pixels. Each registration process consists of a pyramidal approach with $\lambda = 20, 10, 5$. Below the table, we present the templates of the particular registration problem. The reference is equal to the one of the first experiment Fig. 5.2.

A few remarks for the evaluation of this experiment are in order: First, for all algorithms the results at $\lambda = 5$ or $\lambda = 10$ are pretty good. For instance, the translation parts of the results do not vary more than one pixel-size. It is doubtful whether an optimization at $\lambda = 1$ is needed at all in applications. It seems that the QMC-Newton shows the worst behavior when looking at the first and second graphic. However, these errors depend almost completely on the number of chosen sample-points N, as well as the computation time, which is relatively low compared to LM and GN algorithms. Second, the computation-time of the SB-Newton is reciprocal to λ. At each smoothing level, the other algorithms have to incorporate the same amount of pixels and sample-points respectively at each step, but the time for one SB-Newton step depends on the dimension of $\mathcal{S}_\lambda^k$, which increases for $\lambda \to 1$. This effect is most dramatic at $\lambda = 1$ of the SB-Newton where 123 seconds are needed to finish the optimization. In contrast, the computation-time for a coarse smoothing level is extraordinarily good.

Since the algorithms use different kinds of approximations, the particular limits differ. We list them in Fig. 5.8 for a few registration problems. In the first one, we move the template 30 pixels to the right. For the second registration task, we take the same transformation, but cover some parts of the template. In the third task, the requested transformation is a movement along 40 pixels to the right and a 17.2° degree rotation around the image center for the fourth task, respectively. In accordance with the previous experiment, we take the smoothing levels $\lambda = 20, 10, 5$ for each registration process - in particular, we leave out the finest level $\lambda = 1$. The QMC-Newton algorithm is performed with $N = 5000$ sample-points.

Since we use the same cost function for the LM-Newton and the GN-algorithm it is not a surprise that they produce similar results. The only exception is the second task, where parts

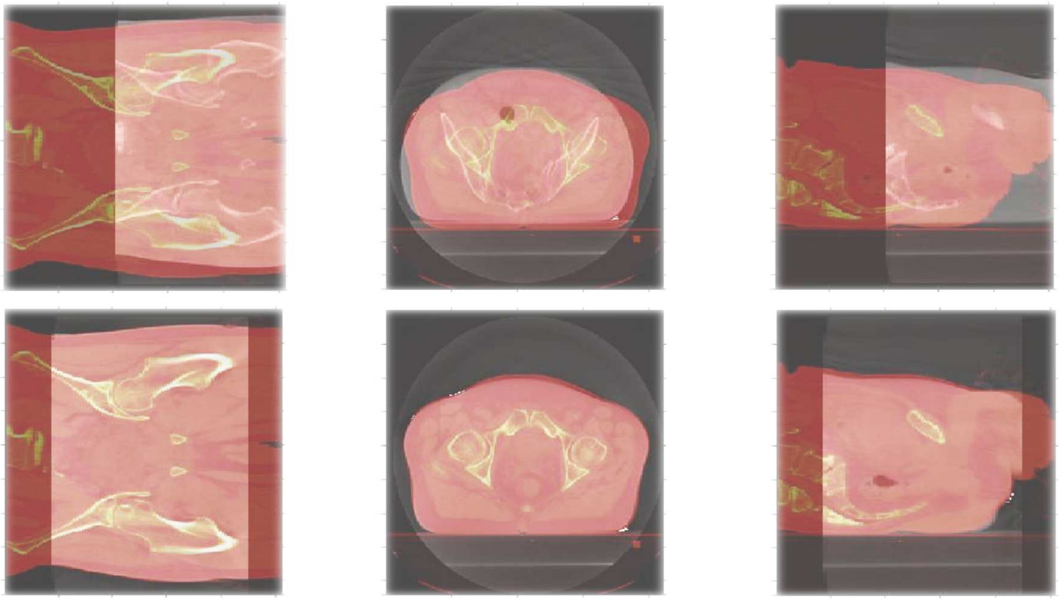


Figure 5.9: The first row shows the initial position of the template with respect to the reference. The second row shows the estimated relative position after 12 SB-Newton-steps. We used a CBCT-image for the reference and a FBCT-image for the template.

of the template are covered. This might lead to many appearing local minima. It is notable that in the third row the detected translations in the case of the LM and GM algorithm are wrong. The main reason for this is that LM minimizes the "sum of squared difference"-measure while the QMC or SB-Newton algorithm maximizes the correlation between two images. This is an effect, which is visible for all registration algorithms and also for the QMC and SB if we move the template 50 pixels to the right. Although, the multi-scale approach pushes the detected transformation more likely to the requested transformation, the all in all algorithm does not necessarily detect global optimum. However, LM and GM could also detect the 40 pixels translation if we used a finer scaling scheme like $\lambda = 30, 25, 20, 15, 10, 5$. The drawback of the correlation measure becomes evident in the last line: The QMC and the SB algorithm make a small mistake in detecting the translation part of the requested transformation.

It turns out that comparing the numerical costs of the algorithms is a difficult task: In [24] the authors overcome the necessity of evaluating the gradient of one image in each iteration, but the numerical costs are of the order N, since they sum over all N pixels on the image. In our QMC-approach we sum only over a uniformly distributed grid of $\tilde{N}$, which creates a certain (but hopefully negligible) error ϵ. Since we have $\epsilon = O((\log \tilde{N})^2/\tilde{N})$ in the case of 2D images, we can usually choose $\tilde{N} \ll N$, which reduces the computational costs. We want to point out that reducing the sum over all pixels to a sum over a grid with low discrepancy is also applicable to the LM-algorithms and leads to quite small approximation errors as in the case of the QMC-Newton method. Regarding the SB-Newton algorithm the computational costs are of the same order as the number of the spline-coefficients taken from the approximation. On the one hand, this number is often extremely small in comparison to the image size, since

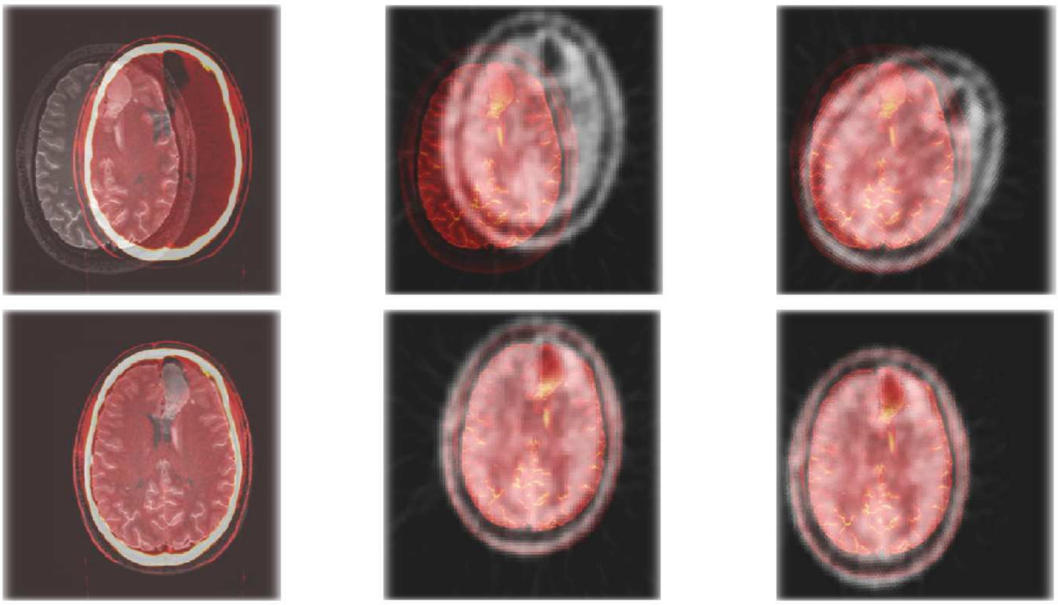


Figure 5.10: The first row shows the initial situation of three registration problems. In the first one we compared a CT- with an NMR-image and in the second and third, we compare NMR- with PET-images. The second row presents the result of the mutual information-based algorithm after ten SB-Newton-steps.

the results of the SB-Newton algorithm are more than sufficient even on a very coarse level of spline-approximation. On the other hand, for each spline coefficient, a series of B-splines has to be evaluated at different values (cf. (5.19)), which accounts for high numerical costs per spline-coefficient. Therefore, the SB-Newton algorithm becomes very slow for a fine level of spline-approximation.

Now, we demonstrate the algorithms with the help of some medically relevant pictures. In Fig 5.9, reference and template are FBCT and CBCT shots of the prostate area. These two datasets consist of $420 \times 420 \times 72$ and $512 \times 512 \times 101$ voxels respectively. We will not use a priori information about their relative positions to each other. Instead, we initially put the reference on the lower part of the template (see also the first row of Fig. 5.9).

The beginning of the registration consists of comprising the data set to $26 \times 26 \times 25$ spline coefficients each. The second row of Fig. 5.9 shows the result of the registration after 12 SB-Newton steps. Any further steps only provide translations below the size of a voxel and rotations below 0.01 degree.

Due to the great similarity of the two recording methods (FBCT and CBCT) in the last example, the monomodal SB-Newton algorithm on $SE(3)$ was used for the registration. With the help of the procedure described in Section 5.2.2 one is also able to use the algorithm for data which have been recorded with very different modalities. We will show this in the following example where we compare a MR picture with a CT (each with 512×512 pixels) by using a PET- picture (128×128 pixels) and we use the mutual information-based registration algorithm on SE(3). The first row of Fig. 5.10 shows the origin of the correlating registration problems, and the second one reveals the result after 10 steps. We use a compression of 64×64 spline coefficients for each case mentioned. In the last column of Fig. 5.10, we moreover add an additional rotation of 30 degrees. Once again, we reach a good match of the pictures within 10 steps. We have already pointed out that this method only provides a local convergence. If the pictures differed in their initial position the algorithm would not provide an acceptable result even if one tries to use more steps. This happens if we rotate the PET- picture, as seen on the third column in Fig. 5.10, more than 30 degrees.

Chapter 6

Image Registration using Stochastic Optimization

In the previous chapter we applied a Newton method to the rigid image registration problem and obtained algorithms with excellent convergence behavior and accuracy. However, these methods reveal their potential only in a (sometimes small) domain around a critical point. In some applications, it is therefore essential to insert a pre-optimization step. Hence, in this chapter we will use the Robbins-Monro type algorithms, and apply them for the special task of rigid image registration.

Algorithms of the Robbins-Monro type are already well accepted in image registration. For instance, we can interpret the method introduced by Viola [86, 88] as an algorithm of this form. Moreover, Klein et al. presented in [52] an extensive comparison of several non-rigid multimodal registration methods with the result that "the Robbins-Monro method is the best choice in most (medical) applications".

The basic idea of this method is to reduce the numerical costs per iteration even if this would reduce the convergence rate and the accuracy. Since the most problematic part of the previous algorithms is the enormous number of image evaluations, it would be good if one can reduce this number. Indeed, the image evaluations appear only in terms which average over all pixels like in (5.4) or (5.20). In the following stochastic approach, we replace such terms and average only over a few, randomly chosen pixels instead.

6.1 Variants of the Robbins-Monro Algorithm

In Section 2.3 we extended the Robbins-Monro algorithm to Lie groups. Now we will apply this algorithm to the monomodal and the multimodal registration task. One key point will be that the original iteration (2.39) was made for root-searching in vector-field. Thus, we are not restricted to gradient-fields, like it was done in the original Robbins-Monro algorithm. For instance, we will present a new stochastic form of the Gauss-Newton method in Section 6.1.1 and in Section 6.1.2.

The other key point is that the optimization problem is not inherently connected with perturbations: Once both images are given, there is no reason, why the evaluations of the images (or their derivatives) should be disturbed. We will see that introducing an artificial perturbation may yield an acceleration of the algorithm.

