

Applying hybrid evolutionary algorithms to deformable image registration of 3D medical images using common B-spline-based transformation models

J.I. Mulder

MSc Thesis

Applying hybrid evolutionary algorithms to deformable image registration of 3D medical images using common B-spline-based transformation models

by

J.I. Mulder

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Thursday April 20, 2023 at 13:00 PM.

Student number: 4492021
Project duration: September 16, 2021 – April 20, 2023
Thesis committee: Prof. dr. P.A.N. Bosman, TU Delft, Algorithmics, supervisor
Dr. T. Alderliesten, LUMC, Radiation Oncology, supervisor
Prof. dr. ir. B. Lelieveldt, TU Delft, Intelligent Systems
Dr. ir. A. Bouter, CWI, daily supervisor
Ir. G. Andreadis, LUMC, daily supervisor

An electronic version of this thesis is available at <http://repository.tudelft.nl/>.

Abstract

Deformable Image Registration (DIR) is a medical imaging process involving the spatial alignment of two or more images using a transformation model that can account for non-rigid deformations. B-spline-based transformation models have emerged as a common approach to express such spatial alignments. However, without additional measures, their flexibility can lead to physically implausible deformations. This flexibility has motivated the inclusion of penalty terms to improve the smoothness and regularity of the transformation. Determining an appropriate weight for this penalty term is difficult, as each registration problem requires a different trade-off between this penalty term and the quality of the transformed image.

Gradient-based methods are commonly used as optimization methods in medical image registration toolboxes due to their computational efficiency and fast convergence rates. However, due to their gradient-based approach, they may converge prematurely in local minima. In this thesis, we investigate the efficacy of a gradient-less alternative: the Real-Valued Gene-pool Optimal Mixing Evolutionary Algorithm (RV-GOMEA), a population-based method that can exploit the problem structure of optimization problems through explicit mappings of dependencies between problem variables. To improve the computational efficiency of RV-GOMEA when applied to DIR, we show how to apply partial evaluations for common image similarity metrics and penalty terms when using B-spline-based transformation models.

We test RV-GOMEA on a synthetic registration problem to better understand its performance in the context of DIR. Based on our findings, we propose several methods that hybridize RV-GOMEA with a gradient-based method and impose specific constraints on the B-spline-based transformation model. We validate the performance of these methods on clinical registration problems and find that RV-GOMEA with a gradient-based local search operator can provide significant benefits over purely gradient-based methods for DIR problems. Additionally, placing specific constraints on the transformation model can increase the regularity of transformations without requiring a penalty term.

Preface

I am pleased to present my Master's thesis, which investigates the application of hybrid evolutionary algorithms to deformable image registration using common B-spline-based methods. This research has been rewarding and challenging, as I have worked hard to maintain focus despite various obstacles. I am grateful for the guidance of my daily supervisors, whose support and expertise have been instrumental in bringing this project to fruition. I would also like to express my appreciation to my thesis advisors for their valuable feedback and insight during our monthly meetings.

In this thesis, I aim to assess the performance of hybrid evolutionary algorithms in improving the accuracy and efficiency of deformable image registration with B-spline-based methods. The findings of this study provide insight into the potential of hybrid evolutionary algorithms for improving deformable image registration.

I extend my thanks to the CWI, LUMC and TU Delft for their assistance in this research. Finally, I am grateful for the support of my family and friends, who have been a constant source of encouragement throughout my academic journey.

*J.I. Mulder
Delft, April 2023*

Contents

1	Introduction	1
1.1	Deformable Image Registration	2
1.2	Real-Valued Gene-pool Optimal Mixing Evolutionary Algorithm	3
1.3	Research Topics	4
1.4	Overview	4
2	Literature Review	5
2.1	Methodology	5
2.2	Deformable Image Registration	5
2.2.1	Deformation Model	5
2.2.2	Similarity Metric.	6
2.2.3	Optimization Method	8
2.3	Optimization Methods	9
2.3.1	RV-GOMEA.	9
2.3.2	UHV-Adam	9
2.3.3	Hybrid	9
2.4	Multi-Objective Deformable Image Registration	9
3	Background	11
3.1	B-spline Deformation Model	11
3.1.1	Bézier curves	11
3.1.2	B-splines	11
3.1.3	Patches & Grids	12
3.1.4	Deformation model	13
3.1.5	Properties.	13
3.2	Evolutionary Optimization	14
3.2.1	EDA	14
3.2.2	GOMEA	15
3.2.3	RV-GOMEA.	15
3.3	Multi-objective Optimization	15
3.3.1	Pareto-dominance	16
3.3.2	(Uncrowded) Hypervolume.	16
3.4	Partial Evaluations	17
3.4.1	Definition	17
3.4.2	Variable Interaction	18
4	Design	19
4.1	Partial Evaluations in Common B-spline Transformation Models	19
4.1.1	Definitions.	20
4.1.2	Mean Squared Difference	20
4.1.3	Thin-plate Bending Energy.	21
4.1.4	Zero-Normalized Correlation Coefficient	21
4.2	Linkage Models.	23
5	Implementation	25
5.1	ITK	25
5.1.1	Image representation.	25
5.1.2	Data representation	25

5.2	RV-GOMEA	26
5.2.1	Shrinkage	26
5.2.2	Inspection Output	26
5.2.3	Optimizations	27
5.3	Elastix	27
5.3.1	Analytic Regularization	28
5.3.2	Sobol Sequence	29
5.3.3	Partial Evaluations	29
5.3.4	Missed Voxels	29
5.3.5	Fold Constraints	30
5.3.6	Hybrid Local Search	31
5.3.7	OpenMP	31
5.3.8	Inspection Output	32
5.3.9	Build Process	32
6	Analysis	33
6.1	Experimental Setup	33
6.1.1	Datasets	33
6.1.2	Validation Metrics	33
6.1.3	Hardware	36
6.2	Experiments	36
6.2.1	Single-resolution Comparison to Gradient-based Optimization	36
6.2.2	Multi-resolution Comparison to Gradient-based Optimization	39
6.2.3	Linkage Models	40
6.2.4	Missed Voxels	42
6.2.5	Fold Constraints	45
6.2.6	Hybridization using Gradient-based Local Search	46
6.2.7	Regularization Weight	48
6.2.8	LEARN	50
6.3	Discussion	54
7	Conclusion and Future Work	57
7.1	Conclusion	57
7.2	Future Work	58
	Bibliography	61

Introduction

In the medical field, imaging data is often collected from patients to guide the treatment. This involves various imaging techniques, including Computed Tomography (CT), Cone Beam Computed Tomography (CBCT), Positron Emission Tomography (PET), and Magnetic Resonance Imaging (MRI). While a single image is often sufficient for diagnosis or treatment, in some cases, analyzing a pair of images of the same anatomical region can provide valuable information. For instance, comparing follow-up scans can help track the progress of multiple treatment iterations. These images may be taken at different time frames (multi-temporal), using different imaging devices (multi-modal), or of different patients (multi-subject). As a result, it is unlikely that a pair of images will immediately provide a perfect correspondence between the anatomical structures found in both images. In other words, the same voxel in different images may not refer to the same position in the body due to unintentional or intentional misalignments. Unintentional misalignments may occur due to slightly shifting organs or minor differences in patient positions during image capture [45]. Intentional misalignments can occur when certain positions or actions can provide specific insights when compared. For example, it is easier to delineate the tumor in images taken with the patient in the prone position in diagnosing breast cancer; the surgery is performed with the patient in the supine position [64].

Any misalignment can increase the difficulty of analyzing pairs or sets of images, making it essential to align the images spatially before any comparison is made. Figure 1.1 provides an example of an alignment requiring only a global source image translation. Such an alignment can be done through rigid registration, which is limited to six degrees of freedom through translations and rotations. However, if the images contain non-rigid differences, a transformation model that allows for non-rigid and elastic deformations is needed. In Deformable Image Registration (DIR), pairs of images are systematically aligned or 'registered' using a transformation model capable of modeling these deformations.

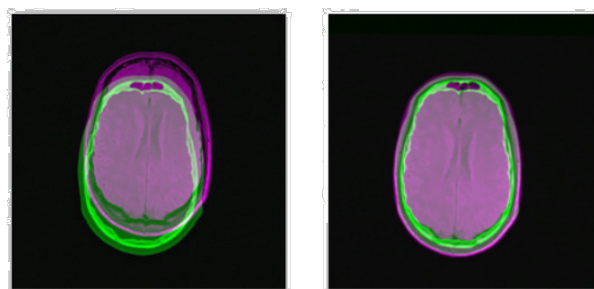


Figure 1.1: Spatial alignment of two brain MRI images.

Medical deformable image registration has shown potential for various applications in clinical practice. For instance, in the diagnosis of Chronic Obstructive Pulmonary Disease (COPD), the detection of trapped air in local lung regions could be assisted by registering an inspiration-expiration CT scan pair [38]. In Radiation Therapy (RT), registration could be used to estimate dose accumulation more accurately from treatment to treatment [60]. Additionally, organ or tumor contours could be propagated from one scan to another by applying the transformation found from their registration, making follow-up checkups easier [46].

Despite numerous potential applications, the state-of-the-art in DIR still is not mature enough for many critical clinical applications. For instance, a recent survey revealed that most clinical centers worldwide have yet to adopt DIR for dose accumulation due to the uncertainty and variability of registration results produced by current commercial DIR packages [46]. In general, it remains challenging to validate DIR algorithms for critical clinical applications [46, 60, 61, 68].

1.1. Deformable Image Registration

In this thesis we will restrict the scope of DIR to pairs of images, which can formally be described as the process of spatially aligning a moving image I_M with a fixed image I_F through a nonlinear transformation T_μ :

$$\arg \min_{\mu} C(T_\mu; I_F; I_M) \quad (1.1)$$

$$C(T_\mu; I_F; I_M) = S(I_F, I_M \circ T_\mu) + \lambda R(T_\mu) \quad (1.2)$$

Here, μ contains the transformation parameters, and C is the objective value, with S denoting the similarity between 2 images according to a registration quality metric, and R denoting the deformation magnitude which is weighted by the parameter λ . Ultimately, this formulation balances 2 objectives, the quality of alignment and the magnitude of deformation needed to achieve this quality.

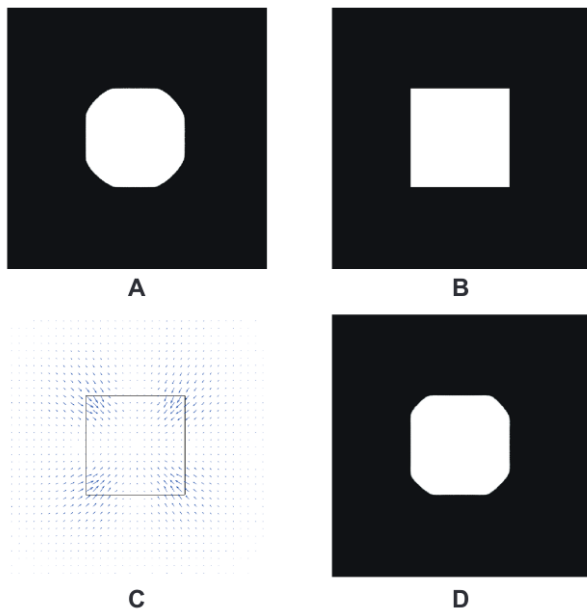


Figure 1.2: A fixed image (A) and moving image (B) that are registered by the 2D Deformation Vector Field (C), resulting in the deformed moving image (D) [60]

In DIR, the transformation model is responsible for defining the space of possible deformable transformations that can alter the anatomical structure displayed on the underlying images. While rigid transformations can only translate, scale, or rotate an imaged object, deformable transformations can alter the object's shape in non-linear ways. Figure 1.2 shows an example of such a transformation as part of a trivial registration problem, where the transformation is visualized using a Deformation Vector Field (DVF). B-splines are a commonly used transformation model in DIR because they can model these transformations while also providing computational tractability, smoothness, and a degree of physical plausibility. As a result, analytic derivations of the gradient are typically available, allowing for the use of gradient-based optimizers in the registration process. More specifically, B-splines are piecewise polynomial functions that can approximate smooth and continuous transformations, and they have been shown to provide good results in various applications of DIR [50, 84]. Additional properties of B-splines will be discussed in Section 3.1.5.