6.1.1 Affine Monomodal Registration

In this subsection, we will apply the previous stochastic optimization method (2.61) to the registration task on $SE(n)$ and $SA(n)$. Thus, we set $\mathcal{M} = SE(n)$ or $\mathcal{M} = SA(n)$. For the local parameterization μ of the Special Euclidean Group we choose either (2.10) or (2.11) and for the Special Affine Group we choose either (2.6) or (2.8), respectively. Hence, we can set $\mathcal{U} = \mathcal{M}$.
Since we want to focus now on the monomodal registration task, we use the SSD-measure to generate the cost function $f : \mathcal{M} \to \mathbb{R}$,

$$f(A,t) = \frac{1}{N}\sum_{i=1}^{N} \left[R(x_i) - T(Ax_i + t)\right]^2 . \tag{6.1}$$

In contrast to the Quasi-Monte Carlo approach in Chapter 5, here the sum is taken over all N pixels $\mathcal{X} := \{x_i\}_{i=1}^{N}$ of a given region of interest and the images R and T are supposed to be continuous differentiable, i.e. of the form (3.6) or (3.8). At a first glance, there is no need to introduce any noise U_k into the registration problem. We could simply set $U_k = 0$ for all $k \in \mathbb{N}$ in the iteration (2.61). The result would be a gradient-based descent algorithm with harmonic sequence for the step size. However, we observed in Section 5.3 that the most time-consuming part in an registration algorithm is the number of image evaluations. In particular, for $U_k = 0$ we would need $2N \dim \mathcal{M}$ image evaluations *ie* per step. Hence, an alternative choice of the noise would be favorable which leads to an decreasing of the number of image evaluations, for instance

$$U_k := \nabla\left(f \circ \mu_{(A_k,t_k)}\right)(0) - \nabla\left(\frac{1}{m}\sum_{i\in J_k}\left[R(x_i) - T(\mu_{(A_k,t_k)}(.)\circ x_i)\right]^2\right)(0) \quad \text{for all } k \in \mathbb{N}, \tag{6.2}$$

where the sum is taken over $|J_k| = m$ randomly chosen elements $J_k \subset \{1,\ldots,N\}$. In particular, the subsets $J_0, J_1, \ldots$ are chosen pairwise independently and also the elements in J_k are uniformly and pairwise independently distributed. Using (6.2) and (6.1) in the iteration (2.61) decreases the number of image evaluations to $2m \dim \mathcal{M}$ *ie*, where $m \ll N$. We will point out this advantage in Section 6.2. The final stochastic gradient algorithm for image registration on $SE(n)$ is summarized in Table 6.1.
In order to apply Corollary 2.16 to this registration method, we have to verify the necessary properties of the noise U_k: First, since the elements of J_k are chosen uniformly and pairwise independently, we have

$$E\left(\nabla\left(\frac{1}{m}\sum_{i\in J_k}\left[R(x_i) - T(\mu_{(A,t)}(\ .\ )\circ x_i)\right]^2\right)(0)\right) = \nabla\left(f\circ\mu_{(A,t)}\right)(0),$$

for all $(A,t) \in SE(n)$. This implies

$$E(U_{k+1}\big|U_1,\ldots,U_k) = 0,$$

Table 6.1: Stoch. Gradient Alg. for Monomodal, Affine Image Registration

Given: $R, T \in C^1(\mathbb{R}^n, \mathbb{R})$, a set of voxels $\mathcal{X} \subset \mathbb{R}^n$, $\mathcal{M} = SE(n)$ or $\mathcal{M} = SA(n)$ and a local parameterization μ of $\mathcal{M}$, e.g. (2.11) or (2.8).
Parameter: $m \in \mathbb{N}$, $a, b \in \mathbb{R}^+$, $\omega \in (0.5, 1]$.

Step 1
Make an initial point $(A_0, t_0) \in \mathcal{M}$. Set $k = 0$.

Step 2
Chose a uniformly and independent distributed set of sample-points $\{y_i\}_{i=1}^m \subset \mathcal{X}$

Step 3
Set $\Psi(A,t) := \frac{1}{m} \sum_{i=1}^m [R(y_i) - T(Ay_i + t)]^2$ and calculate $(\tilde{\Omega}, \tilde{v}) := \nabla(\Psi \circ \mu_{(A_k,t_k)})(0)$.

Step 4
Set $\gamma_k := \frac{a}{(b+k)^\omega}$ and calculate $(A_{k+1}, t_{k+1}) := \mu_{(A_k,t_k)}\left(-\gamma_k \tilde{\Omega}, -\gamma_k \tilde{v}\right)$.

Step 5
Set $k = k + 1$ and goto Step 2.

which verifies (iii) of the *Martingale difference Noise* property. Second, under the assumption that $T \in C^1(\mathbb{R}^n, \mathbb{R})$ has compact support, $\nabla T(x)$ is bounded. The sequence $(U_k)_{k \in N}$ is consequently bounded. Therefore, the statement of Corollary 2.16 is valid for the algorithm in Table 6.1.

A few remarks on this algorithm are in order: According to Corollary 2.16, the gain-sequence $(\gamma_k)_{k \in \mathbb{N}}$ should satisfy (2.62). Since we set $\omega \in (0.5, 1]$, this is guaranteed for $q = 2$. The parameters $a, b \in \mathbb{R}^+$ have to be carefully adjusted by the user. On the one hand, if γ_0 is too large, the iteration $((A_k, t_k))_{k \in \mathbb{N}}$ might diverge or jump to undesirable local minima. On the other hand, if γ_0 is too small, the algorithm will converge very slowly or it will (in numerical experiments) converge to a non-critical point[1]. For a deeper discussion about the parameters a, b, ω we refer to [56]. In Step 3, we introduced the function $\Psi(A,t)$. Since this function satisfies

$$\nabla(\Psi \circ \mu_{(A_k,t_k)})(0) = \nabla(f \circ \mu_{(A_k,t_k)})(0) + U_k$$

the Step 4 fulfills exactly the iteration (2.61). Finally we want to point out that $E(\Psi(A,t)) = f(A,t)$ holds. We can consequently interpret Ψ as a noisy measurement of the function f.

The special form of the function Ψ might tempt one to a variation of the algorithm in Table 6.1: Since $\Psi =: \|\psi\|^2$ is a least-squares cost function, we could use a Gauss-Newton

[1] This is no contradiction to Corollary 2.16. One should keep in mind that the series $\sum_{n=1}^\infty \frac{1}{n}$ does converge in computer experiments, even if we know it better.

Table 6.2: Stochastic Gauss-Newton Algorithm for Monomodal, Affine Image Registration

Given: $R, T \in C^1(\mathbb{R}^n, \mathbb{R})$, a set of voxels $\mathcal{X} \subset \mathbb{R}^n$, $\mathcal{M} = SE(n)$ or $\mathcal{M} = SA(n)$
and a local parameterization μ of $\mathcal{M}$, e.g. (2.11) or (2.8).
Parameter: $m \in \mathbb{N}$, $a, b \in \mathbb{R}^+$, $\omega \in (0.5, 1]$. $\Xi \subset T_{id}\mathcal{M}$ compact neighborhood of 0.

Step 1
Make an initial point $(A_0, t_0) \in \mathcal{M}$. Set $k = 0$.

Step 2
Chose a uniformly and independent distributed set of sample-points $\{y_i\}_{i=1}^m \subset \mathcal{X}$

Step 3
Define $\psi : \mathcal{M} \to \mathbb{R}^m$ with $\psi_i(A, t) := \frac{1}{\sqrt{m}} \left(R(y_i) - T(Ay_i + t)\right),\ i = 1, \dots, m$ and calculate $C := D(\psi \circ \mu_{(A_k, t_k)})(0)$ and $d := \psi(A_k, t_k)$.

Step 4
Solve the linear equation $(C^\star \circ C) \circ [\tilde{\Omega}, \tilde{v}] = -C^\star \circ [d]$ in the unknowns $\tilde{v} \in \mathbb{R}^n$ and $\tilde{\Omega} \in so(n)$ for $\mathcal{M} = SE(n)$ and $\tilde{\Omega} \in sl(n)$ for $\mathcal{M} = SA(n)$, respectively.

Step 5
If $(\tilde{\Omega}, \tilde{v}) \notin \Xi$, set $k = k + 1$ and goto Step 2. Else, set $\gamma_k := \frac{a}{(b+k)^\omega}$ and calculate

$$(A_{k+1}, t_{k+1}) := \mu_{(A_k, t_k)} \left(-\gamma_k \tilde{\Omega}, -\gamma_k \tilde{v}\right).$$

Step 6
Set $k = k + 1$ and goto Step 2.

approach (see e.g. [29, 106]) instead of the gradient to calculate the update $(\tilde{\Omega}, \tilde{v})$ in Step 3. The corresponding algorithm is shown in Table 6.2.

We will see in the numerical results of the next section that this stochastic Gauss-Newton algorithm has a faster convergence rate than the stochastic gradient algorithm in Table 6.1 and it produces similar meaningful results. However, the limit of this algorithm is, in general, not equal to a critical point of the cost function f and we want to give reason for the appearance of this gap.

According to Corollary 2.14, we are not restricted to a gradient field. Hence, we will use the new vector field $F(A, t)$ instead, which is defined by

$$F(A, t) := E(\tilde{\Omega}(A, t), \tilde{v}(A, t))$$

with

$$\left(\tilde{\Omega}(A, t), \tilde{v}(A, t)\right) := D(\psi \circ \mu_{(A,t)})(0)^\dagger \circ \psi(A, t). \tag{6.3}$$

Here $C^\dagger$ denotes the Penrose Inverse of a linear map C. Then, the random vector defined in (6.3) can be seen as a Gauss-Newton step of the function $\Psi(A,t) = \sum_{i=1}^m \psi_i(A,t)^2$, where ψ_i is defined as in Step 3. Applying the iteration (2.57) to this vector field $F(A,t)$ and the perturbations $U_k = F(A_k,t_k) - (\tilde{\Omega}(A_k,t_k), \tilde{v}(A_k,t_k))$ leads exactly to the algorithm in Table 6.2. But, in contrast to the stochastic gradient algorithm, we can not guarantee that the update $\gamma_k(\tilde{\Omega}(A_k,t_k), \tilde{v}(A_k,t_k))$ is bounded. This is the reason why we introduced the set $\Xi \subset T_{id}\mathcal{M}$ and the if-loop in Step 5. Therefore, $\{U_k\}_{k\in\mathbb{N}}$ is bounded and by definition, we get

$$E(U_{k+1} \big| U_1, \ldots, U_k) = 0.$$

With this specifications $\{U_k\}_{k\in\mathbb{N}}$ satisfies the Martingale difference Noise property and, according to Corollary 2.14 and Proposition 2.15, each limit of the algorithm is a zero of the vector field $F(A,t)$.

The main problem with this approach is the question if the zeros of $F(A,t)$ coincide with the critical points of $f(A,t)$. This is, in general, not the case. However, we can still use the algorithm for image registration and we want to give an heuristic for that: Suppose, the linear map $D(\psi \circ \mu_{(A,t)})(0)$ has full rank and let $D(\psi \circ \mu_{(A,t)})(0)^\star$ denote its adjoint operator. By definition, we obtain

$$F(A,t) = E\left(\left[D(\psi \circ \mu_{(A,t)})(0)^\star \circ D(\psi \circ \mu_{(A,t)})(0) \right]^{-1} \cdot \nabla(\Psi \circ \mu_{(A,t)})(0) \right).$$

Assuming the term in brackets and the gradient are uncorrelated, it yields

$$\begin{aligned} F(A,t) &= E\left(\left[D(\psi \circ \mu_{(A,t)})(0)^\star \circ D(\psi \circ \mu_{(A,t)})(0) \right]^{-1} \right) \cdot E\left(\nabla(\Psi \circ \mu_{(A,t)})(0) \right) \\ &= E\left(\left[D(\psi \circ \mu_{(A,t)})(0)^\star \circ D(\psi \circ \mu_{(A,t)})(0) \right]^{-1} \right) \cdot \nabla(f \circ \mu_{(A,t)})(0). \end{aligned} \tag{6.4}$$

In this case, $F(A,t)$ vanishes in each critical point of $f(A,t)$ and a similar argument can be made for the conversion. Although we can verify the uncorrelation in (6.4) numerically (in our experiments on 3D data sets, the range of all correlation coefficients in (6.4) is usually $[-0.07,\ 0.07]$), a further discussion is beyond the scope of this work.

6.1.2 Affine Multimodal Registration

The multimodal registration task can, in principle, be treated in the same fashion as the monomodal one of the previous subsection. We simply have to switch from the SSD-measure to the mutual information, or the delta-distance. For instance, the stochastic gradient algorithm in Table 6.1 can be extended if we replace $\Psi(A,t)$ in Step 3 by

$$\Psi(A,t) := \tilde{H}(\rho_{R(x)}) + \tilde{H}(\rho_{T(Ax+t)}) + \tilde{H}(\rho_{R(x),T(Ax+t)}),$$

where $\tilde{H}$ is an approximation of the entropy (cf. Section 3.3.2). Since this is the only difference between a gradient-based monomodal registration and a gradient-based multimodal registration, it is not necessary for us to list the steps in a separate table.

Since Viola [86] and Wells [88] introduced the mutual information in the image registration, this approach became a standard tool in the multimodal case and is subject of ongoing

Table 6.3: Stochastic Gauss-Newton Algorithm for Multimodal, Affine Image Registration

Given: $R, T \in C^1(\mathbb{R}^n, \mathbb{R})$, a set of voxels $\mathcal{X} \subset \mathbb{R}^n$, $\mathcal{M} = SE(n)$ or $\mathcal{M} = SA(n)$ and a local parameterization μ of $\mathcal{M}$, e.g. (2.11) or (2.8).

Parameter: $m \in \mathbb{N}$, $a, b \in \mathbb{R}^+$, $\omega \in (0.5, 1]$. $\Xi \subset T_{id}\mathcal{M}$ compact neighborhood of 0, window-function $\delta : \mathbb{R} \to \mathbb{R}$, cf. Section 3.3.3

$\vdots$

Step 2
Choose two uniformly and independent distributed sets of sample-points $\{y_i\}_{i=1}^m, \{x_i\}_{i=1}^m \subset \mathcal{X}$

Step 3
Define $\psi : \mathcal{M} \to \mathbb{R}^m$ with

$$\psi_i(A,t) := \frac{1}{\sqrt{m}} \left[\delta\left(R(x_i) - R(y_i)\right) - \delta\left(T(Ax_i + t) - T(Ay_i + t)\right)\right],$$

$i = 1, \ldots, m$ and calculate $C := D(\psi \circ \mu_{(A_k,t_k)})(0)$ and $d := \psi(A_k, t_k)$.

$\vdots$

(Step 1, 4, 5 and 6 are equal to the ones in Table 6.2.)

development (see e.g. [107] for an adaptive gradient algorithm). But this task was always treated as an optimization problem on a vector-space, (i.e. the special orthogonal group was usually parameterized with Euler-angles,) and not as an optimization problem on Lie groups. Similar to the previous subsection, we also want to introduce a stochastic Gauss-Newton version for the multimodal registration problem. Since the approximation of the mutual information has not the form of a squared norm, this approach is only possible for the delta-distance, introducted in Section 3.3.3. Now, for given images R, T and a window-function $\delta \in C(\mathbb{R}, \mathbb{R})$, the Gauss-Newton step $(\tilde{\Omega}(A,t), \tilde{v}(A,t))$ is performed for the function $\Psi : \mathcal{M} \to \mathbb{R}^m$ with the components

$$\psi_i(A,t) := \frac{1}{m} \left[\delta\left(R(x_i) - R(y_i)\right) - \delta\left(T(Ax_i + t) - T(Ay_i + t)\right)\right],$$

where x_i and y_i are uniformly and independently chosen sample-points in the region of interest. We summarize the algorithm in Table 6.3. Similar to the algorithm in Table 6.2, the limit of the iteration is a zero of the vector field $F(A,t) = E(\tilde{\Omega}(A,t), \tilde{v}(A,t))$ with probability 1. Again the problem appears that there is, in general, a gap between the zeros of $F(A,t)$ and the critical points of the actual cost function

$$f(A,t) := \frac{1}{|\mathcal{X}|} \sum_{x,y \in \mathcal{X}} \left[\delta\left(R(x) - R(y)\right) - \delta\left(T(Ax + t) - T(Ay + t)\right)\right]^2.$$

With the same argument as in Section 6.1.1, this gap seems to be negligible in the case of image registration.
Due to the delta-distance the window-function δ appears in the algorithm as an additional degree of freedom. In our experiments we favor the approach $\delta(x) = B_3(\sigma x)$, where $\sigma \in \mathbb{R}^+$ depends on the particular modalities. Moreover, σ also depends on the scaling of the image. The toy example in Fig. 3.6 might lead to the assumption that the result of the algorithm is independent of the particular choice of σ. However, for real medical data, a bad choice of σ can lead to quite incorrect results.

6.1.3 Non-Rigid Registration

In this subsection we want to extend the previous algorithms in the Tables 6.1, 6.2 and 6.3 to the non-rigid (or better non-affine) registration problem. Beside the classification of this task in mono- and multimodal, we also have to distinguish between parametric and non-parametric registration. The latter one searches the best deformation in the space of all diffeomorphism of the Euclidean space $\mathbb{R}^n$. Although this interesting problem has undergone extensive studies (cf. e.g. [82, 33, 34, 30]) most applications still deal with the parametric version of non-rigid registration and we will follow this - more pragmatic - approach:
For defining the space of transformations, we follow the construction in [89]. For the sake of simplicity, we set the region of interest to $Q = [0, 1]^n \subset \mathbb{R}^n$. Let B^k denote the B-splines of order $k \in \mathbb{N}$ as defined in Section 3.2 and let $\lambda \in \mathbb{N}$. Then, we parameterize deformations $\varphi_w : \mathbb{R}^n \to \mathbb{R}^n$ by

$$\varphi_w(x) := x + \left(u_w^1(x), \ldots, u_w^n(x)\right)^\top$$

with

$$u_w^i(x) := \sum_{j_1,\ldots,j_n=-\lfloor\frac{k}{2}\rfloor}^{\lambda+\lfloor\frac{k}{2}\rfloor} w_{i,j_1,\ldots,j_n} B^k\left(\lambda x_1 - j_1, \ldots, \lambda x_n - j_n\right), \quad i = 1, \ldots, n, \tag{6.5}$$

where w is a real-valued multi-array of appropriate size. The bounds of the sum take into account the support of the B-spline; cf. p.52.
This construction has the advantage that it yields a finite-dimensional, affine parameterization of a subspace of all deformations. With the parameter λ we can vary from more coarse to finer deformation subspaces and the parameter k controls the smoothness of the deformation. However, the space of such deformations $\{\varphi_w \mid w_{i,j_1,\ldots,j_n} \in \mathbb{R}\}$ for fixed λ and fixed k contains elements which are not even injective. In fact, most of the deformations in this set are considered as non-physical or simply as senseless. Therefore, minimizing the cost function

$$f(w) := \mathfrak{D}(R, T \circ \varphi_w) \tag{6.6}$$

is, in general, not promising. Another reason for rejecting this cost function is that the corresponding optimization problem is, in general, ill-posed; cf [10].
A classical loophole is to equip the cost function with a penalty term

$$f(w) := \mathfrak{D}(R, T \circ \varphi_w) + \alpha\|\mathcal{A}\mathrm{vec}(w)\|^2, \tag{6.7}$$

Table 6.4: Stoch. Gradient Alg. for Monomodal, Non-Rigid Image Registration

Given: $R, T \in C^1(\mathbb{R}^n, \mathbb{R})$ and a set of voxels $\mathcal{X} \subset [0,1]^n$
Parameter: $m \in \mathbb{N}$, $a, b \in \mathbb{R}^+$, $\omega \in (0.5, 1]$, $k, \lambda \in \mathbb{N}$, $\mathcal{A} \in \mathbb{R}^{p\times nq^n}$ with $q = \lambda + 2\lfloor\frac{k}{2}\rfloor$, $\alpha \in \mathbb{R}^+$.

Step 1
Make an initial point $w_0 \in \mathbb{R}^{n\times q\times\ldots\times q} \cong \mathbb{R}^{nq^n}$. Set $j = 0$.

Step 2
Chose a uniformly and independent distributed set of sample-points $\{y_i\}_{i=1}^m \subset \mathcal{X}$

Step 3
Set $\Psi(w) := \frac{1}{m}\sum_{i=1}^m \left[R(y_i) - T(\varphi_w(y_i))\right]^2 + \alpha\|\mathcal{A}w\|^2$ and calculate $\tilde{v} := \nabla\Psi(w_k)$.

Step 4
Set $\gamma_j := \frac{a}{(b+j)^\omega}$ and calculate $w_{j+1} := w_j - \gamma_j\tilde{v}$.

Step 5
Set $j = j + 1$ and goto Step 2.

where vec(w) denotes the lexicographical order of w into a vector of the size np^n with $q = \lambda + 2\lfloor\frac{k}{2}\rfloor$, $\alpha \in \mathbb{R}^+$ and $\mathcal{A} \in \mathbb{R}^{p\times nq^n}$ is a user-defined matrix. One might simply set $\mathcal{A} = I$, which would mean that large distortions are panelized in (6.7). In our experiments, we use a different form of $\mathcal{A}$: For each tuple of integers $J := (j_1, \ldots, j_n) \in \left[-\lfloor\frac{k}{2}\rfloor,\ \lambda + \lfloor\frac{k}{2}\rfloor\right]^n$ we define the multi-arrays $X_{1,J}, \ldots, X_{n,J}$ of the size $q^n = q \times \ldots \times q$ by

$$(X_{s,J})_{l_1,\ldots,l_n} = \begin{cases} 1 & \text{if } (l_1, \ldots, l_n) = J \text{ and } j_s \neq \lambda + \lfloor\frac{k}{2}\rfloor \\ -1 & \text{if } (l_1, \ldots, l_s + 1, \ldots, l_n) = J \text{ and } j_s \neq \lambda + \lfloor\frac{k}{2}\rfloor \\ 0 & \text{else.} \end{cases}$$

Then, we use all vectors of the form vec$(X_{s,J})$ to create the columns of the matrix $\mathcal{B}$ (we can use an arbitrary order for this construction). The matrix $\mathcal{A} := I_n \otimes \mathcal{B}^\top$ is then defined by the Kronecker product of the identity matrix I_n and $\mathcal{B}^\top$. With this construction, the set of translations are not penalized in the cost function (6.7). To see this, one should keep in mind that φ_w is a translation if and only if all entries in the subarrays $(w_{l,i_1,\ldots,i_n} | i_k = 1, \ldots, q, k = 1, \ldots, n)$ are equal, for all $l = 1, \ldots, n$. Thus $\mathcal{A}$vec$(w) = 0$. We refer to [10] for other, usually physically motivated, penalization terms.

With this specification at hand, the cost function (6.7) and thus the non-rigid registration problem becomes well defined, we only have to chose the image distance measure $\mathfrak{D}$. For instance, in the monomodal case, we might pick the SSD-measure and get

$$f(w) := \frac{1}{N}\sum_{i=1}^N \left[R(y_i) - (T \circ \varphi_w)(y_i)\right]^2 + \alpha\|\mathcal{A}\text{vec}(w)\|^2, \tag{6.8}$$

Table 6.5: Stochastic Gauss-Newton Algorithm for Monomodal, Non-Rigid Registration

Given: $R, T \in C^1(\mathbb{R}^n, \mathbb{R})$ and a set of voxels $\mathcal{X} \subset [0,1]^n$
Parameter: $m \in \mathbb{N}$, $a, b \in \mathbb{R}^+$, $\omega \in (0.5, 1]$, $k, \lambda \in \mathbb{N}$, $\mathcal{A} \in \mathbb{R}^{p \times nq^n}$ with $q = \lambda + 2\lfloor \frac{k}{2} \rfloor$, $\alpha \in \mathbb{R}^+$, $\Xi \subset \mathbb{R}^{n \times q \dots \times q} \cong \mathbb{R}^{nq^n}$ compact neighborhood of 0.

Step 1
Make an initial point $w_0 \in \mathbb{R}^{n \times q \times \dots \times q} \cong \mathbb{R}^{nq^n}$. Set $j = 0$.

Step 2
Chose a uniformly and independent distributed set of sample-points $\{y_i\}_{i=1}^m \subset \mathcal{X}$

Step 3
Define $\psi : \mathbb{R}^{n \times q \times \dots \times q} \to \mathbb{R}^{m+p}$ with $\psi_i(w) := \frac{1}{m}\left(R(y_i) - T(\varphi_x(y_i))\right)$ for $i = 1, \dots, m$ and $(\psi_{m+1}(w), \dots, \psi_{m+p}(w))^\top := \sqrt{\alpha}\mathcal{A}w$. Calculate $C := D\psi(w)$ and $d := \psi(w)$.

Step 4
Solve the linear equation $(C^\star \circ C) \circ [\tilde{v}] = -C^\star \circ [d]$ in the unknowns $\tilde{v} \in \mathbb{R}^{nq^n}$

Step 5
If $\tilde{v} \notin \Xi$, set $j = j + 1$ and goto Step 2. Else, set $\gamma_j := \frac{a}{(b+j)^\omega}$ and calculate $w_{j+1} := w_j - \gamma_j \tilde{v}$.

Step 6
Set $j = j + 1$ and goto Step 2.

where the sum is taken over all N pixels $\{y_i\}_{i=1}^N$ of the region of interest $Q = [0,1]^n$. We can subject the first summand to the same approximation as we did in the affine registration case in Section 6.1.1. The resulting stochastic gradient algorithm is summarized in Table 6.4. We want to point out that this algorithm is well known in image processing, and also the variation where the mutual information measure is taken for $\mathfrak{D}$. It is nothing more than a direct application of the original Robbins-Monro algorithm applied to the cost function (6.8). In particular, the set of the deformations φ_w forms no Lie group and the convergence of this algorithm is justified by classical stochastic optimization techniques. In Section 6.2, we will use this method for a comparison.

In the cases, where we chose the SSD-measure or the delta-distance for $\mathfrak{D}$, the cost function f in (6.8) preserves the form of a squared norm, even if we incorporate the penalty term. Hence, we can, once more, construct a stochastic Gauss-Newton method for minimizing f. We just apply the same idea as in Section 6.1.1 to vector space optimization. The final algorithm is summarized in Table 6.5. However, one problem appears by switching from affine to non-rigid registration: The system of the linear equation in Step 4 of Table 6.5 can be quite large.

This was not a problem in the previous Gauss-Newton methods in Table 6.3 and 6.2. There we had to handle at most 11 unknowns, since the image dimension is at most 3. This kind of equations can be handled by a Gauss-elemination. But the number of unknowns in Step 4 of Table 6.5 can easily exceed 1000. In this case, we use a Krylov subspace method for solving the linear system (cf. [108] for an introduction).

As we can see in Table 6.5, a couple of new parameters appear in the non-rigid case: In our experiments, we set the parameter k to 3. But the value λ is more problematic. It controls the degrees of freedom in the registration and thus, the size of the linear equation in Step 4. Therefore, if λ is unnecessarily large, the algorithm would slow down. In [83] a deterministic Levenberg-Marquard version for non-rigid registration was studied. There, it is mentioned that because of practical reasons $q = \lambda + 2\lfloor \frac{k}{2} \rfloor$ should not be greater than 7 for 3-dimensional images. Although our approach is slightly different, this seems to be a good guideline. Perhaps the most problematic parameter is the regularization value $\alpha \in \mathbb{R}^+$. The form of the cost function (6.7) might remind one about a Tikhonov regularization scheme. Thus, it might be better to replace α by a sequence $\{\alpha_l\}_{l\in\mathbb{N}}$, converging to zero, and solve the minimization problem for each α_l. However, if α_l becomes to small, the system of linear equations in Step 4 becomes ill-conditioned. One should also keep in mind that the penalty term has a physical interpretation. In medical application, the requested transformation is usually not a minimum of the function in (6.6) but a minimum of (6.7) for some, unknown parameter α and matrix $\mathcal{A}$.

We end this subsection with the remark that the algorithms in the Tables 6.4 and 6.5 are, of course, extendable to the non-rigid multimodal registration. We simply have to change the function Ψ and ψ, respectively, in the Step 3 of the particular algorithm, like it was done in Section 6.1.2. This time, we skipped the details.