Gradient-based optimizers are commonly used in current open-source medical image registration toolboxes [49, 78]. However, DIR as an optimization problem is usually highly non-convex, containing local minima to which gradient-based optimizers are susceptible. Figure 1.3 shows the contours of an example function that contains such local minima or “valleys” where gradient-based optimizers could prematurely converge. Registration is often done using multi-resolution schemes to decrease the likelihood of premature convergence. In a multi-resolution scheme, the image and transformation grid are iteratively refined from coarse to fine to capture global transformations in the former stages and local transformations in the latter stages of these schemes. Although effective in many cases [48], these schemes do not guarantee to resolve the susceptibility of gradient-based optimizers to local optima.

Moreover, the registration objective is usually posed as a weighted combination of objectives, including the deformation magnitude as a regularization term to penalize transformations that require a large amount of energy to perform and are often less likely to be anatomically feasible. In essence, these weights allow the user to define preferences for each objective, and with each unique set of preferences, the objective function landscape changes. In other words, with each set of preferences, the optimizer could find a different solution to the same registration problem. For example, if the preference for a minimal deformation magnitude is too high, the transformations might be too constrained, leading to insufficient registration. Alternatively, if this preference is too small, the optimizer might find transformations that result in highly similar images but using high bending energy, which could be anatomically infeasible. The optimal set of weights depends on the structure of each registration problem, as some only require minor adjustments, but others include disappearing structures (e.g.,

tumors removed by surgery). As such, estimating the optimal set of weights has proven to be a difficult challenge in practice [63, 64].

If the problem is viewed without the weights, it becomes a multi-objective optimization problem. With multi-objective optimization, the goal is no longer to find a single (near-)optimal solution to a problem. Instead, we would like to find a set of non-dominating (near-)optimal solutions that provide varying trade-offs between the different objectives. They are non-dominating, as each solution is equal to, or better than, the other solutions for all objectives. The goal of multi-objective optimization is to closely approximate the theoretically optimal set of solutions, called the Pareto set. Gradient-based optimizers are usually incapable of multi-objective optimization as they optimize a single solution at a time. Moreover, if the posed problem is highly non-convex, gradient-based optimizers might leave the more promising regions in objective space unexplored due to premature convergence in local minima.

A class of optimizers called Evolutionary Algorithms (EA) have proven robust and effective at solving non-convex black-box optimization problems [15, 40]. Moreover, they are a natural fit for multi-objective optimization due to their population-based approach in combination with effective exploration operators, resulting in near-optimal and well-spread solutions across both the convex and concave regions of the objective function domain [31]. Accordingly, these EAs have seen an increase in applications for multi-objective DIR over the past decade [3, 5, 19].

1.2. Real-Valued Gene-pool Optimal Mixing Evolutionary Algorithm

In this thesis, the application of a model-based EA, the Real-Valued Gene-pool Optimal Mixing Evolutionary Algorithm [20] (RV-GOMEA) is explored for its efficacy and applicability in DIR. RV-GOMEA is the real-valued extension of GOMEA [18], a

model-based EA that can be used to solve discrete optimization problems. It can efficiently exploit the structure of a problem by using a so-called linkage structure, which describes dependencies between problem variables of an optimization problem. GOMEA can estimate these dependencies during optimization using various specialized operators, or a static structure can be specified in advance. The latter fits with the application of this algorithm to DIR since static parameters locality is present in the B-spline transformation model. Furthermore, the real-valued extension enables continuous optimization by using parameter sampling using normal distributions. A more detailed description of GOMEA and RV-GOMEA is given in Sections 3.2.2 and 3.2.3.

Other model-based EAs, such as the Covariance Matrix Adaptation Evolutionary Strategy [39] (CMA-ES), have been applied to DIR before, but with limited success. Klein et al. compared its performance against gradient-based methods on various CT images and magnetic resonance (MR) scans, using B-splines as the transformation model, and found it underperforming in both convergence rates and registration precision [48]. However, CMA-ES and RV-GOMEA differ significantly in their method of exploiting problem structure. CMA-ES uses a covariance matrix that is reduced in size to keep the optimization computationally tractable by parameterizing it with a smaller set of parameters that account for the most variance. Although successful at solving other optimization problems [40], it is hard to determine how this affects the underlying dependent variables to be optimized, especially for high dimensional problems such as in DIR. Using an explicit mapping of problem dependencies—which RV-GOMEA allows—could prove more effective at solving DIR problems.

Additionally, the variation operator used in GOMEA to explore the solution space can be exploited to substantially improve computational performance since it allows for the partial evaluation of compatible objective functions when only a subset of the problem variables is manipulated. Common B-spline transformation models naturally align with this methodology since the transformation of a local patch of voxels is evaluated using only a subset of surrounding control points. Any alteration of the coefficients of these control points would require only a re-evaluation of the corresponding influenced voxels. For a more formal and complete description of B-splines and their use in transformation models, see Section 3.1.

In order to utilize these techniques in the context of multi-objective optimization, Bouter et al. proposed an extension of Multi-Objective GOMEA [52] (MO-GOMEA), the Multi-Objective RV-GOMEA [21] (MO-RV-GOMEA). It combines MO-GOMEA with aspects of the Multi-objective Adapted Maximum Likelihood Gaussian Model [16] (MAMaLGaM), which simultaneously advances mutually exclusive sets of solutions towards the

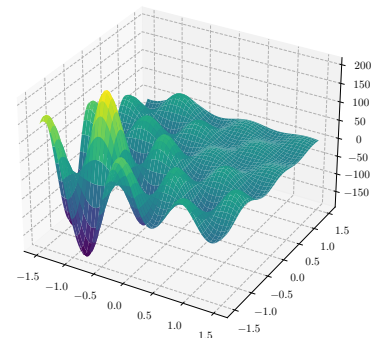


Figure 1.3: 3D contour plot of Shubert function containing multiple local minima.

Pareto front by making use of clustering, a discretized elitist archive and similar methods for Gaussian estimation and adaptation as found in RV-GOMEA. Thus far, it has seen only limited use in Multi-Objective DIR (MODIR), including a recent application using a 3D dual-dynamic grid based on simplex meshes as a transformation model for registering images including large deformations and content mismatches [5].

1.3. Research Topics

In this thesis, the following topics are explored:

1. The application of (MO)-RV-GOMEA to deformable image registration of 3D medical images when using common B-spline-based transformation models. Does it offer improvements in registration quality over more traditional gradient-based methods? Does explicitly mapping problem dependencies offer benefits over current state-of-the-art model-based EAs?
2. Partial evaluations are well suited to (MO)-RV-GOMEA and can improve computational performance dramatically. How can these be applied when using B-spline-based transformation models in combination with commonly used similarity metrics and penalty terms?
3. How does the mapping of problem dependencies defined by the linkage structure affect registration results when using B-spline-based transformation models? Can the principles of multi-resolution schemes used in DIR also be applied to these mappings?

We aim to implement our added functionality in Elastix [49], a popular medical image registration toolbox that includes implementations for common B-spline-based transformation models, a multi-resolution image registration framework, and various image similarity metrics and penalty terms.

1.4. Overview

The rest of this thesis is structured as follows. First, Chapter 2 provides an overview of relevant academic literature. Then, in Chapter 3, formal definitions and informal explanations are given for the theoretical concepts discussed throughout this thesis. Next, chapter 4 details the design of the implemented solutions, whereas Chapter 5 describes their actual implementation. In Chapter 6, the experimental design and their results and discussion are given. Lastly, the thesis is concluded in Chapter 7 with a summary and recommendations for future work.

2

Literature Review

The problem of Deformable Image Registration (DIR) is multi-faceted and can be solved in many ways. We review relevant existing literature to gain a solid understanding of the problem in general and its related fields.

The review is structured as follows: Section 2.1 describes the methods and process used in conducting the literature review, Section 2.2 gives an overview of the topic of DIR, Section 2.3 gives an overview of optimization methods relevant for this project, and Section 2.4 gives an overview on multi-objective DIR.

2.1. Methodology

The goal of the literature review is to gain a solid understanding of the problem and its related fields and a clearer understanding of the goal of the thesis by identifying a possible gap in the existing literature. This section gives a brief overview of the process.

As a first step, we find relevant surveys on the DIR field, as they already review the existing literature. We consider nine surveys from 1992 to 2020 on varying sub-topics within DIR. We distill the typical registration framework as depicted in Section 2.2. For this framework's individual components, further literature reviews are done using similar principles. Next, we examine relevant optimization methods to this thesis. Namely: RV-GOMEA [20], UHV-Adam [32], and UHV-GOMEA [56]. This is done in Section 2.3 by reviewing the original papers for these techniques and other papers on their efficacy and applicability to particular problems. Lastly, a review is done on applying the abovementioned methods to multi-objective DIR in Section 2.4. This is done by first identifying active authors in this field, after which publications from these authors are searched for directly. Forty-seven publications are included, of which 37 are referenced in this review. Table 2.1 gives an overview of the number of publications per related field.

2.2. Deformable Image Registration

DIR refers to spatially aligning a moving image I_M with a fixed image I_F through some nonlinear transformation μ . It is a subset of Image Registration (IR), where the nature of the transformation can be rigid, affine, projective, or deformable [55].

In general, registration can be performed using two or more images, but in the interest of simplicity, we will only focus on pairs of images. The typical framework for registration can be formally defined as the optimization problem defined in Section 1.1 [49]. Furthermore, we can classify a registration method using the deformation model, similarity metric, and optimization method. These classifications will be discussed in the following sections.

2.2.1. Deformation Model

Although many deformations models exist [80], in this work, we will be focusing on B-spline deformation [72]. This model has received wide acceptance in the medical imaging community [80] due to its simplicity, smooth local deformations, and fast computation due to the B-spline basis functions having local support [49]—i.e., any point on a B-spline curve is only affected by a limited number of control points.

In general, any deformation model can be classified according to its properties. These include, but are not limited to: inverse-consistency, symmetry, topology conservation, and diffeomorphism [80]. Inverse consistent

Category	Publications		
Evolutionary Algorithms	11		
	Surveys	9	
	Techniques	Deep learning	5
		Evolutionary Algorithms	9
Image Registration	Gradient-based	1	
	B-splines	6	
	Tools	1	
	Other	5	
Total		47	

Table 2.1: Literature review: overview of publications per category.

transformations simultaneously estimate both a forward and backward transformation and constrain these as inverse mappings. Symmetric transformations employ objective functions that are by nature symmetric or use two transformation functions that map to a common domain. One is then inverted and composed with the other. Topology-preserving transformations produce a continuous, onto, and locally one-to-one mapping with a continuous inverse. Finally, diffeomorphic transformations preserve topology, are invertible, and the function and inverse are differentiable. For a detailed explanation of these properties, see [80].