6.2 Experimental Results

In the previous section, we presented several algorithms for the different facets of the image registration task and each one of those methods has an inherent random aspect. We already mentioned that this stochastic component is an artificial supplement to the classical, deterministic methods and that the corresponding optimization problem shows no random aspect at all. For instance, the choice of the stochastic perturbations U_k in (6.2) is an additional degree of freedom: Instead of choosing m randomly sample-points, we could also simply take all N pixels in the region of interest, which would imply $U_k = 0$ for all $k \in \mathbb{N}$. In this section, we will justify this approach. We will see that the random aspect has two consequences. First, it will make the convergence rate worse. It is known in the literature that the speed of convergence is only $\mathcal{O}(\frac{1}{\sqrt{k}})$ for the classical Robbins-Monro algorithm, where k denotes the number of iterations (see e.g. [56] p. 112 and the references therein). This is extraordinary slow in comparison to the quadratic or the superlinear convergence rate of various well-known deterministic Newton methods. On the contrary, the numerical cost per iteration is reduced dramatically. This is not surprising since the number of image evaluations has the biggest contribution to the computation time. Hence, in the case of multimodal registration where we have to incorporate all pairs of pixels in an image, the benefit of the stochastic methods become particularly apparent.

If we weigh up both effects, we still achieve a reduction of the computation time: Usually, the

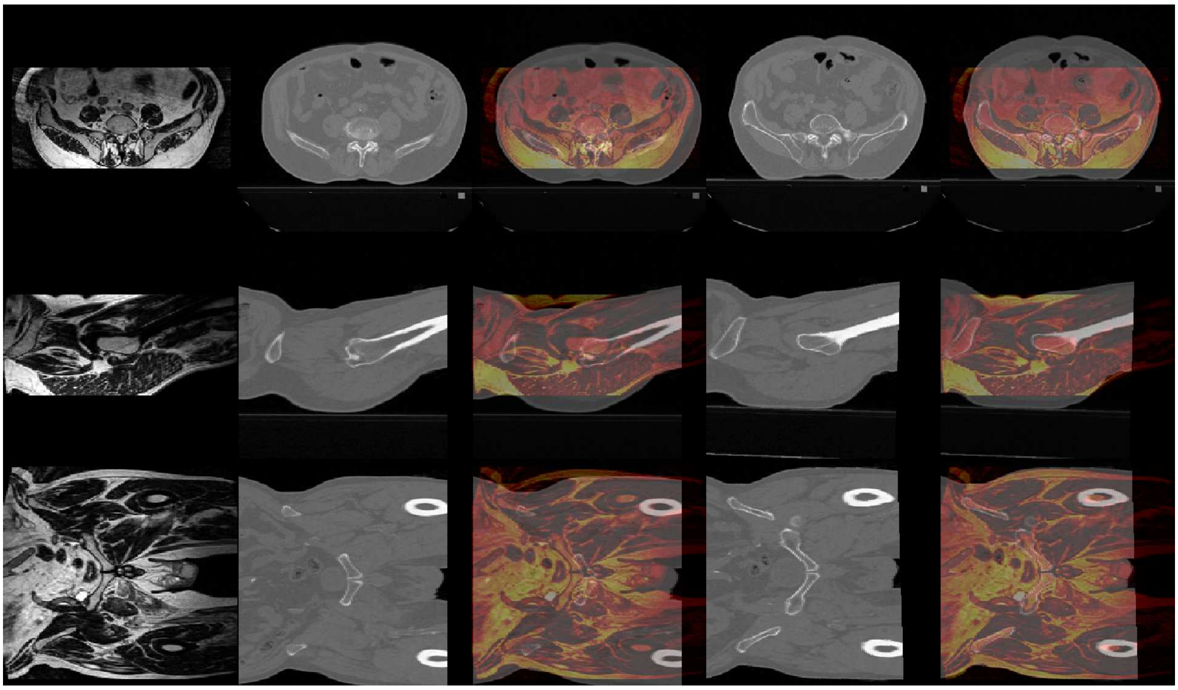


Figure 6.1: Rigid registration of a 3D CT data set with a 3D MR data set. In the three rows, we plot cross-sections of the particular data set along xy, yz, and xz planes. In the first column, we present the MR reference data. In the second column, the CT template data are shown and in the third column, we overlay the reference and the template (the gray values of the reference are presented in yellow and red, the ones of the template in black and white). The fourth column shows the template, transformed by the result of a deterministic registration algorithm (benchmark). In the fifth column, we overlay the reference with the transformed template data.

stochastic methods provide sufficiently good results, even before the first iteration of a QMC or SB-Newton algorithms is done. However, the random aspect causes also uncertainties: In contrast to the deterministic approaches, we have no domain of attraction, here. Hence, even if the initial point of the iteration is very close to the optimum, the iteration may diverge. Even if the sequence converges, we may get a worse (or a better) result if we restart the iteration with exactly the same initial values. We will examine these effects in the following.

6.2.1 Rigid Registration methods in Comparison

In this subsection, we compare the stochastic Gauss-Newton algorithm for multimodal rigid registration (cf. Table 6.3) with two standard stochastic registration methods, which are introduced by Wells et al. [88] and by Klein et al. [107], respectively.

In the first example, the reference R consists of a 3D data set of $384 \times 384 \times 187$ voxels, which was generated by magnetic resonance imaging of a pelvis. The template T shows the same region in a resolution of $512 \times 512 \times 111$ voxels, but it was generated by x-ray computer tomography. The range of the gray-values is $[0, 726]$ for the reference and $[0, 3856]$ for the template. The task is to find the best Euclidean transformation (A, t) such that the reference $R(x)$ matches the transformed template $T(Ax + t)$. Initially, we choose a $172 \times 368 \times 175$

region of interest in the reference and calculate the partial derivatives of T in each voxel.
The first two columns of Fig. 6.1 show the initial position of both data sets. We plot cross-sections of the particular data set along the xy, yz, and xz planes (and we use the same planes for each column in Fig. 6.1). We overlay both data sets in the third column. In the following, this initial position is used to compare several stochastic registration algorithms.
Before we can start our comparison, we have to know the true result of this problem from the beginning. For this experiment the result of the deterministic method of Unser and Thévenaz [90] plays the role of such a benchmark transformation. We choose this method, since it was also favored in [107] and [52]. This algorithm discretizes the gray value range (we use an equidistant discretization with 43 nodes for both data sets) to get an approximation of the mutual information. Additionally, we do not involve all voxel-information in the region of interest, but we reduce the calculations to a grid of small mesh-size, i.e. $35 \times 74 \times 36$. Hence, the used benchmark is already subject to several approximations. However, the calculated benchmark transformation yields a -visually- satisfying registration of both data sets as one can see in the fourth and fifth column of Fig. 6.1 and the method of Unser and Thévenaz [90] is also well-accepted in the literature. The fourth column of Fig. 6.1 shows the resulting transformed template of the deterministic approach and in the fifth column, we overlay the reference with the transformed template data. For the reader it might be helpful to focus on the structure of the bones in both sets. We also want to point out that this algorithm minimizes a certain approximation of the mutual information measure. The approximation used by Wells et al. [88] is already different to the one in [90] and also our stochastic Gauss-Newton method minimizes a different cost function. Thus, one might expect little differences in the particular results and it is not clear which one is really the best.
The aim of this experiment is to compare the accuracy and the speed of several stochastic methods for this registration task shown in Fig. 6.1. We start each stochastic registration algorithm with the initial value $M_0 := (A_0, t_0)$, $A_0 = I$ and $t_0 = (0, 0, 0)^\top$, stop them after $k = 50, 150, 500, 1000$ iterations and measure the distance between the kth iteration $M_k = (A_k, t_k)$ and the benchmark $M^\star := (A^\star, t^\star)$. We use the Frobenius-norm $\|A_k - A^\star\|_{\mathrm{Fro}}$ for measuring the rotation-part and the Euclidian norm $\|t_k - t^\star\|_2$ for the translation part, respectively. For each k, we repeat this procedure 30 times and visualize the distances in form of boxplots in Fig. 6.2. We refer to [109] for an introduction of the boxplots. We carry out this process for the following methods and specifications:

- First, we take the stochastic Gauss-Newton method of Table 6.3 to minimize the δ-distance between both data sets. In each iteration step, we use $m = 2000$ randomly chosen sample-points in the region of interest. The window-function in Step 3 is set to $\delta(x) := B^3(100x)$. Since there is no one-to-one correspondence between the gray-values and the different sort of tissues in the MR data set - the reader might already recognize some darker and brighter areas in the reference image - we do not choose the samples x_i and y_i in Step 2 of Table 6.3 independently. In our implementation, y_i is restricted to be in a neighborhood of x_i, i.e. $\|x_i - y_i\|_\infty < 30$. We take the following specifications for the parameter of the length of the step-size in Step 4 of Table 6.3: $\omega = 1$, $a = 1300$ and $b = 1$. The results are presented in the first line of Fig. 6.2.

- We repeat the previous experiment with the same specifications, but using $m = 12000$ sample-points in each iteration step. The results are presented in the second line of

Fig. 6.2.

- Next, we run the registration task with the adaptive stochastic gradient algorithm, presented by Klein et al. [107]. Here, the update step $(\tilde{\Omega}_k, \tilde{v}_k)$, which is the gradient of the mutual information measure, is stretched or compressed, depending on the sign of $\left\langle (\tilde{\Omega}_k, \tilde{v}_k), (\tilde{\Omega}_{k+1}, \tilde{v}_{k+1}) \right\rangle$. Moreover, they presented a method for estimating the best value for the parameter a in the gain-sequence $\gamma_k = \frac{a}{(b+k)^\omega}$ of the Robbins-Monro algorithm. For $m = 2000$ sample-points per iteration, the method yields $a = 735$. We set $b = 1$ and $\omega = 1$. This method is based on a stochastic optimization algorithm developed by Plakhov and Cruz [110]. However, it is mentioned in [107] that the necessary assumptions for this algorithm are partially not and partially only approximately satisfied in the special case of image registration. Thus, a formal proof of the convergence of this registration method is missing, although this method produces quite appealing results. This gap appears in all stochastic registration algorithms and also our presented Gauss-Newton methods make no exception (we mentioned in Section 6.1.2 the gap between the limiting point and the critical points of the cost function). We used the same discretization of the gray-value range as in the benchmark method. In fact, the cost function in both methods are the same. Since the method, presented by Klein et al. [107], is based on an optimization technique on a vector-space, we have to choose a suitable parameterization of the Euler angles and the translation. We use

$$(a, b, c, d, e, f) \mapsto \exp \begin{pmatrix} 0 & \frac{a}{\sigma} & \frac{b}{\sigma} & d \\ -\frac{a}{\sigma} & 0 & \frac{c}{\sigma} & e \\ -\frac{b}{\sigma} & -\frac{c}{\sigma} & 0 & f \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{6.9}$$

  An exhaustive experimental search yields the best result for $\sigma = 1000$. The results are shown in the third line of Fig. 6.2.

- We repeat the previous experiment with the same specifications, but using $m = 175$ sample-points in each iteration step. The estimation method for the gain sequence yields $a = 77$ for $b = 1$ and $\omega = 1$. The results are presented in the fourth line of Fig. 6.2.

- In order to demonstrate the influence of the parameter σ, we also repeated the experiment for $\sigma = 500$ and $m = 2000$ and plot the result in the fifth line of Fig. 6.2.

- Finally, we run the registration task with the method proposed by Viola and Wells [88], who lay the foundations for various publications in medical image registration (cf. [10] and the references therin). The implementation is taken from Modersitzki [10]. Here, the mutual information is approximated via (3.15). The variance matrix in (3.15) is chosen to be a multiple of the identity matrix $\Sigma = \tau I$. If $m = 400$ is fixed, $\tau = 5 \cdot 10^{-5}$ yields a good workflow (which is the result of another exhaustive search). However, it is mentioned in [10] that the parameters m and τ are not decoupled. Therefore, it is nearly impossible to create a benchmark transformation with a deterministic version for this specific cost function. Once again, we ignore this discrepancy and use the previous benchmark transformation for a comparison. We choose the following specifications for the parameter in the gain-sequence $\gamma_k = \frac{a}{(b+k)^\omega}$ of the Robbins-Monro algorithm: $\omega = 1$, $a = 5000$ and $b = 1$. The result is shown in the last row of Fig. 6.2.

Since the introduced registration algorithms vary in the cost function and the optimization method, a direct comparison is rather difficult: First, we want to point out that it is not fair to take the same number m of sample-points per iteration for each method. In our implementation of the δ-distance, we use only the differences $R(x_i) - R(y_i)$ and $T(Ax_i + t) - T(Ay_i + t)$, for $i = 1, \ldots, m$, while in each implementation of the mutual information, differences of the form $R(x_i) - R(y_l)$ and $T(Ax_i + t) - T(Ax_l + t)$, for $i, l = 1, \ldots, m$ are used (cf. eq. (3.15)). Hence, an increasing of m leads to more computational costs for the mutual information, than for the δ-distance. We list the time for the calculation of $k = 5000$ iterations in the following table.

Algorithm	Gauss-Newton $m = 2000$	Gauss-Newton $m = 12000$	adap. Gradient $m = 2000$	adap. Gradient $m = 175$	Gradient $m = 400$
Time	27.8s	214.1s	239.0s	32.5s	53.4s

With this table at hand, it is more reasonable, in terms of computation time, to compare the stochastic Gauss-Newton method using $m = 2000$ (i.e. the first row in Fig. 6.2) with the adaptive stochastic gradient descent method using $m = 174$ (i.e. the fourth row of Fig. 6.2). Similarly, we should compare the second and the third row of Fig. 6.2. Hence, in our experiments, the stochastic Gauss-Newton method shows a faster convergence to the benchmark transformation than the adaptive stochastic gradient method does. Yet, the stochastic Gauss-Newton method has more outliers as one can see in the first row of Fig. 6.2. Additionally, a variation of the initial transformation (A_0, t_0) gives rise to the suspicion that the catchment area is smaller than for the stochastic gradient methods.

One of the main advantages of the stochastic Gauss-Newton method is its use of local charts. For the other methods we have to find previously a good ratio-coefficient σ (cf. eq.(6.9)) of the rotation and translation part. As it is shown in the fifth row of Fig. 6.2, a slightly worse choice (here, we took $\frac{\sigma}{2}$) of this parameter has big influence on the convergence. Higher changes of σ usually has the effect, that the method either does not converge, or that the rotation part stagnates in the initial transformation. This effect was already mentioned in literature (cf. eg. [107]). Usually, they overcome this problem with an exhaustive search of the best value for σ. However, such problems do not arise in the stochastic Gauss-Newton method in Table 6.3.

The presented non-adaptive stochastic gradient method of Wells et al. [88] shows the worst registration results. For us, the main reason for taking it in this experiment is the relation between the δ-distance and the mutual information measure, which is pointed out in Proposition 3.3. However, we did not find a way to adapt the adaptive stochastic gradient approach of [107] to this approximation of the mutual information, like it has been done in the work of Klein et al. for the approximation used in [90]. In the paper of Wells et al. [88], it was mentioned that this algorithm shows also reasonable results for $m = 50$. In this experiment we were not able to confirm this lower bound.

In order to sum up, the stochastic Gauss-Newton algorithm shows the fastest convergence behavior, it even dominates the adaptive algorithm of Klein et al. The main problem of our method appears in the number of outliers, which can be seen in the first row of Fig. 6.2. Even

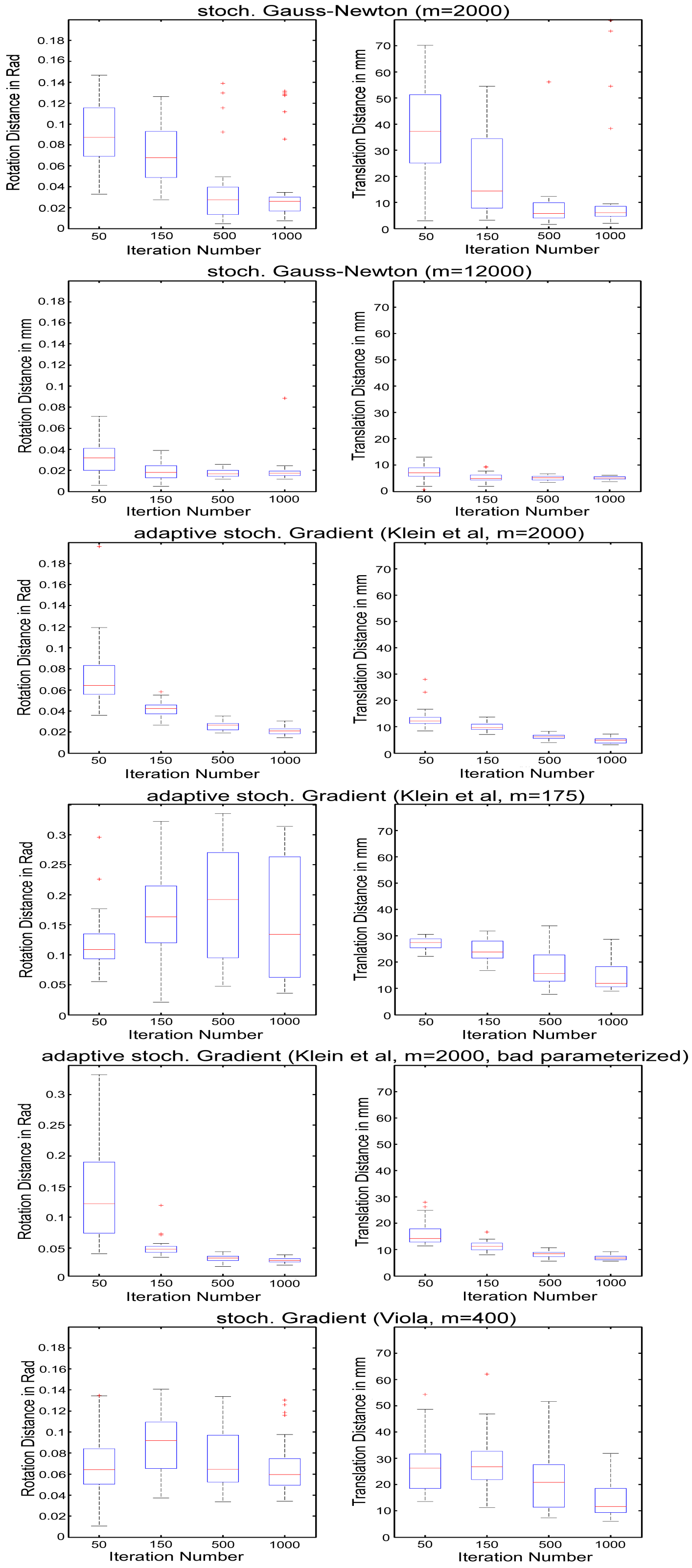


Figure 6.2: Boxplots of several stochastic optimization algorithms for the registration problem shown in Fig. 6.1 . We used a deterministic method calculate the rigid benchmark transformation $M^\star$. In the first column we show on the rotation part of the difference $M_k - M^\star$, where M_k denotes the kth iteration, and in the second column we focus on the translation part. Each boxplot is built of 30 repetition of the same experiment. m denotes the number of image-samples used for one iteration.

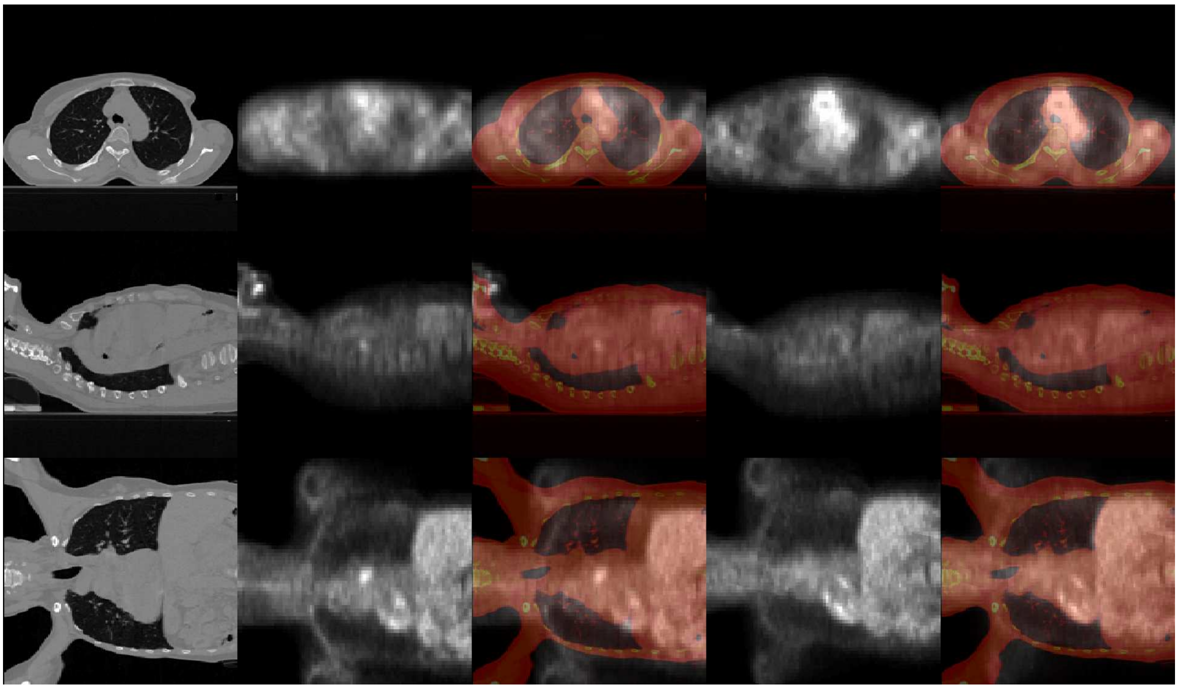


Figure 6.3: Rigid registration of a 3D CT data set with a 3D PET data set. In the three rows, we plot a cross-sections of the particular data set along xy, yz, and xz planes. In the first column, we present the PET reference data. In the second column, the CT template data are shown and in the third column, we overlay the reference and the template (the gray values of the reference are presented in yellow and red, the ones of the template in black and white.) The fourth column shows the template, transformed by the result of an deterministic registration algorithm. In the fifth column, we overlay the reference with the transformed template data.

for $k = 1000$ iterations, we might have to reject the result and restart the process. But this problem seems to be negligible if we choose a moderate number of sample-points for each step (see e.g. the second row of Fig. 6.2). In terms of the accuracy, our method and the one of Klein et al. shows similar good results. For both methods, it is in almost all cases impossible to identify - visually - differences to the benchmark transformation after $k = 1000$ iterations, if we choose a moderate size of sample-points. This might be surprising since both methods minimize different cost functions. Thus, this experiment does not only verify usefulness of the stochastic Gauss-Newton method, but also the usefulness of δ-distance for multimodal registration tasks.

In our next experiment, we examine a PET-CT registration task. The reference (CT) data set consists of $512 \times 512 \times 149$ voxel and we choose a $389 \times 492 \times 183$ cuboid out of it for the region of interest. The template (PET) data set consists of $128 \times 128 \times 168$ voxels. We want to mention that in this experiment the resolution of both data-set varies extremely. One voxel represents a $0.7 \times 0.7 \times 3.0$ mm^3 cuboid in real space for the CT-data and a $5.1 \times 5.1 \times 5.1$ mm^3 cuboid for the MR-data respectively. Again, we smooth both data sets with a $7 \times 7 \times 7$ Gaussian kernel. The PET-data has a gray-value range of $[0,\ 32766]$ while the CT-data range is of $[0,\ 4095]$. The initial position of the reference with respect to the template

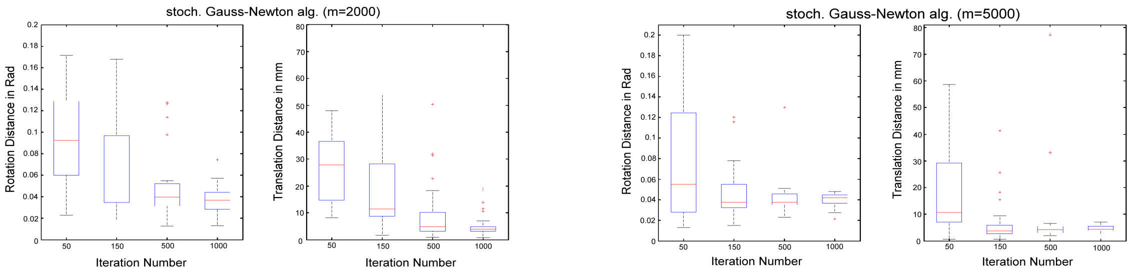


Figure 6.4: Boxplots of several stochastic optimization algorithms for the problem described in Fig. 6.3. We used a deterministic method detect the rigid benchmark transformation $M^\star$. In the first and second figure, we focus on the rotation and translation parts of the difference $M_k - M^\star$, where M_k denotes the kth iteration, and in the second column we focus on the translation part. We took $m = 2000$ image evaluations per iteration. Each boxplot is built of 30 repetition of the same experiment. In the third and fourth figure, we repeat this registration with $m = 5000$ iteration.

is shown in the third column of Fig. 6.3. Again, we create a benchmark transformation with the deterministic method of Unser and Thévenaz [90], using the same discretization of the region of interest and the gray-value range as in the first example. The result can be seen in the fourth column of Fig. 6.3. In the fifth column we overlay the transformed template with the reference.

Just like in the first experiment, we start each stochastic registration algorithm with the initial value $M_0 := (A_0, t_0)$, $A_0 = I$ and $t_0 = (0,0,0)^\top$, stop them after $k = 50, 150, 500, 1000$ iterations and measure the distance between the kth iteration $M_k = (A_k, t_k)$ and the benchmark $M^\star := (A^\star, t^\star)$. We carry out this process for the following methods:

- First, we run this registration task with the stochastic Gauss-Newton method for $m = 2000$ image evaluations per iteration. We choose the window-function $\delta(x) = B^3(1000x)$ and we make the same restriction to the sample-points, i.e. $\|x_i - y_i\|_\infty < 30$, as in the previous experiment. The statistic of this registration process is shown on the left of Fig. 6.4.

- We repeat the previous experiment with the same specifications, but by using $m = 5000$ sample-points in each iteration step. The result is presented on the right of Fig. 6.4.

The first conclusion that we get from this experiment is a similar convergence-behavior as in the CT-MR registration. However, there is a gap between the final result of the stochastic Gauss-Newton method $M^\star_{\mathfrak{D}_\delta}$ and the deterministic, benchmark method $M^\star$:

$$M^\star_{\mathfrak{D}_\delta} \approx \begin{pmatrix} 0.9999 & -0.0029 & 0.0149 & 138.3 \\ 0.0029 & 1.0000 & 0.0015 & 166.8 \\ -0.0149 & -0.0015 & 0.9999 & 249.4 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad M^\star = \begin{pmatrix} 0.9993 & -0.0372 & 0.0068 & 142.7 \\ 0.0371 & 0.9993 & 0.0077 & 166.2 \\ -0.0071 & 0.0074 & 0.9999 & 246.6 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

We get the approximation for $M^\star_{\mathfrak{D}_\delta}$ by averaging over 30 evaluation of M_{5000}. The difference in the translation part of $M^\star_{\mathfrak{D}_\delta}$ and $M^\star$ can be neglected since it is smaller than the size of one voxel in the PET data set. The gap in the rotation part of $M^\star_{\mathfrak{D}_\delta}$ and $M^\star$ about 1.7° degree seems to be more problematic. The reason for this might be found in the different cost functions used for the stochastic and the deterministic methods; the later one uses the mutual information measure. But there is also no guarantee that the benchmark $M^\star$ is closer to the original, or "true" transformation than our stochastic result does, since the several approximation steps were needed to calculate $M^\star$. Moreover, this approach of Unser and Thévenaz is an optimization method on vector-spaces and we already documented the problems which appear by applying these methods on manifolds. Finally, we want to mention that the applicability of the δ-distance depends on whether condition (3.16) is satisfied or not. Especially in the PET data-set we are far away form the situation that we can assign each sort of tissue an unique gray-value. But even for such a registration task, the stochastic Gauss-Newton method produces reasonable results.