The B-spline deformation model is inherently asymmetric, but in [59], an inverse-consistent symmetric B-spline deformation model is proposed. It simultaneously optimizes two transformations, both a forward mapping from the moving image to the fixed image and a backward mapping from the fixed image to the moving image. Moreover, they use additional penalty terms in the objective function, which enforce inverse consistency, symmetry, and topology conservation.

Another improvement to the B-spline deformation model was made in [84], where the authors identified an issue with the gradient of the objective function when using uniform B-spline basis functions. These functions inherently create long, narrow valleys in the landscape of the objective function (see Eq. 1.2) due to the control points being weighted differently. This can lead to bad performance of typical gradient descent algorithms due to “hemstitching” – the oscillation of the gradient direction as the search zig-zags between valley walls [7]. They proposed a solution to this problem by fitting the gradient field to a prescribed space of B-spline transformations at each iteration, thereby projecting the gradient field onto a more restricted domain less prone to hemstitching.

2.2.2. Similarity Metric

The function S from Eq. 1.2 describes the similarity between two images, but there are many ways of calculating it, depending on the nature of the registration. In 1998, a classification framework for medical registration methods was proposed [55], which divides possible natures of registration as either landmark-based, segmentation-based, or voxel-based. We also include contour-based metrics here, which are similar to segmentation-based metrics but defined using sets of points around the contours of regions of interest instead of binary masks.

Landmark-based

Landmark-based methods use significant anatomical locations on the images to define similarity. For each image, a set of corresponding landmarks is defined. The goal is to achieve correspondence between these sets by spatially transforming the moving image. Given the sets of corresponding landmarks on both images, the problem can be solved using some optimization method. However, reliably locating these landmarks remains an open research question [80].

Segmentation-based

Segmentation-based methods use predefined or learned segmentations of the image to define similarity. For example, given two segmentations X, Y on both the moving and fixed image, respectively, the similarity can be

defined using the Dice similarity coefficient:

$$DSC(X, Y) = \frac{2|X \cap Y|}{|X| + |Y|}$$

where $|\cdot|$ refers to the number of voxels in a given segmentation.

Predefined segmentations are usually not readily available in clinical practice due to the labor-intensive annotation process required to acquire them. Therefore, approaches have been proposed [71, 87] that learn segmentations using deep convolutional neural networks (CNNs), which can then be used in registration.

Contour-based

Contour-based methods use sets of points C_M, C_F defined for both the moving and fixed images. These points delineate the contours of specific organs or other regions of interest in both images. Distances between pairs of sets of points C_{M_i} and C_{F_i} from the moving and fixed image can be calculated to measure the registration quality regarding how well the contours match. This is done by comparing the transformed points of the moving image contour set $T_\mu(C_{M_i})$ with the points in C_{F_i} using some distance measure.

Contour sets have been used in several registration approaches. One of these, the Thin-Plate Splines Robust Point Matching approach (TPS-RPM), deforms contours using a thin-plate spline model [37]. This method has been shown to perform well in modeling large deformations [10], but it does require regularization weights to be set prior to registration. It does not guarantee the physical plausibility of the found transformations. Another approach, called MORFEUS, provides these guarantees using a bio-mechanical model. They model the organs of interest using a tetrahedral mesh and then perform a Finite Element Analysis (FEA) to perform the registration [26]. While this increases the likelihood of finding physically plausible transformations, their method does not consider individual voxels' image intensity values and solely relies on (user-supplied) contours.

Recent methods propose a combined approach in which both contour- and voxel-based methods are used, such as the ANatomically CONstrained Deformation Algorithm (ANACONDA) [86], which uses the quasi-Newton algorithm to optimize a weighted combination of both metrics, which introduces an additional parameter which needs to be set in advance by the user.

Voxel-based

Voxel-based methods are the easiest to apply since no pre-segmentation or landmark identification is necessary to perform the registration. These methods rely on voxel intensity information to compare the images according to some similarity metric. The chosen metric depends on the registration problem's modality, which can be mono- or multi-modal.

A) *Mono-modal*: When registration is mono-modal, both images are captured using the same image acquisition method. The sum of squared or absolute differences (SSD, SAD) can be used as a similarity metric when it can be assumed that the same anatomical structures correspond to similar intensity values. If a linear relationship between the signal intensities can be assumed, cross-correlation (CC) and the correlation coefficient (CCoef) are suitable metrics [80].

B) *Multi-modal*: The images are captured using differing image acquisition methods when registration is multi-modal. This leads to the choice of similarity metric becoming less straightforward. Mutual information is often used due to its generality [80]. It does not assume any relationship between the absolute voxel intensities, thus making it a useful metric for multi-modal registration.

Mutual information for two images A, B can be defined using Shannon Entropy [76]:

$$I(A, B) = H(A) + H(B) - H(A, B). \quad (2.1)$$

Finding a registration is then equivalent to maximizing this mutual information. However, this formulation can lead to situations in which increasing misregistration leads to increasing mutual information. This can occur when the marginal entropies $H(A), H(B)$ increase faster than the joint entropy $H(A, B)$ [69]. Studholme et al. propose a less sensitive method for this issue [81], called the Normalized Mutual Information (NMI). This measure has since been used in numerous studies with promising results [69].

Further improvements to this metric can be made by, for example, considering spatial dependencies of neighboring voxel intensities. Rodriguez et al. show that this is possible by using Jumarie entropy [70], which considers the intensity differences of neighboring voxels in an image.

The interpolation method used during entropy-based registration can negatively affect the optimization process due to interpolation artifacts [67]. In addition, sudden changes in the mutual information measure can

occur when grid-aligned transformations cause local extrema. It has been shown that a slight re-sampling of one of the images can improve the smoothness of the mutual information [67].

Lastly, a hybridization of any of these types of metrics is also possible. For example, we can combine a voxel-based similarity metric with corresponding landmarks to guide the registration. Most straightforwardly, this can be done by taking a weighted combination of the combined metrics.

2.2.3. Optimization Method

Given a deformation model and similarity metric, a method needs to be specified to optimize this metric. Methods exist for both the discrete and continuous domain [80]. The deformation model in this work is defined continuously; thus, we will focus on continuous methods here. Broadly, these can be classified into three main categories: gradient-based, deep learning-based, and evolutionary-based.

Gradient-based

Gradient-based methods use the gradient of the transformation parameters, w.r.t. to the objective function, to guide the search towards an optimum. A distinction can be made based on the method being either deterministic or stochastic. Deterministic methods assume that the gradient can be computed directly, while stochastic methods only need stochastic approximations of the gradient.

In [48], a comparative study was done between both methods when using mutual information as a similarity metric and cubic B-splines as the deformation model. The authors compared deterministic gradient-based methods (Gradient Descent, Quasi-Newton, and Nonlinear Conjugate Gradient) with stochastic gradient-based methods (Kiefer-Wolfowitz, Simultaneous Perturbation, and Robbins-Monro). They found the stochastic Robbins-Monro method to be the preferred approach. Experiments were done across differing image modalities, image sizes, and hyperparameters. The Robbins-Monro method decreases the needed computation time by only using a subset of the voxels to calculate the gradient of the mutual information [48].

Deep learning-based

The previous approach still requires an iterative, slow optimization method. This motivated the development of approaches that directly predict transformations using neural networks. Deep learning-based methods have shown promising results in the field of DIR [8, 30, 41, 83]. A recent survey divides the methods into three categories: deep iterative registration, supervised transformation estimation, and unsupervised transformation estimation. We briefly describe these categories below; for an in-depth overview of these methods, see [41].

Deep iterative registration methods use deep learning methods to learn a similarity metric. This metric is then used in a traditional registration framework (e.g., Elastix [49]) to perform registration. Supervised approaches require ground truth labels to train the networks, which are challenging to obtain [41]. Unsupervised approaches use some existing similarity metric to train the network in estimating transformations directly, therefore not requiring any ground truth labels. These methods can also be feature-based by extracting features from moving and fixed images. In [30], a deep learning framework for unsupervised affine and DIR is introduced that is comparable in performance to conventional methods on tested problems while being several orders of magnitude faster.

Evolutionary-based

Evolutionary-based methods can be broadly classified as being either single- or multi-objective. All previously discussed methods are single-objective. Multi-objective approaches are less prevalent in the literature, but due to the often conflicting trade-offs in DIR problems, they provide a more complete approach at solving them.

A) Single-objective: In the single-objective setting, a single objective is defined, which the evolutionary algorithm uses during optimization to optimize a population of solutions. CMA-ES [39] is generally considered state-of-the-art in this area. However, Klein et al. compared it to gradient-based methods in [48], which showed that it provides substantially lower convergence rates when applied to typical DIR problems.

B) Multi-objective: From Eq. 1.2, we can see that the DIR problem is inherently multi-objective. In all previous methods, some λ must be chosen before the optimization process. There is, however, no theoretical basis on which to choose this parameter [2]. Multi-objective evolutionary-based algorithms remove the need for this predetermination (see Section 2.4). They have shown to be effective at solving many MO problems [52]. In [2, 3], they are applied to 2D DIR problems using both the MO-GOMEA [52], and iMAMaLGaM-X+ [17] algorithms. These algorithms deliver good results but are computationally inefficient, and thus improvements are proposed in [19, 25] by making use of the GPU and tailored linkage models. This leads to considerable speed-ups while finding approximation sets with comparable quality. A recent method extends this approach

for 3D DIR problems using a 3D dual-dynamic grid transformation model based on simplex meshes, capable of modeling large deformations while still supporting the incorporation of annotated guidance information and multi-resolution schemes [5].

2.3. Optimization Methods

Three optimization methods are discussed in this review. First, a single-objective real-valued EA, which utilizes Gene-pool Optimal Mixing. Second, a multi-objective method that derives the gradient from the Uncrowded Hypervolume (UHV) of a set of solutions to solve the problem by gradient descent. Third, a multi-objective EA in a hybrid approach with the UHV gradient ascent method.

For a more detailed overview of these methods and related background material, see Chapter 3.

2.3.1. RV-GOMEA

RV-GOMEA [20] refers to the real-valued version of the original GOMEA [18], which incorporates techniques from the real-valued Estimation of Distribution Algorithm (EDA) AMaLGaM [15]. RV-GOMEA is especially competent at solving Gray-Box Optimization (GBO) problems for which some structure can be inferred from the problem definition. This structure is formalized as dependencies between the problem variables, allowing for partial evaluation of the objective value, improving the computational performance dramatically [23]. RV-GOMEA has been used to solve many black- and GBO problems efficiently [20].

2.3.2. UHV-Adam

In [32], gradient expressions for the UHV [56] are derived and used to solve multi-objective problems using traditional gradient ascent methods. This method calculates the direction to improve a solution, which most efficiently increases the UHV (see Section 3.3.2).

This method was compared to UHV-GOMEA [56] and was found to be more sample efficient; it showed improved performance when the budget for evaluations was small but comparable performance when this budget was large. In a black-box setting, finite differences can be used to approximate the gradient, but in this setting, the method was found to be only slightly more efficient [32].