6.2.2 Non-Rigid Registration Methods in Comparison

In this subsection, we apply the non-rigid registration algorithms developed in Section 6.1.3 to a patient-to-patient registration task. Both data sets are x-ray computed tomography. The first column of Fig. 6.5 shows cross-sections of the 3D reference data. It was taken from the torso of an older patient and consist of 256 × 256 × 125 voxels. However, we reduce the region of interest and choose a 186 × 188 × 84 cuboid, such that the focus is more on the thorax. The template data set consists of 256 × 256 × 109 voxels. It was taken from the thorax of a younger patient. Both data sets share approximately the same range of gray values $[0,\ 4095]$. We smooth both data sets with a Gaussian kernel of the size 17 × 17 × 17 before the start of registration processes.

In order to compare several non-rigid registration methods, we make use of the following measure: We take a fixed grid $Z \subset ROI$ of 50 × 50 × 30 equidistant sample-points and calculate

$$\frac{1}{|Z|}\sum_{z\in Z}\left(R(z) - T(\varphi_w(z))\right)^2 \tag{6.10}$$

for a given tensor w. Each algorithm begins with $w = 0$ and is stopped after 50, 100, 150, 200 and 300 iterations. Afterwards, we calculate (6.10) for the currently detected tensor w. The graphics in Fig. 6.6 show the boxplots which are built of 50 repetitions of this procedure. In each iteration, we take $m = 400$ sample-points of the region of interest. Since this problem is a monomodal task, we carry out this process for the following methods and specifications:

- First, we take the stochastic Gauss-Newton algorithms of Table 6.5. Hence, the set of transformation is parameterized with B-splines of order $k = 3$, like it was introduced in (6.5). For the unknown tensor w, we pick the size $3 \times 9 \times 9 \times 8$. The penalty term of the cost function (6.7) is constructed as in Section 6.1.3, with the specification $\alpha = 2$. The results are presented in the upper left picture of Fig. 6.6.

- We repeat this experiment with the same specifications, but using the stochastic gradient algorithm of Table 6.4. The results are presented in the upper right picture of Fig. 6.6.

Yet, we have considered the task as a monomodal registration problem, but we can also ignore the fact that both images are made by the same modality and apply the stochastic methods for multimodal registration:

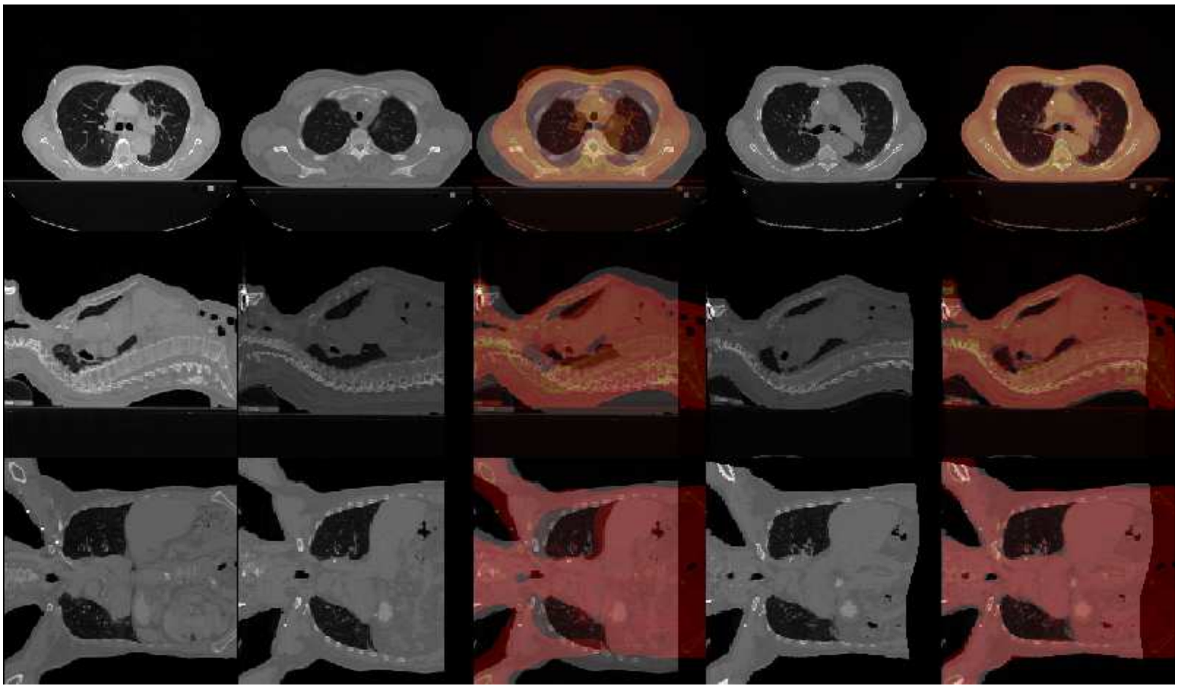


Figure 6.5: Non-rigid registration of two 3D CT data sets. One is taken of an older (reference data) and one of a younger man (template data). In the three rows, we plot cross-sections of the particular data set along xy, yz, and xz planes. In the first column, we present reference data. In the second column template data are shown and in the third column, we overlay the reference and the template (the gray values of the reference are presented in yellow and red, the ones of the template in black and white.) The fourth column shows the template, transformed by the result of an deterministic registration algorithm. In the fifth column, we overlay the reference with the transformed template data.

- We pick the stochastic Gauss-Newton method for non-rigid multimodal registration, as it was proposed in Section 6.1.3. We take $m = 1000$ sample-points per iteration and use the window-function $\delta(x) = B^3(5 \cdot 10^{-4})$, the parameter of the gain-sequence are set to $a = 2$, $b = 1$ and $\omega = 0.602$. We use the same initial data and the same B-spline parameterization of the set of transformations as in the previous experiments. The results are presented in the lower right picture of Fig. 6.6.

- Finally, we compare the previous methods with the classical stochastic gradient algorithm for the cost function (6.7), with the mutual information measure for $\mathfrak{D}$. Here, the approximation of the entropy is done by the approach of Unser and Thévenaz [90], where we also used $m = 1000$ sample-points. The remaining parameters are set to $a = 90000$, $b = 1$ and $\omega = 0.602$. The results are presented in the lower left picture of Fig. 6.6.

One result of this experiment is that the stochastic Gauss-Newton method has a faster convergence behavior than the stochastic gradient method. However, one should keep in mind that we have to solve a quite large linear equation in each iteration of Table 6.5. Our optimization problem, for instance, handles 1994 unknowns. Hence, a stochastic Gauss-Newton step needs more computation time than a stochastic gradients step. For instance, 300 stochastic

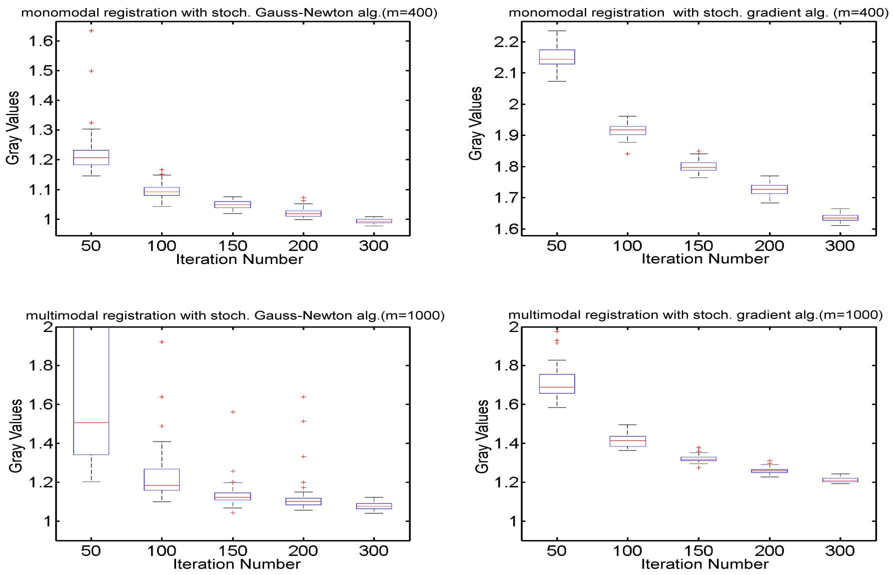


Figure 6.6: Boxplots of the stochastic Gauss-Newton (left) and the stochastic gradient algorithm (right) for the non-rigid registration problem described in Fig. 6.5. In the first row, we tackle the problem as monomodal registration and take $m = 400$ image evaluations per iteration. Both algorithms are used to minimize the SSD-measure, which is equipped with an additional penalty term. In the second row, we tackle the problem as a multimodal registration task and take $m = 1000$ image evaluations per iteration. The Gauss-Newton method (left) is used to minimize the δ-distance and the gradient algorithm (right) minimizes the MI-distance. In all four cases, we stop the algorithm after 50, 100, 150, 200 and 300 steps, calculate the gray-values difference and repeat this procedure 50 times. Each boxplot is built from these 50 repetitions.

Gauss-Newton iterations take 263 sec on MATLAB R2010b running on a PC with Intel Core 2 processor (3,0 GHz, 6 MB). We use the `bicgstabl`-command in MATLAB to solve the system of linear equation in Step 4 (which is an implementation of a Krylov subspace method). Comparatively, the stochastic gradient method needs only 123 sec. But even if we take this delay into account, the stochastic Gauss-Newton method still shows the faster convergence behavior.

In a direct comparison of the multimodal experiments, we observe a faster speed of convergence in the case of the Gauss-Newton method. However, the stochastic gradient algorithm has less outliers and gives better results in the first few iteration than the other method. In contrast to the first experiment, the difference of both methods in the computation time per iteration is nearly negligible: the stochastic Gauss-Newton 300 iterations take 373 sec whereas 391 sec are needed in the stochastic gradient case. Hence, once more we get a net profit in time, if we switch from the stochastic gradient to the stochastic Gauss-Newton method.

Appendix A

The Quasi-Monte Carlo Method

In Section 5.1.1 the problem of how to evaluate the higher-dimensional integral in the cost function (5.2) appeared. We solved this problem with the QMC-Newton algorithm by evaluating the integrand in a couple of sampling-points given by the so-called Halton-sequence. This yielded a good approximation to the cost function. In this section, we provide a short insight into the underlying theory.

Let f be a real-valued function, defined on a d-dimensional domain Q. For the sake of simplicity, we set $Q = [0,1]^d$. We consider the task of the numerical integration of f over Q. A general approach for solving this problem has the form

$$\int_{[0,1]^d} f(x)dx \approx \frac{1}{N} \sum_{i=1}^{N} a_i f(x_i), \tag{A.1}$$

for appropriate $a_i \in \mathbb{R}$ and $x_i \in \mathbb{R}^n$. In the case of $d = 1$, the Simpson rule, for instance, uses $a = \frac{1}{3}(1\ 4\ 2\ 4\ 2 \ldots 4\ 1)$ and an equidistant distribution for the sample-points $x_i = i/N$. This yields an approximation error of order $\mathcal{O}(N^{-4})$. It is also possible to extend the method to higher dimensions $d > 1$, creating a so called *product rule* (cf.[111]). Here, the sample-points x_i form a Cartesian grid in $[0,1]^d$ and the approximation error has the order $\mathcal{O}(N^{-4/d})$. Thus, if we increase the dimension d in our abstract numerical integration problem, N has to grow exponentially, if the approximation error is not to explode. This phenomenon is mentioned in literature as the *curse of dimensionality.* Of course, there are also other methods in literature for the numerical integration in one dimension, like the Gauss rule. Nevertheless, the same phenomenon appears when we extend this method to higher dimensions.

One way to avoid this *curse of dimensionality* is to use the Monte Carlo method for the approximation (A.1). Here, the sample-points x_i are independent and identical distributed random vectors in $[0,1]^d$ and $a_i = 1$ for $i = 1, \ldots, N$. The resulting approximation error is of order $\mathcal{O}(N^{-1/2})$. In particular, the order is independent of the dimension d, which is a crucial property of high dimensional integration. However, there is yet a better way of solving the problem. There are a couple of deterministic sequences $\{x_i\}_{i=1}^N \subset [0,1]^d$ in the literature where the approximation error has the order $\mathcal{O}(\frac{(\log N)^d}{N})$. One example of such a sequence is given by Halton [100]. Other sequences are known from Sobol, Faure and Niederreiter; cf. [101]. Such deterministic sequences $\{x_i\}_{i=1}^N$ with $a_i = 1$ for $i = 1, \ldots, N$ in (A.1) are sometimes referred to the so-called Quasi-Monte Carlo methods.