2.3.3. Hybrid

Multi-objective EAs are inherently robust against local minima due to their stochastic population-based approach. Gradient-based methods, however, are effective at exploiting local problem structures efficiently. Ideally, we would combine these strengths into a single hybrid approach. This could be done by combining UHV-GOMEA [56] with UHV-GA [32]. UHV-GOMEA is comparable to RV-GOMEA, with the main difference being that the UHV is used as the objective function, as well as some minor modifications to better align it with indicator-based multi-objective optimization problems [56].

2.4. Multi-Objective Deformable Image Registration

Gradient-based and deep-learning-based methods provide state-of-the-art performance when it comes to DIR [8, 30, 49, 89]. However, these methods all require some combination of weights for the objectives to be determined *a priori*, which has been shown to be hard [63, 64].

An approach that can circumvent this issue is to solve the DIR problem from a multi-objective perspective. This does not require the predetermination of weights and instead returns an approximation of the Pareto set: a set of solutions providing some (near) optimal trade-off between the differing objectives. This gives the decision-maker insightful context to decide on the optimal trade-off *a posteriori*.

In [65], an intermediate approach was taken, in which sets of weights for specific DIR problems were learned in an offline training phase using a multi-objective EA. An approximation of the Pareto front can be found by performing registrations for each of the weights in such a set. However, doing this online for each DIR instance would typically be too computationally intensive. Nevertheless, they found the approach effective for cases with limited deformations; cases with large deformations typically still require per-patient tuning of the weights.

In [62], gradient information was used with the multi-objective EDA iMAMaLGAM [17] when trying to solve DIR problems using a dual-dynamic B-spline transformation model. The gradient information was used by improving each solution in the population using gradient descent after each iteration. This method, however, does not use the "true" multi-objective gradient but instead performs gradient descent for one objective randomly at each iteration [62]. Still, the method showed improved results over both methods independently.

To further improve upon this approach, the UHV-GA [32] method could be used to perform gradient descent after each iteration of the EA. However, it is not immediately clear how to best integrate EAs with a gradient ascent method since both incorporate information from past iterations [32].

3

Background

This chapter provides a theoretical foundation for the rest of this thesis. First, we provide a more formal definition and explanation of B-splines and how they are used in a deformation model. Second, we give an overview of the applied evolutionary optimization methods. Third, we provide a primer on multi-objective optimization, which includes Pareto-dominance, hypervolume, and uncrowded hypervolume. Fourth, partial evaluations are formally defined.

3.1. B-spline Deformation Model

B-splines provide a tool to create complex smooth curves using only a set of control points. Their basis lies in parametric curves and, more specifically, Bézier curves.

3.1.1. Bézier curves

Bézier curves are parametric curves defined by a small set of control points c_0, \dots, c_n , and a corresponding polynomial order n . For example, given two control points c_0 and c_1 , the corresponding parametric equation is given by

$$p_1(t) = (1 - t)c_0 + tc_1.$$

Here, p_d refers to a polynomial of order d . This equation shows that the linear Bézier curve corresponds to a linear interpolation between two control points. Given three control points c_0, c_1, c_2 , we can create two linear Bézier curves:

$$\begin{aligned} p_1^1(t) &= (1 - t)c_0 + tc_1 \\ p_1^2(t) &= (1 - t)c_1 + tc_2. \end{aligned}$$

These two curves can then be linearly interpolated for $t \in [0, 1]$ to obtain a quadratic Bézier curve as follows:

$$p_2(t) = (1 - t)^2 c_0 + 2t(1 - t)c_1 + t^2 c_2.$$

We can obtain the cubic Bézier curve by linearly interpolating two quadratic Bézier curves. We can repeat this process recursively to obtain any Bézier order n by linearly interpolating two Bézier curves of order $n - 1$. See Figure 3.1 for a more intuitive visualization of this process.

3.1.2. B-splines

B-splines naturally follow from Bézier curves since they can be seen as Bézier curves joined together. Besides this, additional constraints must be imposed for the curve to be considered a B-spline curve. First, C^0 continuity states that for any Bézier curve in the sequence, the last control point must be the same as the first control point of the next curve in the sequence, if any. Second, C^1 continuity states that the first derivatives at these points must be the same. Third, C^2 continuity states that the second derivatives at these points must be the same.

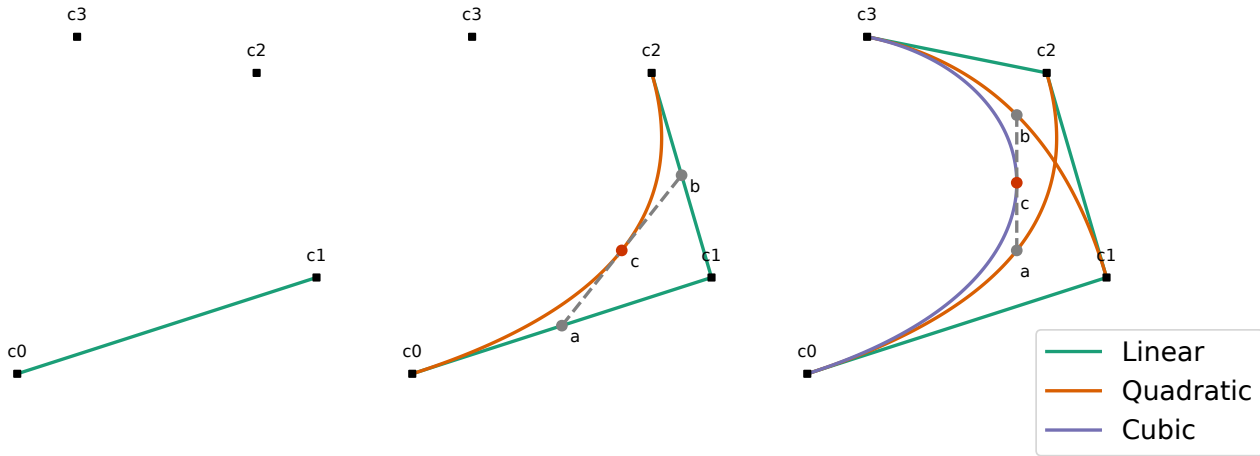


Figure 3.1: Bézier curves visualization of recursive definition as interpolation of lower order Bézier curves. Specific points c of the interpolation process are shown. To fully draw the resulting curve, one must imagine the interpolation going across the entire interval $[0, 1]$ on the lower order curves for a, b .

A d degree B-spline is a piecewise polynomial of degree d . When defined by n control points, it will consist of $n - d$ Bézier curves. It can be defined as

$$p_d(t) = \sum_{i=1}^n B_{i,d}(t)c_i, \quad (3.1)$$

where $B_{i,d}(t)$ are the basis spline functions. These are defined using the Cox-de Boor recursion formula [12].

Intuitively, B-splines can be seen as a linear combination of control points, where the basis functions give the weights. These basis functions are defined so that their combination approximates d -degree Bézier curves. The contribution of a particular control point c_i to a point on the curve depends on the parameter t . This could be zero (see [12]) due to B-spline basis functions having only local support; any point on the curve of a B-spline of degree d is only defined by $d + 1$ control points.

There are multiple advantages of B-splines over Bézier curves. First, the number of control points is not determined by the degree of the curve. With an increasing number of control points, the degree of a Bézier curve would increase too, which poses computational problems. Second, Bézier curves are always changed globally, i.e., if one control point changes, the entire curve changes, further decreasing the applicability of deformation models, especially when local deformations are required.

As stated before, only $d + 1$ terms in Eq. 3.1 are non-zero for any given t . Using this fact, we can simplify the recursive formulation into a simpler one. It is reformulated as a linear combination of only the $d + 1$ basis functions and control points, which define the current point t on the curve. The range of each basis function is now $[0, 1]$. With $i = \lfloor t \rfloor$, the new formulation is given by

$$p_d(t) = \sum_{n=0}^d B_{n,d}(t - i)c_{i+n}. \quad (3.2)$$

With this formulation, the basis functions can be defined non-recursively. For example, Table 3.1 gives simplified definitions for the linear, quadratic, and cubic degrees.

3.1.3. Patches & Grids

We need to generalize our current B-splines to a multivariate setting to arrive at a functional deformation model. In the current setting, a single parameter t and some control points generate a curve in 2D space. In deformable image registration, however, we need to have some way of transforming a grid of 2D/3D image voxels from one image space to another.

The previous univariate case can be transformed into a multivariate one by recursively replacing the control points from Eq. 3.2 with B-splines defined on control points indexed by one more variable [53]. Given that

Degree	Basis functions
Linear	$B_{0,1}(t) = 1 - t$
	$B_{1,1}(t) = t$
Quadratic	$B_{0,2}(t) = t^2/2$
	$B_{1,2}(t) = (-2t^2 + 2t + 1)/2$
	$B_{2,2}(t) = (t^2 - 2t + 1)/2$
Cubic	$B_{0,3}(t) = (-t^3 + 3t^2 - 3t + 1)/6$
	$B_{1,3}(t) = (3t^3 - 6t^2 + 4)/6$
	$B_{2,3}(t) = (-3t^3 + 3t^2 + 3t + 1)/6$
	$B_{3,3}(t) = t^3/6$

Table 3.1: Simplified B-spline basis functions for linear, quadratic, and cubic degrees.

$i = [x], j = [y]$, the bi-variate case can then be rewritten as

$$p_d(x, y) = \sum_{m=0}^d \sum_{n=0}^d B_{m,d}(x - i) B_{n,d}(y - j) c_{i+m, j+n}. \quad (3.3)$$

The resulting B-spline no longer defines a curve but a patch using a set of 2D control points.

Given that $k = [z]$, grids can similarly be defined as

$$p_d(x, y, z) = \sum_{m=0}^d \sum_{n=0}^d \sum_{l=0}^d B_{m,d}(x - i) B_{n,d}(y - j) B_{l,d}(z - k) c_{i+m, j+n, k+l}. \quad (3.4)$$

We can deform any 2D or 3D image using these patches and grids by overlaying it with control points. Then, by shifting the control points, a deformation field is defined by the B-spline patch/grid. Figure 3.2 shows how a 2D grid can deform patches locally when control points are moved after initializing them with zero displacements.

3.1.4. Deformation model

We now define the used deformation model using B-spline patches. A frequent confusion with this model is the direction in which it is defined. Previously we have stated that in the image registration problem, the moving image I_M is deformed to match the fixed image I_F . To this end, the deformation has to be defined as a coordinate mapping from the fixed image domain to the moving image domain: $T : \Omega_F \subset \mathbb{R}^d \rightarrow \Omega_M \subset \mathbb{R}^d$. This ensures that the deformed image is fully defined for all voxels; if the mapping is defined in the other direction, undefined voxels are possible due to the mapping not being one-to-one.

Given the coordinate mapping T , the deformed image can be computed as follows: for each voxel position $x \in \{x_1, \dots, x_n\}$ in the fixed image domain Ω_F , compute its mapped position $y \in \{y_1, \dots, y_n\} = T_\mu(x)$, interpolate the moving image voxel intensity at y , and use this value in the output image at x . Here, T_μ refers to a parameterized transformation using, e.g., cubic B-spline patches/grids as defined in Equations 3.3, 3.4:

$$T_\mu(x) = p_{d,\mu}(x). \quad (3.5)$$

This notation differs somewhat from the notation used by authors of the Elastix toolbox [49] as they use a specific cubic B-spline polynomial definition [85]. This toolbox and its implemented deformation models are used in this thesis; thus, the notation defined above primarily provides intuitive insight into the construction of a B-spline deformation model.

3.1.5. Properties

B-spline-based transformation models provide the following beneficial properties [11]:

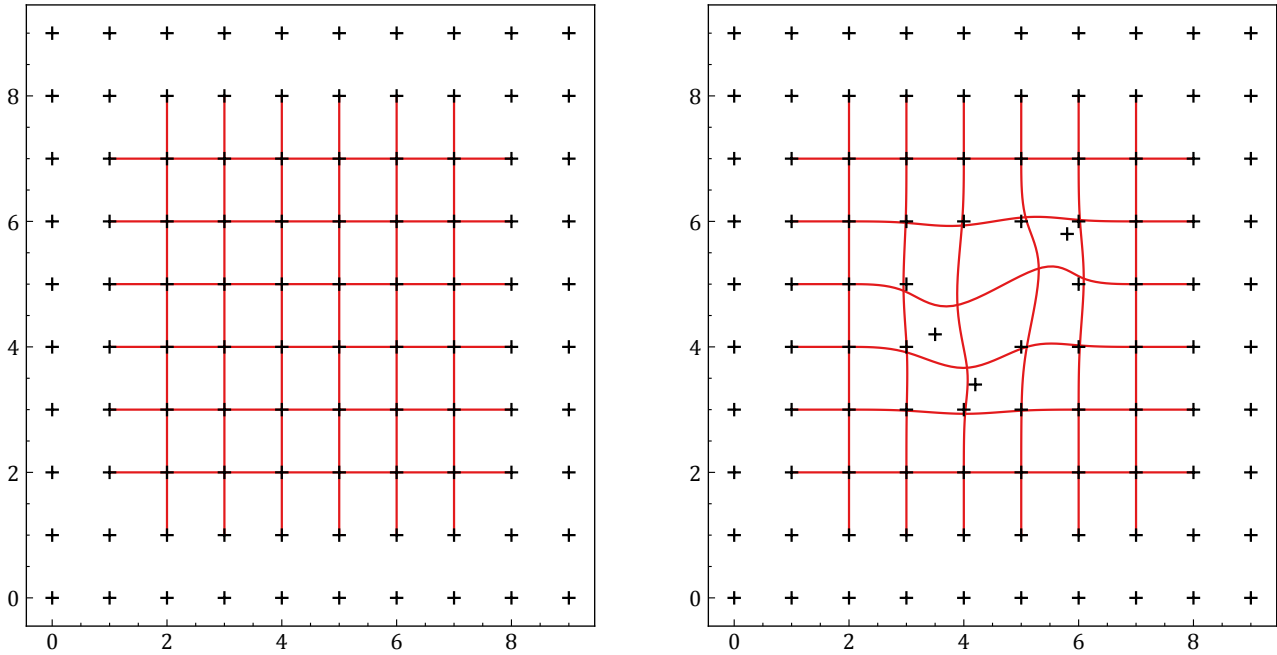


Figure 3.2: Examples of cubic B-spline patches defined by 9x9 grids of control points, indicated by the red crosses. On the left, the control points have not been moved from their initial positions. On the right, some control points are moved, which deform the patch locally.

- **Computational Tractability:** They are computationally tractable because they have local support - i.e., the basis functions are non-zero for a subset of their domain; even though their formulation might seem complex, they are by definition just a linear combination of polynomials.
- **Smoothness:** Due to their overlap in line segments, the resulting linear combination of B-splines produces a smooth spline curve of degree d , which has $d - 1$ continuous derivatives. For example, a cubic spline curve has continuous first and second derivatives (called C^2 continuity). C^2 continuity makes them well-suited for DIR optimization when gradient-based methods are used.
- **Physical plausibility:** Cubic splines minimize the "strain energy" or the euclidean norm of the second derivative of the transformation [50].

3.2. Evolutionary Optimization

Evolutionary Algorithms are nature-inspired methods for optimization, which iteratively apply variation and selection to a population of candidate solutions. In this thesis, we will focus on model-based EAs, which aim to exploit the structure of the optimization problem in some way to solve it more efficiently. We discuss two relevant techniques: Estimation of Distribution Algorithms (EDA) and Gene-pool Optimal Mixing Evolutionary Algorithms (GOMEA).

Given is an optimization problem

$$\arg \min_x f(x),$$

where f is some objective function and $x \in \mathbb{X} \subset \mathbb{R}^d$ with d the problem dimensionality.

3.2.1. EDA

An EDA aims to solve this optimization problem by identifying and using dependencies between the decision variables $\{x_1, \dots, x_d\}$ during the search, estimating a probability distribution of the subset of solutions of fixed size $S \subset X$ with minimal $\sum_{x \in S} f(x)$ for the entire population at each iteration. New solutions are then sampled using this probability distribution at the next iteration.

The AMaLGaM EDA [15] uses Maximum-Likelihood (ML) estimates of Gaussians to estimate the probability distributions. Besides this, it uses several improvements to negate some of the drawbacks of using only ML estimates. First, Adaptive Variance Scaling (AVS) and the Standard Deviation Ratio (SDR) are used to

ensure that the variance does not decrease too rapidly, which would lead to premature convergence. Second, Anticipated Mean Shift (AMS) preserves a sense of direction across iterations.

iAMaLGA_M [15] improves upon AMaLGA_M by combining decreasingly weighted estimations of previous generations with the current estimation. This lowers the required population size, as information from past iterations can be reused and is not discarded.

3.2.2. GOMEA

EDAs such as the AMaLGA_M [15] or CMA-ES [39] make use of multivariate Gaussians to model problem dependencies. However, while effective, these models provide an unnecessarily high model complexity for specific optimization problems, e.g., problems that do not require rotational invariance. Thus, performance in these cases can be increased using a model that narrowly fits the problem's complexity. GOMEAs provide the flexibility to do so.

GOMEAs use a linkage model described by a Family of Subsets (FOS), denoted \mathcal{F} , to model dependencies between decision variables explicitly. This FOS contains subsets of the set of all variable indices $\{0, 1, \dots, d-1\}$, and can be written as $\mathcal{F} = \{F^0, F^1, \dots, F^{|\mathcal{F}|-1}\}$, with $F^i \subseteq \{0, 1, \dots, d-1\}$. These subsets, named linkage sets, specify groups of decision variables that should be updated simultaneously during the variation phase of the EA. Using a linkage model allows for dependencies between decision variables to be modeled and the problem structure to be exploited when these are modeled correctly.

GOMEA can use a linkage tree (LT) FOS built hierarchically bottom-up. This can be done using the Unweighted Pair Grouping Method with Arithmetic-mean [34] (UPGMA). This method starts with subsets containing only one decision variable each and consecutively merges the two closest subsets until a set containing all decision variables are formed. This is done using some distance metric, e.g., Mutual information. This process can be done in $O(nl^2)$ time [34] and must be performed at each generation's start.

Variation occurs through a method called Optimal Mixing (OM). For each candidate solution x , this method iterates randomly over all $F^i \in \mathcal{F}$, then randomly selects a donor d from the population and applies the decision variables d_{Fi} to x_{Fi} . If this results in an improved solution, the change is accepted; otherwise, it is reverted. This operation is called optimal since the quality of solutions during variation monotonically improves.

Using an explicit linkage model described by a FOS and optimal mixing allows for partial evaluations, which can substantially improve performance if the problem is known to a certain degree [23].

3.2.3. RV-GOMEA

RV-GOMEA provides an adaptation of GOMEA using techniques from AMaLGA_M to target real-valued optimization problems. In short, this is done by estimating a $|F^i|$ -variate Gaussian for each $F^i \in \mathcal{F}$. Then, during OM, proposals are sampled from these distributions instead of obtaining them from donor solutions in the population. Again, AVS, SDR, and AMS are used as in AMaLGA_M, but now for each distribution individually. Lastly, when a linkage model is learned every generation, distribution parameters are passed along from generation to generation through a matching algorithm, which finds a one-to-one mapping of FOS elements that are most similar between generations. This procedure is not necessary for static linkage models.

Consequently, RV-GOMEA combines the exploitation of linkage structure from GOMEA with the exploration by sampling multivariate Gaussians from AMaLGA_M to cover the search space of a real-valued optimization problem efficiently.

3.3. Multi-objective Optimization

The previously described methods are all single-objective, but some problems are multi-objective. These problems specify multiple optimization objectives of interest, often posing conflicting trade-offs between possible solutions. These can be solved using single-objective methods by specifying weights in advance to optimize a linear combination of the objectives. However, it is often hard to express in advance what trade-off between the objectives would ultimately be desirable. A set of high-quality solutions, with different trade-offs between the objectives, could give the necessary context to decide on the preferred solution. Finding such a set is the goal of multi-objective optimization.

More formally, we can define a multi-objective optimization problem as

$$\min(f_1(x), f_2(x), \dots, f_k(x)) \text{ s.t. } x \in X, \quad (3.6)$$

where $k \geq 2$, X is the set of feasible solutions, and f_1, \dots, f_k some objective functions to be minimized, representing the conflicting objectives. The goal of multi-objective optimization is to find the Pareto set $P_S \subseteq X$, or an

approximation to it $A(S) \in X$. In the following sections, we provide theoretical background for these concepts.

3.3.1. Pareto-dominance

A solution x_1 is said to (Pareto) dominate another solution x_2 (denoted $x_1 \succ x_2$), if and only if

$$\begin{aligned} \forall (i \in \{1, 2, \dots, k\}) : f_i(x_1) \leq f_i(x_2) \wedge \\ \exists (i \in \{1, 2, \dots, k\}) : f_i(x_1) < f_i(x_2). \end{aligned}$$

Its objective values are better or equal for all objectives and strictly better for at least one objective. A solution x_1 is considered Pareto optimal if and only if $\nexists x_2 : x_2 \succ x_1$. Then we can define the set of all Pareto optimal solutions as $P_S = \{x_1 \in X \mid \nexists x_2 : x_2 \succ x_1\}$. Lastly, the Pareto front P_F of all objective function values corresponding to the solutions in P_S is defined as $P_F = \{f(x) = (f_1(x), f_2(x), \dots, f_k(x)) \mid x \in P_S\}$. We visualize these concepts in Figure 3.3.

A multi-objective optimization problem can be optimally solved if P_S can be found. However, in many cases, this is intractable due to the dimensionality and complexity of the problem and the fact that P_S may be infinitely large. The aim then is to find an approximation to P_S . However, the problem of finding such an approximation is multi-objective on its own since the solutions in this set should be close to the Pareto front but evenly spread out across it as well; that is, we aim for proximity and diversity.

We can solve a multi-objective problem single-objectively by using performance indicators. Many performance indicators have been proposed to capture multiple objectives into a single function. Domination-based methods also exist, but they suffer from stagnation when most of the population becomes non-dominated, preventing convergence to the Pareto set [56]. The Hypervolume indicator is the only known measure that is strictly monotonic with respect to Pareto-dominance [56]. Thus, an approximation set with optimal Hypervolume has to be a subset of the Pareto set. A limitation of Hypervolume is that there is no selection pressure from dominated solutions toward non-dominated solutions. To combat this, Uncrowded Hypervolume [56] was introduced.