Example: Halton-sequence

Let $p \in \mathbb{N}$. Then we can express any other integer n a basis in p as follows:

$$n = \sum_{j=0}^{m} a_{i,n} p^j, \quad \textit{with} \quad a_{i,n} \in \{0, 1, \ldots, p-1\},$$

where $m = \lfloor \log_p n \rfloor$. Therefore, we get a fraction $\Phi_p(n) \in [0, 1]$ if we reveres the digits in n:

$$\Phi_p(n) := \sum_{j=0}^{m} a_{i,n} \frac{1}{p^{j+1}},$$

which is known as the van der Corput sequence. The Halton-sequence in d dimensions is then given by

$$x_i = \left(\Phi_{p_1}(i), \ldots, \Phi_{p_d}(i)\right)^\top,$$

where $p_1, \ldots, p_k$ are prime to each other.

We do not want to give the impression that the integration with the Halton-sequence yields, without exception, better results than the Monte-Carlo method. Even though the approximation error of the Halton-sequence is asymptotically better than the one of the Monte Carlo method, it can take quite many sample points to reach this advantage. In the literature it is suggested to use the Quasi Monte-Carlo methods only for moderate dimensions $d \leq 15$ (see e.g. [111]). However, Traub and Paskov compared in [112] numerically (but without analysis of the convergence) Quasi-Monte Carlo methods with Monte Carlo methods for a mathematical finance problem. In this specific case, the Quasi-Monte Carlo method showed a faster convergence throughout, even for an integration in 360 dimensions.

Index

- $B^k(x)$, *see* B-Spline
- $C^r(\mathcal{M},\mathcal{N})$, 66
- $C^r_S(\mathcal{M},\mathcal{N})$, *see* strong topology
- $C^r_W(\mathcal{M},\mathcal{N})$, *see* weak topology
- $G \cdot \mathcal{X}$, Orbit of $\mathcal{X}$, 71
- $J^r(M,N)$, *see* Jet
- $SA(n)$, *see* Special Affine Group, *see* Special Euclidean Group
- $SE(n)$, *see* Special Euclidean Group
- $SO(n)$, *see* Special Orthogonal Group
- S_Φ, 33
- $\mathbb{R}^+$, 33
- $\mathbb{L}_{\text{sym}}$, 70
- $\mathcal{L}(A,[a,b])$, *see* total variation
- $\mathcal{P}_{n,l}$, 74, 100
- $\mathcal{S}^k_\lambda$, 52
- $\mathcal{X}$, set of sample-points, 71–74, 80, 132
- $\mathcal{X}^G$, Orbit of $\mathcal{X}$, 72, 73
- $\mathfrak{D}_\delta$, *see* Image Distance Measure
- $\mathfrak{D}_{\text{MI}}$, *see* Image Distance Measure
- $\mathfrak{D}_{\text{SSD}}$, *see* Image Distance Measure
- $\text{CEmb}^r_S(\mathcal{M},\mathcal{N})$, *see* closed embedding
- $\text{Emb}^r_S(\mathcal{M},\mathcal{N})$, *see* embedding
- $\text{Prop}^r_S(\mathcal{M},\mathcal{N})$, *see* proper
- $\phi_{R,\mathcal{X}}$, 71
- $\pitchfork$, *see* Transversality
- ρ_M, *see* Transformation, Affine
- $\rho_{(A,t)}$, *see* Transformation, Affine
- $\triangle_X$, 83
- *ie*, image evaluations, 124, 132
- j^r_x, *see* Jet
- $sa(n)$, Lie algebra of $SA(n)$, 18
- $se(n)$, Lie algebra of $SE(n)$, 22

- B-Spline, 52, 137

- closed embedding, 72, 86
- Complete Vector Field, 33, 36, 41
- Computerized Tomography (CT), 9
- CT, *see* Computerized Tomography

- embedding, 72, 86
- eval, 71

- Gain Sequence, 36

- Halton-sequence, 95, 117
- Homogenous Coordinates, 21

- Image Distance Measure
  - δ-Distance, 57
  - Mutual Information (MI), 55
  - Sum of Squared Differences (SSD), 55
- Interpolation
  - constant, 36
  - linear, 36

- Jet, 100

- Magnetic Resonance Imaging (MR), 9
- Martingale difference Noise, 47
- Matlab Solver
  - `bicgstabl`, 150
  - `fmincon`, 122
  - `fminunc`, 121
  - `knitro`, 124
  - `snopt`, 123
- MR, *see* Magnetic Resonance Imaging
- MTF, 37
- Multiresolution, 63

- Newton method
  - (μ,ν), 23
  - (μ,ν)-Newton, 30, 32
  - QMC, 96, 97
  - SB, 108
  - SQP, 122

Perfect Match, 57
PET, *see* Proton Emission Tomography
prolongation map, 70
proper, 72, 86
Proton Emission Tomography (PET), 9
Pseudotrajectory, asymptotic, 34

Semiflow, 33
Special Affine Group, 18
Special Euclidean Group, 21, 22
Special Orthogonal Group, 21
Stochastic Optimization Method
  Gauss-Newton, 134, 136, 139
  general, 47
  gradient, 48, 133, 138

Topology
  strong, 66
  weak, 66
Transformation
  Affine, 21
  Euclidean, 22
Transversality, 66

Variation, total, 38

Window-Function, 57, 137, 142, 147, 149

List of Tables

2.1 Approximate-Newton Algorithm for SE(n) Group Actions 30
2.2 Approximate-Newton Algorithm for SA(n) Group Actions 32
2.3 Root-Finding Scheme for Vector Fields on Manifolds 47

3.1 Characterizations of a perfect match for different image distance measures. . 58
3.2 Multiresolution Approach for Image Registration 63

4.1 Geometric formulation of Condition 4.1 for $SE(n)$ and $SA(n)$. 82

5.1 QMC-Newton Registration-Algorithm on SE(n) 96
5.2 QMC-Newton Registration-Algorithm on SA(n) 97
5.3 Mutual Information-based Registration-Algorithm on SE(n) 113
5.4 Mutual Information-based Registration-Algorithm on SA(n) 114

6.1 Stoch. Gradient Alg. for Monomodal, Affine Image Registration 133
6.2 Stochastic Gauss-Newton Algorithm for Monomodal, Affine Image Registration 134
6.3 Stochastic Gauss-Newton Algorithm for Multimodal, Affine Image Registration 136
6.4 Stoch. Gradient Alg. for Monomodal, Non-Rigid Image Registration 138
6.5 Stochastic Gauss-Newton Algorithm for Monomodal, Non-Rigid Registration 139

Bibliography

[1] B. Fischer and J. Modersitzki. Ill-posed medicine-an introduction to image registration. *Inverse Problems*, 24(1):1–19, 2008.

[2] J. P. W. Pluim, J. B. A. Maintz, and M A Viergever. Mutual-information-based registration of medical images: A survey. *IEEE Transaction on Medical Imaging*, 22(8):986–1004, 2003.

[3] P. M. Evans. Anatomical imaging for radiotherapy. *Physics in Medicine & Biology*, 53:R151–191, 2008.

[4] U. Nestle, W. Weber, M. Hentschel, and A. L. Grosu. Biological imaging in radiation therapy: role of positron emission tomography. *Physics in Medicine & Biology*, 54:R1–25, 2009.

[5] M. L. Kessler. Image registration and data fusion in radiation therapy. *British Journal of Radiology*, 79(8):99–108, 2006.

[6] D. A. Jaffray and J. H. Siewerdsen. Cone-beam computed tomography with a flat-panel imager: initial performance characterization. *Medical Physics*, 27:1311–1323, 2000.

[7] D. A. Jaffray, J. H. Siewerdsen, J. W. Wong, and A. A. Martinez. Flat-panel cone-beam computed tomography for image-guided radiation therapy. *International Journal of Radiation Oncology, Biology, Physics*, 53:1337–1349, 2002.

[8] J. Pouliot. Megavoltage imaging, megavoltage cone beam CT and dose-guided radiation therapy. *Frontiers of Radiation Therapy & Oncology*, 40:132–142, 2007.

[9] J. Pouliot, A. Bani-Hashemi, J. Chen, M. Svatos, F. Ghelmansarai, M. Mitschke, M. Aubin, P. Xia, O. Morin, K. Bucci, M. Roach 3rd, P. Hernandez, Z. Zheng, D. Hristov, and L. Verhey. Low-dose megavoltage cone-beam CT for radiation therapy. *International Journal of Radiation Oncology, Biology, Physics*, 61:552–560, 2005.

[10] J. Modersitzki. *Numerical Methods for Image Regisration*. Numerical Mathematics and Scientific Computation. Oxford University Press, New York.

[11] Frédéric Cao. *Geometric Curve Evolution and Image Processing*. Number 1805 in Lecture Notes in Mathematics. Springer Verlag, New York, Berlin, Heidelberg, February 2003.

[12] A. Mencarelli, S. Van Beek, L. J. Zijp, C. Rasch, M. Van Herk, and J. Sonke. Automatic detection system for multiple region of interest registration to account for posture changes in head and neck radiotherapy. *Performance Evaluation*, pages 1–4, 2010.

[13] I. Isgum, M. Staring, A. Rutten, M. Prokop, M. A. Viergever, and B. van Ginneken. Multi-atlas-based segmentation with local decision fusion - application to cardiac and aortic segmentation in CT scans. *IEEE Transactions on Medical Imaging*, 28(7):1000–1010, 2009.

[14] E. Kerkhof, S. Ramani, M. Kessler, A. Antonuk, D. McShan, and j. Balter. Development of an atlas to aid in MRI-guided treatment planning: Initial estimation of requirements. In *XVIth International Conference on the Use of Computers in Radiation Therapy*, 2010.

[15] S. Angenent, E. Pichon, and A. Tannenbaum. Mathematical methods in medical image processing. *Bulletin of the American Mathematical Society*, 43:365–396, 2006.

[16] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, New York, NY, USA, 2 edition, 2003.

[17] G. Simone, A. Farina, F. C. Morabito, S. B. Serpico, and L. Bruzzone. Image fusion techniques for remote sensing applications. *Information Fusion*, 1(1):3–15, 2002.

[18] B. F. Hutton and M. Braun. Software for image registration: Algorithms, accuracy, efficancy. *Seminars in Nuclear Medicine*, 33(3):180–192, 2003.

[19] M. Urschler, H. Ditt, and H. Bischof. Partially rigid bone registration in CT angiography. *Proc. of the Computer Vision Winter Workshop*, pages 34–39, 2006.

[20] Jeff Orchard. Globally optimal multimodal rigid registration: An analytic solution using edge information. *IEEE International Conference on Image Processing*, 1(16):485–488, 2007.

[21] T. C. Pataky, J. Y. Goulermas, and R. H. Crompton. A comparision of seven methods of within-subjects rigid-body pedobarographic image registration. *Journal of Biomechanics*, 41:3085–3089, 2008.

[22] P. A. Woerdeman, P. W. A. Willems, H. J. Noordmans, C. A. F. Tulleken, and J. W. B. Van der Sprenkel. Application accuracy in frameless image-guided neurosurgery: A comparison study of three patient-to-image registration methods. *Journal of Neurosurgery*, 106:1012–1016, 2007.

[23] G. Li, H. Xie, H. Ning, D. Citrin, J. Capala, R. Maass-Moreno, P. Guion, B. Arora, N. Coleman, K. Camphausen, and R. W. Miller. Accuracy of 3D volume image registration based on CT, MR and PET/CT phantom experiments. *Journal of Applied Clinical Medical Physics*, 9(4):17–36, 2008.

[24] P. Thévenaz, U. E. Ruttimann, and M. Unser. A pyramid approach to sub-pixel registration based on intensity. *IEEE Transactions on Image Processing*, 7(1):27–41, 1998.

[25] S. Lee, M. Choi, H. Kimm, and F. C. Park. Geometric direct search algorithms for image registration. *IEEE Transaction on Image Processing*, 16(2):2215–2224, 2007.

[26] S. Gwak, J. Kim, and F. C. Park. Numerical optimization on the euclidean group with applications to camera calibration. *IEEE Trans. Robot. Autom.*, 19(1):65–74, 2003.

[27] U. Helmke, K. Hüper, P. Y. Lee, and J. B. Moore. Essential matrix estimation using gauss-newton iterations on a manifold. *International Journal on Computer Vision*, 74:117–136, 2007.

[28] Yi Ma, S. Soatto, J. Kosecka, and S. Sastry. *An Invitation to 3-D Vision*, volume 26 of *Interdisciplinary Applied Mathematics*. Springer, New York, Berlin, Heidelberg, 1995.

[29] P. A. Absil, R. Mahony, and R. Sepulchre. *Optimization Algorithms on Matrix Manifold*. Princeton University Press, New Jersey, 2008.

[30] M. F. Beg, M. I. Miller, A. Trouve, and L. Younes. Computing large deformation metric mappings via geodesic flows of diffeomorphisms. *International Journal Computer Vision*, 61(2):139–157, 2005.

[31] L. Younes. Jacobi fields in groups of diffeomorphisms and applications. *Quarterly of Applied Mathematics*, 65:113–134, 2007.

[32] V. Arsigny, O. Commowick, X. Pennec, and N. Ayache. A log-euclidean framework for statistics on diffeomorphisms. In *Proc. MICCAI'06*, volume 4190 of *LNCS*, pages 924–931. Springer-Verlag, 2006.

[33] M. Hernandez, M. N. Bossa, and S. Olmos. Registration of anatomical images using paths of diffeomorphisms parameterized with stationary vector field flows. *Int. J. Comput. Vision*, 85:291–306, December 2009.

[34] T. Vercauteren, X. Pennec, A. Perchant, and N. Ayache. Diffeomorphic demons: Efficient non-parametric image registration. *NeuroImage*, 45(1):61–72, 2009.

[35] C. Papazov and D. Burschka. Stochastic global optimization for robust point set registration. *Comput. Vis. Image Underst.*, 115:1598–1609, 2011.

[36] S. Gold, A. Rangarajan, C. Lu, and E. Mjolsness. New algorithms for 2d and 3d point matching: Pose estimation and correspondence. *Pattern Recognition*, 31:957–964, 1997.

[37] S. Rusinkiewicz and M. Levoy. Efficient variants of the icp algorithm. In *INTERNATIONAL CONFERENCE ON 3-D DIGITAL IMAGING AND MODELING*, 2001.

[38] S. Granger and X. Pennec. Multi-scale em-icp: A fast and robust approach for surface registration. In *European Conference on Computer Vision (ECCV 2002), volume 2353 of LNCS*, pages 418–432. Springer, 2002.

[39] J. Ho and M. Yang. On affine registration of planar point sets using complex numbers. *Computer Vision and Image Understanding*, 115(1):50–58, 2011.

[40] B. Jian and B. Vemuri. Robust point set registration using gaussian mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33:1633–1645, 2011.

[41] Y. Tsin and T. Kanade. A correlation-based approach to robust point set registration. In *In ECCV*, pages 558–569, 2004.

[42] H. Li and Richard R. Hartley. The 3d-3d registration problem revisited. In *ICCV*, pages 1–8. IEEE, 2007.

[43] H. Pottmann, Q. Huang, Y. Yang, and S. Hu. Geometry and convergence analysis of algorithms for registration of 3D shapes. *Int. J. Computer Vision*, 67(3):277–296, 2006.

[44] D. Breitenreicher and C. Schnörr. Model-based multiple rigid object detection and registration in unstructured range data. *Int. J. Comp. Vision*, 92:32–52, 2011.

[45] K. Hüper and J. Trumpf. Newton-like methods for numerical optimization on manifolds. In *Thirty-Eighth Asilomar Conference on Signals, Systems and Computers*, pages 136–139, November 2005.

[46] U. Helmke, K. Hüper, and J. Trumpf. Newton's method on Grassmann manifolds. *arXiv:0709 2205v2*, 2007.

[47] M. Benaïm and M. W. Hirsch. Asymptotic pseudotrajectories and chain recurrent flows, with applications. *Journal of Dynamics and Differential Equations*, 8(1):141–176, 1996.

[48] M. B. Wakin, D. L. Donoho, S H. Choi, and R. G. Baraniuk. The multiscale structure of non-differentiable image manifolds. In *Proc. Wavelets XI at SPIE Optics and Photonics*, 2005.

[49] M. Alder. Lie group transformations of objects in video images. *J. Math. Imaging Vis.*, 26:73–84, 2006.

[50] M. W. Hirsch. *Differential Topology*, volume 33 of *Graduate Texts in Mathematics*. Springer, Berlin, Heidelberg, New York, 1976.

[51] J. C. Spall. Adaptive stochastic approximation by the simultanious perturbation method. *IEEE Transactions on Automatic Control*, 45(10):1839—1853, 2000.

[52] K. Klein, M. Staring, and J. Pluim. Evaluation of optimization methods for nonrigid medical image registration using mutual information and b-splines. *IEEE Transactions on Image Processing*, 16(12):2879–2890, 2007.

[53] U. Helmke and J. B. Moore. *Optimization and Dynamical Systems*. CCES. Springer, London, 1994.

[54] M. Shub. Some remarks on dynamical systems and numerical analysis: Proceedings of the VII ELAM. pages 69–91. Univ. Simon Bolivar, Caracas, 1986.

[55] J. H. Manton. Optimization algorithms exploiting unitary constraints. *IEEE Transactions on Signal Processing*, 50(3):635–650, 2002.

[56] J. Spall. *Introduction to Stochastic Search and Optimization.* John Wiley & Sons, Hoboken, New Jersey, 2003.

[57] H. J. Krushner and G.G. Yin. *Stochastic Approximation and Recursive Algorithms and Applications.* Springer-Verlag, New York, Berlin, Heidelberg, 2003.

[58] C. Lageman. *Convergence of gradient-like dynamical systems and optimization algorithms.* PhD thesis, Universität Würzburg, 2007.

[59] G. Dirr, U. Helmke, and C. Lageman. Nonsmooth riemannian optimization with applications to sphere packing and grasping. In *Lagrangian and Hamiltonian Methods for Nonlinear Control 2006: Proceedings of the 3rd IFAC Workshop on Lagrangian and Hamiltonian Methods in Nonlinear Control*, pages 19–21. Nagoya, Japan, July, 2007.

[60] J. Hilgert and K. H. Neeb. *Lie-Gruppen und Lie-Algebren.* Vieweg, Braunschweig, 1991.

[61] E. Malis and S. Benhimane. Real-time image-based tracking of planes using efficient second-order minimization. In *IEEE/RSJ Interantional Conference on Intelligent Robots and Systems*, volume 1, pages 943–948, Sendia, Japan, October 2004.

[62] J. Jost. *Riemannian Geometry and Geometric Analysis.* Universitext. Springer-Verlag, New York, Berlin, Heidelberg, 2008.

[63] G. Köhler. *Analysis*, volume 14 of *Berliner Studienreihe zur Mathematik.* Heldermann, Lemgo, 2006.

[64] D. Hinrichsen and A. Pritchard. *Mathematical System Theory I.* Texts in Applied Mathematics. Springer-Verlag, New York, Berlin, Heidelberg, 2005.

[65] M. Benaïm. *Dynamics of stochastic approximation algorithms.* Séminaire de Probabilités XXXIII. Springer, Berlin, Heidelberg, New York, 1999.

[66] T. Breuel. Implementation techniques for geometric branch-and-bound matching methods. *CVIU*, 90:258–294, 2003.

[67] C. Olsson, F. Kahl, and M. Oskarsson. Branch and bound methods for euclidean registration problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(5):783–794, 2009.

[68] R. Duda and P. Hart. *Pattern Classification and Scene Analysis.* John Wiley and Sons Inc, New York, 1973.

[69] A. V. Oppenheim, R. V. Schafer, and J. R. Buck. *Discrete-time Digital Signal Processing.* Prentice Hall, New Jersey, second edition, 1999.

[70] I. J. Schoenberg. Cardinal interpolation and spline functions. *Journal of Approximation Theory*, 2:167–206, 1969.

[71] M.Unser, A. Aldroubi, and M. Eden. Enlargement or reduction of digital images with minimum loss of information. *IEEE Transactions on Image Processing*, 4(3):247–258, March 1995.

[72] M. Unser. Splines: A perfect fit for signal and image processing. *IEEE Signal Processing Magazine*, 16(6):22–38, 1999.

[73] J. H.Ahlberg, E. N. Nilson, and J. L. Walsh. *The Theory of Splines and their Applications*. Graduate Texts in Mathematics. Academic Press, New York, 1967.

[74] M. Unser A. Aldroubi and M. Eden. Cardinal spline filters: Stability and convergence to the ideal sinc interpolation. *Signal Processing*, 28:127–138, 1992.

[75] R. C. Gonzalez, R. E. Woods, and S. L. Eddins. *Digital Image Processing*. Eearson Education, Prentice Hall, New Jersey, 2004.

[76] M. Salotti, F. Bellet, and C. Garbay. Evaluation of edge detectors: Critics and proposal. In *Proceedings Workshop Performance Characteristics Vision Algorithms*, 1996.

[77] F. Luisier, T. Blu, B. Forster, and M. Unser. Which wavelet bases are the best for image denoising? In *Proceedings of the SPIE Conference on Mathematical Imaging: Wavelet XI*, volume 5914, pages 59140E–1–59140E–12, 2005.

[78] Y. Tsaig and D L. Donoho. Compressed sensing. *IEEE Trans. Inform. Theory*, 52:1289–1306, 2006.

[79] R. G. Baraniuk and M. B: Wakin. Random projections of smooth manifolds. *Found. Comput. Math.*, 9(1):51–77, 2009.

[80] M. Unser, A Aldroubi, M. Eden, and L. Fellow. B-spline signal processing: Part I - theory. *IEEE Trans. Signal Processing*, 41:821–833, 1993.

[81] M. Unser, A Aldroubi, M. Eden, and L. Fellow. B-spline signal processing: Part II - efficient design and applications. *IEEE Trans. Signal Processing*, 41:834–848, 1993.

[82] E. Haber and J. Modersitzki. A multilevel method for image registration. *SIAM J. Sci. Comput.*, 27(5):1594–1607, 2006.

[83] S. Kabus, T. Netsch, B. Fischer, and J. Modersitzki. B-spline registration of 3d images with Levenberg-Marquardt optimization. In *Proceedings of the SPIE 2004, Medical Imaging, San Diego Februar 14-19, 2004*, volume 5370, pages 304–313. SPIE, 2004.

[84] D. Salas-Gonzalez, J.M. Gorriz, J. Ramirez, A. Lassl, and C.G. Puntonet. Improved gauss-newton optimisation methods in affine registration of spect brain images. *Electronics Letters*, 44(22):1291–1292, 2008.

[85] E. Haber and J. Modersitzki. *Intensity gradient based registration and fusion of multimodal images*, volume 4191 of *Lecture Notes in Computer Science*. Springer, Berlin, Heidelberg, 2006.

[86] A. P. Viola. *Alignment by Maximization of Mutual Information.* PhD thesis, Massachusetts Institute of Technology, 1995.

[87] S. M. Ermakov. *Die Monte-Carlo Methode und verwandte Fragen (translated from the Russian).* R. Oldenburg Verlag, München, 1975.

[88] W. Wells III, P. Viola, H. Atsumi, S. Nakajima, and R. Kikinis. Multi-modal volume registration by maximization of mutual information. *Medical Image Analysis*, 1(1):35–51, 1996.

[89] J. Modersitzki. *Flexible Algorithms for Image Registration.* Fundamental of Algorithms. SIAM, Philadelphia, 2009.

[90] P. Thévenaz and M. Unser. Optimization of mutual information for multiresolution image registration. *IEEE Transactions on Image Processing*, 9:2083–2099, 2000.

[91] T. Lange, N. Papenberg, J. Olesch, B. Fischer, and P. M. Schlag. Landmark constrained non-rigid image registration with anisotropic tolerances. In *IFMBE Proceedings 25/IV*, pages 2238–2241, 2009.

[92] S. B. Park, J.W. Sohn, F. C. Rhee, and J. I. Monroe. Spatially weighted mutual information image registration for image guided radiation therapy. *Medical Physics*, 37(9):4590–4601, 2010.

[93] B. Forster, T. Blu, and M. Unser. Complex b-splines. *Applied and Computational Harmonic Analysis*, 20(2):261–282, 2006.

[94] O. Scherzer, M. Grasmair, H. Grossauer, M. Haltmeier, and F. Lenzen. *Variational Methods in Imaging.* Springer Publishing Company, Incorporated, New York, Berlin, Heidelberg, 2008.

[95] S. Z. Kovalsky, G. Cohen, R. Hagege, and J. M. Francos. Decoupled linear estimation of affine geometric deformations and nonlinear intensity transformations of images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(5):940–946, 2010.

[96] M. Golubitsky and V. Guillemin. *Stable Mappings and Their Singularities.* Graduate Texts in Mathematics. Springer-Verlag, New York, Berlin, Heidelberg, 1973.

[97] W. Klingenberg. *Riemannian Geometrie.* Studies in Mathematics. Walter de Gruyter, Berlin, New York, 1995.

[98] D. L. Donoho and C. Grimes. Image manifolds which are isometric to euclidean space. *J. Math. Imaging Vis.*, 23(1):5–24, 2005.

[99] V. V. Gorbatsevich, A. L. Onishchik, and E. B. Vinberg. *Foundations of Lie Theory and Lie Transformation Groups.* Springer, Berlin, Heidelberg, New York, 1997.

[100] J. H. Halton. On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals. *Numerische Mathematik*, 2:84–90, 1960.

[101] L. Kuipers and H. Niederreiter. *Uniform Distribution of Sequences.* John Wiley, New York, NY, 1974.

[102] H. Whitney. *Elementary structure of real algebraic varieties*, volume 66. 1957.

[103] R. Fletcher. *Practical Methods of Optimization.* John Wiley and Sons, Chichester, 1987.

[104] T. F. Coleman and Y. Li. An interior trust region approach for nonlinear minimization subject to bounds. *SIAM J. Optim.*, 6:418–445, 1996.

[105] C. Geiger and C. Kanzow. *Theorie und Numerik Restringierter Optimierungsaufgaben.* Springer, Berlin, Heidelberg, New York, 2002.

[106] P. Deufelhard. *Newton Methods for Nonlinear Problems*, volume 35 of *Springer Series in Computational Mathematics.* Springer, Berlin, Heidelberg, New York, 2004.

[107] K. Klein, M. Staring, J. Pluim, and M. Viergever. Adaptive stochastic gradient descent optimisation for image registration. *International Journal of Computer Vision*, 81:227–239, 2009.

[108] H. A. van der Vorst. *Iterative Krylov Methods for Large Linear Systems.* Cambridge University Press, 2003.

[109] R. McGill, J. W. Tukey, and W. A. Larsen. Vaiations of boxplots. *The American Statistician*, 32:12–16, 1978.

[110] A. Plakhov and P. Cruz. A stochastic approximation algorithm with step size adaptation. *Journal of Mathematics and Sciences*, 120(1):964–973, 2004.

[111] F. Kuo and I. Sloan. Lifting the curse of dimensionality. *Notices of the AMS*, 52:1320–1329, 2005.

[112] S. Paskov and J. Traub. Faster valuation of financial derivatives. *The Journal of Portfolio Management*, pages 113–120, Fall 1995.