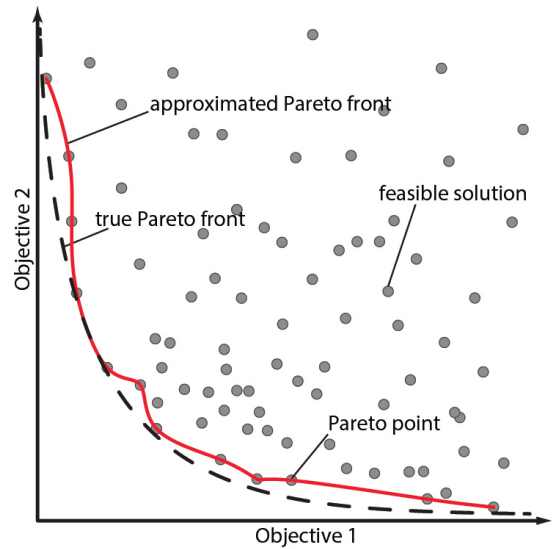


Figure 3.3: Pareto front of a minimization problem concerning two objectives.

3.3.2. (Uncrowded) Hypervolume

Domination-based optimization methods are applied by viewing the solutions separately; a solution can be improved locally by finding a solution that dominates it. Hypervolume-based methods view the problem from the perspective of the entire solution set.

The hypervolume $HV(S)$ for a solution set S measures the volume dominated in objective space by all $x \in S$, bounded by a reference point r . The hypervolume improvement $HVI(x, S)$ is defined as the increase in Hypervolume when x is added to S , i.e., $HVI(x, S) = HV(S \cup \{x\}) - HV(S)$. The approximation set $A(S)$ is the most significant subset of S that contains only non-dominated solutions, i.e., $A(S) = \{x \mid x \in S, \nexists y \in S : y \succ x\}$.

For any dominated solution $x \in S$, the hypervolume improvement $HVI(x, S) = 0$. This means there is no difference between a dominated solution close to the approximation boundary (the boundary between dominated and non-dominated region in objective space, also see Figure 3.4) and a dominated solution that lies far from it. This results in inefficient use of the population during optimization since an EA will not know if it has improved dominated solutions.

For this reason, the uncrowded distance $ud(x, S)$ is introduced [56]. It measures the shortest Euclidean distance between a dominated solution $x \in S$ and the approximation boundary of S . If x is non-dominated, $ud(x, S) = 0$. The uncrowded hypervolume improvement (UHVI) can then be defined as $UHVI(x, S) = HVI(x, S) - ud(x, S)$. With this, we can define the uncrowded Hypervolume (UHV), which measures the quality of the entire

solution set. It is defined as

$$UHV(S) = HV(S) - \frac{1}{|S|} \sum_{x \in S} ud(x, S)^m, \quad (3.7)$$

where m is the dimensionality of the objective space. The exponentiation with m ensures that the uncrowded distance is in the same unit scale as the Hypervolume. $\frac{1}{|S|}$ is used to ensure that an improvement in Hypervolume does not lead to a more negative influence of the increased uncrowded distances. UHV, in contrast to HV, is not strictly monotonic w.r.t Pareto dominance [56].

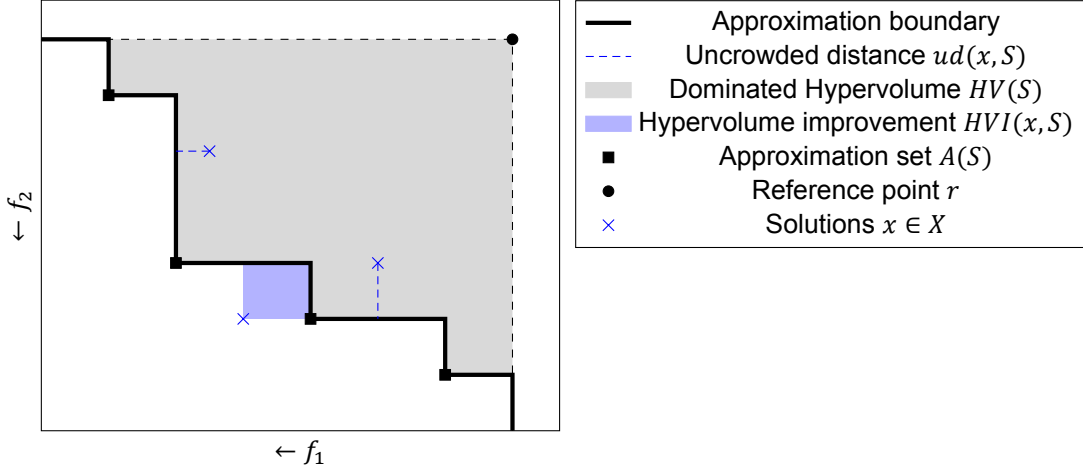


Figure 3.4: Visualization of the hypervolume (improvement), uncrowded distance, approximation set and boundary for an example solution set $S \in X$.

3.4. Partial Evaluations

In a Gray-Box Optimization (GBO) setting, some problem structures can allow partial evaluation of the objective function based on the changed problem variables since the last evaluation. These evaluations can be utilized in specific optimization algorithms for improved computational efficiency. More specifically, these can be utilized when the optimization procedure applies variation to (small) subsets of variables, as is the case for the GOMEA variation procedure as discussed in Section 3.2.2. A partial evaluation efficiently calculates the objective value after modifying a subset of problem variables, subtracting their contribution from the overall objective value before the modification and adding their contribution after it.

3.4.1. Definition

Formally, partial evaluations can be applied to an optimization problem when the objective function is known to be constructed from k subfunctions, for which it is known on which problem variables they depend. These indices are defined by $I = \{\mathcal{J}_0, \mathcal{J}_1, \dots, \mathcal{J}_{k-1}\}$, where \mathcal{J}_j defines the indices on which the subfunction f_j^M depends. Given k subfunctions f_0^M, \dots, f_{k-1}^M and I , partial evaluations can be formulated in the following manner [22]:

$$f(x) = f^P(f_0^M(x|\mathcal{J}_0) \oplus \dots \oplus f_{k-1}^M(x|\mathcal{J}_{k-1})) = f^P\left(\bigoplus_{j=0}^{k-1} f_j^M(x|\mathcal{J}_j)\right), \quad (3.8)$$

where for each $j \in [0, \dots, k-1]$, $f_j^M : \mathbb{R}^{|\mathcal{J}_j|} \rightarrow \mathbb{R}$ defines a function with as input a subset of the indices of x , which are defined in \mathcal{J}_j . The \oplus can be any commutative binary operator for which an inverse operator exists. f^P can either be the identity function or some non-invertible function, but in this thesis, we will only consider the former. A complete definition of partial evaluations is given in [22].

Most interestingly, for this thesis, we can define the new objective value for a particular solution x , given a modification to index i , resulting in x' as follows:

$$f^{part}(x, f_x, x', i) = f_x \ominus \bigoplus_{\mathcal{J}_j \ni i} f_j^M(x|\mathcal{J}_j) \oplus \bigoplus_{\mathcal{J}_j \ni i} f_j^M(x'|\mathcal{J}_j), \quad (3.9)$$

with f_x the previous objective value for x , and $\mathcal{J}_j \ni i$ shorthand for $\{\mathcal{J}_j \in I \mid i \in \mathcal{J}_j\}$, i.e. all sets of indices \mathcal{J}_j in which index i occurs. With this equation, we can partially evaluate certain similarity metrics and penalty terms in DIR when using common B-spline-based transformation models. This is explained in detail in Section 4.1.

3.4.2. Variable Interaction

In order to model the dependencies between problem variables, a Variable Interaction Graph (VIG) can be constructed. A VIG $G = (V, E)$ contains a vertex $u \in V$ for each problem variable $X_u \in X$, and an edge $(u, v) \in E$ for each pair of problem variables X_u, X_v that are specified to be dependent. A dependency between problem variables X_i, X_j exists if they jointly occur in the required input of at least one subfunction.

For a B-spline transformation model, we can describe the dependencies more naturally by aggregating the problem variables for each control point. For example, the VIG for a 6x6 grid is shown in Figure 3.5, indicating many overlapping dependencies between problem variables for these transformation models.

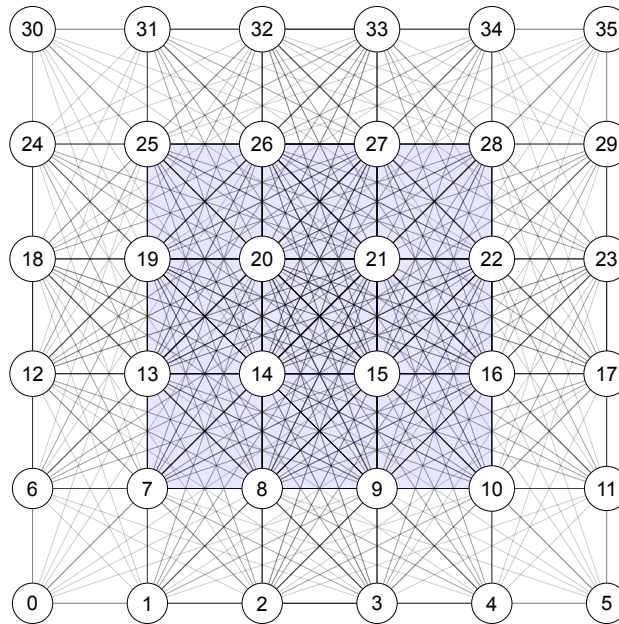


Figure 3.5: Variable Interaction Graph (VIG) with each node a control point in a 6x6 B-spline grid. The underlying (fixed) image is visualized by the region shaded in blue. The outermost control points are added so that the transformation of each region of voxels is supported by the required number of control points (4^d with d the dimensionality of the voxels).

A traditional linkage tree (LT), as used in GOMEA, is inherently less suited for these overlapping dependencies due to its hierarchical construction in a bottom-up fashion. This limits the overlapping dependencies that can be modeled since two sets must always be merged completely. To address this issue, different linkage models have been proposed. In [18], Bosman et al. propose the linkage neighbors (LN) model, which aims to model for each problem variable its closest neighbors in terms of linkage. These neighbors can be learned from the population data using various proposed methods. A method must form sets for each problem variable X_i of limited size with only its closest neighbors that meet some linkage criterion or threshold. Another method is proposed by Bouter et al. [24], leveraging the conditional linkage between problem variables. It does so by sampling new solutions for a subset of variables by conditioning on the variables not contained in this set. This can be done efficiently by viewing the VIG as a Gaussian Markov Field (GMF) between problem variables.

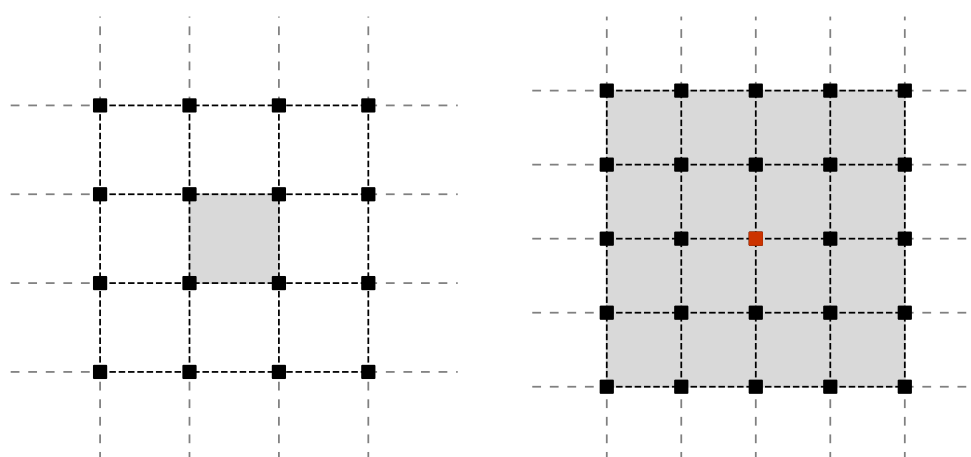
Although Figure 3.5 displays many overlapping dependencies, it does not say anything about the magnitude of these dependencies in terms of their linkage in practice. It is essential to measure these linkages first for various registration problems when using common B-spline transformation models; applying specific overlapping linkage models might not provide their theoretic benefits in practice if most dependencies can be filtered based on empirical evidence.

This work is the first to apply partial evaluations to common B-spline transformation models in deformable image registration (DIR). In this chapter, we provide our design for this application. Section 4.1 first provides the framework required for partial evaluations when using common B-spline transformation models in general and subsequently provides the design for their implementation when used during registration with specific similarity metrics or penalty terms. Then, in Section 4.2, we provide definitions for the linkage models used in this thesis.

4.1. Partial Evaluations in Common B-spline Transformation Models

As stated in Section 3.1.2, B-spline basis functions have local support, which means that a limited set of surrounding control points influence any point on a B-spline curve. In common B-spline transformation models, the surrounding control points only influence the transformation of a patch or block of voxels.

When using uniform cubic B-splines, 16 and 64 control points define the transformation of either a patch or block of voxels in 2D or 3D, respectively (see Figure 4.1). Consequently, this allows for the partial evaluation of similarity metrics or penalty terms if these can be defined as a commutative combination of calculations based on these sets of voxels (see Section 3.4). If the coefficients of a control point of the B-spline grid are altered, we will only have to recalculate the similarity metric for the voxels influenced by this point (see Figure 4.1).



(a) The transformation of the highlighted 2D patch of voxels is influenced only by the surrounding 16 control points. (b) If the coefficients of the highlighted control point (red) are altered, the highlighted patches of voxels will need to be re-evaluated.

Figure 4.1: A transformation grid using uniform cubic B-splines is shown, which can transform the moving image by moving the control points. This visualization demonstrates the local support of B-spline basis functions when they are used in 2D transformation models.

For example, the sum of absolute differences (SAD) is the sum of absolute voxel-to-voxel differences between the moving image I_M and fixed image I_F . Formally, it can be defined as

$$SAD(I_M, I_F) = \sum_{x \in I_F} |I_F(x) - I_M(T(x, \mu))| \quad (4.1)$$

$T(x, \mu)$ transforms a voxel x using a parameterized transformation model defined by μ , and I is the evaluated voxel intensity at a voxel from the moving or fixed image. Instead, we can also take this summation over the set $P = \{P_1, \dots, P_n\}$ of n mutually exclusive subsets of voxels as defined by the grid of a B-spline transformation model. Then, the SAD can be calculated by taking $SAD(I_M, I_F) = \sum_{P_i \in P} \sum_{x \in P_i} |I_F(x) - I_M(T(x, \mu))|$, a double summation over the absolute differences between patches of voxels. Suppose a change is made to a control point c such that it becomes c' . In that case, we only need to take the old SAD for the whole image, and subtract from it the SAD for the patches of voxels influenced by c , after which we add the SAD for the same patches but now with the updated transformation parameters from c' incorporated in μ . This significantly improves the computational performance of the registration when using sufficiently large grid sizes since it removes the need for a full re-evaluation each time a small subset of control points is changed. Full evaluations will always need to be done for newly generated solutions to obtain their initial objective values.

4.1.1. Definitions

To allow for the formal and concise formulation of partial evaluations when applied to the individual metrics, we have defined the following:

- I_M and I_F , the moving and fixed image respectively, with $I_{F|M}(x)$ the evaluated voxel intensity at point x on either image. Here a point can lie between voxels, in which case some form of interpolation is used. Each image is defined on its spatial domain: $\Omega_F \subset \mathbb{R}^d$ and $\Omega_M \subset \mathbb{R}^d$ with d the image dimension. As stated before, the set $P = \{P_1, \dots, P_n\}$ defines the total n mutually exclusive regions of voxels covering the entirety of the fixed image as defined by the B-spline grid.
- μ parameterizes the B-spline transformation model, defined by the set of control points C with size $|C| = C_1 \times \dots \times C_d$ with C_1, \dots, C_d the number of control points along each axis. With this, the B-spline parameters in the 2D case are composed as follows: $\mu = \{c_{1x}, c_{2x}, \dots, c_{|C|x}, c_{1y}, c_{2y}, \dots, c_{|C|y}\}$ with c_i individual control points. $T : \mathbb{R}^d \times \mathbb{R}^{|C|d} \rightarrow \mathbb{R}^d$ defines a mapping from any point on the fixed image to a point either on or outside the moving image.
- To simplify further formulations, we define the mapping $P : \mathcal{P}(C) \rightarrow \mathcal{P}(P)$, which maps any subset of control points $\{c_1, \dots, c_x\}$ to the corresponding subset of influenced regions of voxels $\{P_1, \dots, P_x\}$. And the mapping $C : \mathcal{P}(P) \rightarrow \mathcal{P}(C)$, which maps any subset of voxel regions to the control points that influence it.

4.1.2. Mean Squared Difference

Similar to the SAD, the mean squared difference (MSD) is based on the voxel-to-voxel difference between two images. It is defined as follows:

$$MSD(\mu; I_F; I_M) = \frac{1}{|\Omega_F|} \sum_{x_i \in \Omega_F} (I_F(x_i) - I_M(T_\mu(x_i)))^2 \quad (4.2)$$

Now, to allow for partial evaluations, the division by the total number of voxels in the fixed image domain needs to be taken into the summation so that partial evaluations for subsets of voxels can be added and subtracted from each other; i.e., all summations are in the same scale:

$$MSD(\mu; I_F; I_M) = \sum_{x_i \in \Omega_F} \frac{(I_F(x_i) - I_M(T_\mu(x_i)))^2}{|\Omega_F|}. \quad (4.3)$$

With this, we can formulate how to partially update a given solution μ with current MSD m of which the subset of control points $C' \in \mathcal{P}(C)$ were altered such that it resulted in μ' :

$$MSD_{\text{part}}(\mu; \mu', C'; m; I_F; I_M) = m - \left(\sum_{P_i \in \mathcal{P}(C')} \sum_{x_i \in P_i} \frac{(I_F(x_i) - I_M(T_\mu(x_i)))^2}{|\Omega_F|} \right) + \left(\sum_{P_i \in \mathcal{P}(C')} \sum_{x_i \in P_i} \frac{(I_F(x_i) - I_M(T_{\mu'}(x_i)))^2}{|\Omega_F|} \right). \quad (4.4)$$

4.1.3. Thin-plate Bending Energy

The thin-plate bending energy measures any given deformation field's deformation magnitude (or smoothness). When uniform cubic B-splines parameterize this deformation field, we can calculate the bending energy directly from the coefficients of the control points using an analytical derivation [75], which expresses it as a linear combination of bending energies for each region of voxels P_i . The smoothness of a single region is calculated as a summation of squared second-order partial derivatives using the coefficients of the control points that support the region. By taking \vec{c}_x , \vec{c}_y and \vec{c}_z to be vectors containing the ordered coefficients of all control points for each dimension, respectively, we can define the smoothness of S of a region P_i as:

$$S_{P_i}(C) = \sum_{(c_x, c_y, c_z) \in C(P_i)} (\vec{c}_x V^{(c_x^i, c_y^i, c_z^i)} \vec{c}_x^T + \vec{c}_y V^{(c_x^i, c_y^i, c_z^i)} \vec{c}_y^T + \vec{c}_z V^{(c_x^i, c_y^i, c_z^i)} \vec{c}_z^T), \quad (4.5)$$

Where c_x^i , c_y^i and c_z^i correspond to the grid indices of a control point, and $V^{(x,y,z)}$ a specialized matrix operator derived in [75]. The definition was constructed in this manner—using vectors and matrix operators—to optimize for computational performance. Now the smoothness of the entire fixed image becomes

$$S(C; P) = \sum_{P_i \in \mathcal{P}} S_{P_i}(C). \quad (4.6)$$

To partially update a solution with control points C (parameterized in μ) with current smoothness s , given that a subset of control points $C' \in \mathcal{P}(C)$ were altered, we have to calculate

$$S_{\text{part}}(C; P; C'; s) = s - \left(\sum_{P_i \in \mathcal{P}(C')} S_{P_i}(C) \right) + \left(\sum_{P_i \in \mathcal{P}(C')} S_{P_i}(C|C') \right), \quad (4.7)$$

with $C|C'$ corresponding to the set of all control points with the updated coefficients from C' .

4.1.4. Zero-Normalized Correlation Coefficient

The zero-normalized correlation coefficient (ZNCC) is calculated as a summation of the voxel-to-voxel correlations, normalized by the product of both the image's standard deviations in terms of voxel intensities. It is defined as

$$ZNCC(\mu; I_F; I_M) = \frac{\sum_{x_i \in \Omega_F} (I_F(x_i) - \overline{I_F}) \cdot (I_M(T_\mu(x_i)) - \overline{I_M})}{\sqrt{\sum_{x_i \in \Omega_F} (I_F(x_i) - \overline{I_F})^2 \cdot \sum_{x_i \in \Omega_F} (I_M(T_\mu(x_i)) - \overline{I_M})^2}}, \quad (4.8)$$

with $\overline{I_F} = \frac{1}{|\Omega_F|} \sum_{x_i \in \Omega_F} I_F(x_i)$ and $\overline{I_M} = \frac{1}{|\Omega_F|} \sum_{x_i \in \Omega_F} I_M(T_\mu(x_i))$ the average voxel intensity values. This formulation

does not allow for partial evaluations due to the $\overline{I_M}$ term appearing in every addend for two summations. Thus, if only a subset of voxels is changed, these summations must be reevaluated for all voxels from the fixed image. To remedy this, we have to rewrite the equation slightly so that it no longer uses $\overline{I_M}$ in these individual addends. First, we rewrite the numerator:

$$= \sum_{x_i \in \Omega_F} (I_F(x_i) - \bar{I}_F) \cdot (I_M(T_\mu(x_i)) - \bar{I}_M) \quad (4.9)$$

$$= \sum_{x_i \in \Omega_F} I_F(x_i) \cdot I_M(T_\mu(x_i)) - \bar{I}_M \cdot \sum_{x_i \in \Omega_F} I_F(x_i) - \bar{I}_F \cdot \sum_{x_i \in \Omega_F} I_M(T_\mu(x_i)) + \sum_{x_i \in \Omega_F} \bar{I}_F \cdot \bar{I}_M \quad (4.10)$$

$$= \sum_{x_i \in \Omega_F} I_F(x_i) \cdot I_M(T_\mu(x_i)) - 2 \cdot \frac{\sum_{x_i \in \Omega_F} I_F(x_i) \cdot \sum_{x_i \in \Omega_F} I_M(T_\mu(x_i))}{|\Omega_F|} + \sum_{x_i \in \Omega_F} \frac{\sum_{y_i \in \Omega_F} I_F(y_i) \cdot \sum_{y_i \in \Omega_F} I_M(T_\mu(y_i))}{|\Omega_F|^2} \quad (4.11)$$

$$= \sum_{x_i \in \Omega_F} I_F(x_i) \cdot I_M(T_\mu(x_i)) - \frac{\sum_{x_i \in \Omega_F} I_F(x_i) \cdot \sum_{x_i \in \Omega_F} I_M(T_\mu(x_i))}{|\Omega_F|} = S_{FM} - S_F \cdot S_M \cdot \frac{1}{|\Omega_F|}, \quad (4.12)$$

with S_{FM} , S_F , and S_M being shorthand notation for the individual summations. With this, we have reduced the numerator such that it no longer depends on \bar{I}_M within any of the summations addends. It can be calculated using just three summations that do not depend on any globally changing term so that it can be partially evaluated. We can rewrite the summations from the denominator in a similar fashion:

$$= \sum_{x_i \in \Omega_F} (I_M(T_\mu(x_i)) - \bar{I}_M)^2 \quad (4.13)$$

$$= \sum_{x_i \in \Omega_F} I_M(T_\mu(x_i))^2 - 2\bar{I}_M \cdot \sum_{x_i \in \Omega_F} I_M(T_\mu(x_i)) + \sum_{x_i \in \Omega_F} \bar{I}_M^2 \quad (4.14)$$

$$= \sum_{x_i \in \Omega_F} I_M(T_\mu(x_i))^2 - 2 \cdot \frac{\sum_{x_i \in \Omega_F} I_M(T_\mu(x_i)) \cdot \sum_{x_i \in \Omega_F} I_M(T_\mu(x_i))}{|\Omega_F|} + \sum_{x_i \in \Omega_F} \frac{\sum_{y_i \in \Omega_F} I_M(T_\mu(y_i)) \cdot \sum_{y_i \in \Omega_F} I_M(T_\mu(y_i))}{|\Omega_F|^2} \quad (4.15)$$

$$= \sum_{x_i \in \Omega_F} I_M(T_\mu(x_i))^2 - \frac{\sum_{x_i \in \Omega_F} I_M(T_\mu(x_i)) \cdot \sum_{x_i \in \Omega_F} I_M(T_\mu(x_i))}{|\Omega_F|} = S_{MM} - S_M \cdot S_M \cdot \frac{1}{|\Omega_F|}, \quad (4.16)$$

again, with S_{MM} as shorthand notation. Now, we can formulate the ZNCC as follows:

$$ZNCC(\mu; I_F; I_M) = \frac{S_{FM} - \frac{S_F \cdot S_M}{|\Omega_F|}}{\sqrt{(S_{FF} - \frac{S_F^2}{|\Omega_F|}) \cdot (S_{MM} - \frac{S_M^2}{|\Omega_F|})}} \quad (4.17)$$

Which only consists of summations without any terms that necessitate a reevaluation of the entire sum whenever they are changed. By keeping track of these five summations for every solution, we can similarly evaluate them as with the MSD by removing the old addends from the summations and adding the new ones. For convenience, we define

$$S_M(\mu; C'; \mu') = \sum_{x_i \in \Omega_F} I_M(T_\mu(x_i)) - \sum_{P_i \in P(C')} \sum_{x_i \in P_i} I_M(T_\mu(x_i)) + \sum_{P_i \in P(C')} \sum_{x_i \in P_i} I_M(T_{\mu'}(x_i)) \quad (4.18)$$

As the partial evaluation of the sum of fixed image voxel intensities S_M given changed control points in C' resulting in parameters μ' , as well as similar definitions for the summations S_F , S_{FM} , S_{FF} , and S_{MM} (with I_F , I_M implicit). With this, the full, partial evaluation of the ZNCC becomes

$$ZNCC_{\text{part}}(\mu; C'; \mu'; I_F; I_M) = \frac{S_{FM}(\mu; C'; \mu') - \frac{S_F(\mu; C'; \mu') \cdot S_M(\mu; C'; \mu')}{|\Omega_F|}}{\sqrt{\left(S_{FF}(\mu; C'; \mu') - \frac{S_F(\mu; C'; \mu')^2}{|\Omega_F|} \right) \cdot \left(S_{MM}(\mu; C'; \mu') - \frac{S_M(\mu; C'; \mu')^2}{|\Omega_F|} \right)}} \quad (4.19)$$

4.2. Linkage Models

GOMEA allows any linkage model to be set and used for optimization. In this thesis, we focus on four specific models when considering registration problems in 3D. Let $I_{(i)} = \{\mu_{(i-1)d}, \mu_{(i-1)d+1}, \mu_{(i-1)d+2}\}$ for $i = 1, \dots, |C|$ be the decision variables corresponding to the B-spline coefficients of control point c_i . We can then define the following linkage models:

$$\mathcal{F}_{Uni} = \{\mu_1, \mu_2, \dots, \mu_{3|C|}\}, \quad (4.20)$$

$$\mathcal{F}_{MP} = \{I_{(1)}, I_{(2)}, \dots, I_{(|C|)}\}, \quad (4.21)$$

$$\mathcal{F}_{SLT} = UPGMA(I_{(1)}, I_{(2)}, \dots, I_{(|C|)}), \quad (4.22)$$

$$\mathcal{F}_{Full} = \{I_{(1)} \cup I_{(2)} \cup \dots \cup I_{(|C|)}\}. \quad (4.23)$$

The univariate linkage model \mathcal{F}_{Uni} consists of one set for each of the decision variables independently and does not consider there to be any dependencies between them. In the marginal linkage model \mathcal{F}_{MP} , only the decision variables corresponding to the same coordinates of one control point are considered dependent. Then, in the bounded static linkage tree model \mathcal{F}_{SLT} , the sets of individual control points are merged bottom-up using the unweighted pair group method with arithmetic mean (UPGMA) algorithm [34]. The distance matrix this routine uses is calculated using the Euclidean distances between each pair of control points. With this distance matrix, the two closest sets are merged iteratively until a single set contains all control points. This tree is computed once at the start of optimization and can be bounded by filtering sets with sizes falling outside some range. Lastly, in the full linkage model \mathcal{F}_{Full} , all the variables are considered to be dependent as defined by one subset with all decision variables in it. A visualization of each linkage model for a 4x4 2D grid is given in Figure 4.2.

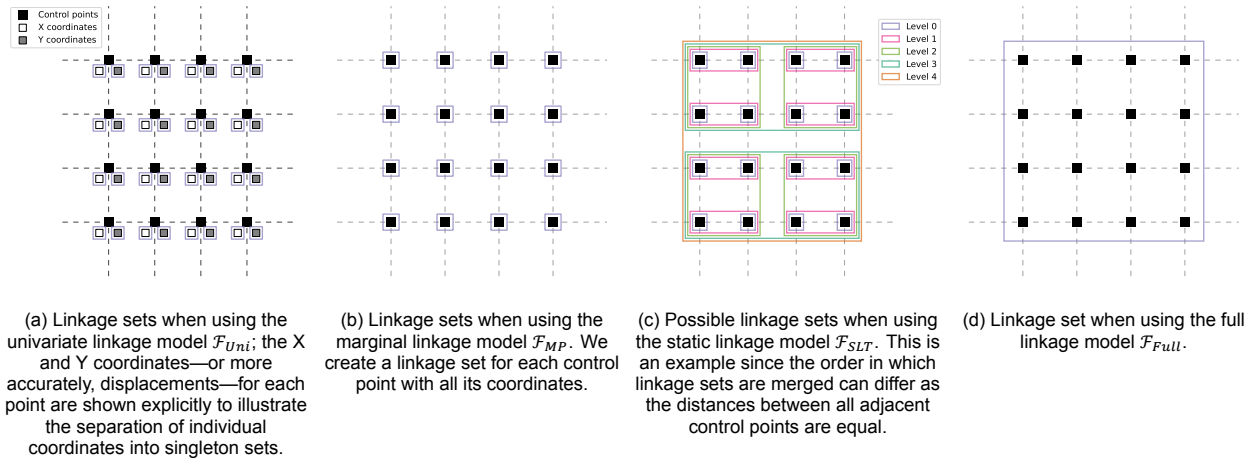


Figure 4.2: Visualization of linkage models when using a 4x4 2D B-spline grid. The linkage sets are visualized for each linkage model using purple-colored rectangular regions.

5

Implementation

Open-source image registration frameworks which facilitate the rapid prototyping of new registration algorithms and ideas are readily available, e.g., the Insight Toolkit (ITK) [58], Elastix [49], and Plastimatch [77]. In this thesis, we have implemented RV-GOMEA within the Elastix framework due to its modular design and readily available set of transformation models, similarity metrics, and support for multi-resolution registration. It is written in C++ and uses ITK for the underlying image representations and transformations.

This chapter gives an architectural overview of Elastix and our implementation of RV-GOMEA within it. First, we provide a more detailed description of ITK in Section 5.1 since Elastix heavily relies on it. Then, in Section 5.2, we show how RV-GOMEA is ported from its source code in C [20] to an ITK-compatible component usable in registration frameworks. Besides this, further additions and improvements, in the form of, e.g., optimizations or additional inspection output, are also described. Section 5.3 shows how RV-GOMEA is integrated into Elastix. Most importantly, we show how the support for partial evaluations was implemented and how this has affected several components of the registration framework.

5.1. ITK

The Insight Toolkit (ITK) is a library of well-tested software components that can be used to develop image segmentation and registration programs. Its code base is in C++ with the generic programming paradigm in mind. This means most classes and functions are templated and can be instantiated at compilation using either native types (e.g., a float or int) or user-defined types (e.g., a class). Its components are used extensively in Elastix, so its most relevant concepts are described briefly. A more detailed overview is given in [47].

5.1.1. Image representation

Images from the medical domain are taken using devices of varying modality, but ultimately they all aim to capture and visualize something from the real-world physical domain. As such, it makes sense to describe images using their inherent "object" domain as described by their resolution in terms of voxels and by mapping from this object domain to the physical domain. This is done by specifying each image's spacing, origin, and direction according to some real-world coordinate system. The spacing specifies the distance per dimension between each voxel in *mm*. The origin specifies the displacement of the origin of the image, also in *mm*. Finally, the direction allows for the mirroring of the image w.r.t. physical space.

With this representation, medical images can be registered, manipulated, and compared in the physical domain, providing a shared frame of reference. Then, the actual interpolation and voxel-specific operations can be done by mapping the actions back to the object domain of a specific image. Any image lacking this spatial information cannot reliably be used in a clinical context since the results will rely on the mapping of the image in the object domain, which can be altered at will by, e.g., down-sampling the image to a lower resolution (a common occurrence in registration frameworks).

5.1.2. Data representation

In ITK, two main types of objects exist: data and process objects. Data objects represent data (e.g., images, meshes, masks), while process objects are used to produce or process data. Process objects, also commonly called filters, can take data objects as input and can produce data objects as output. If either the internal state

of a filter or its input changes, it is updated automatically. This way, data, and process objects can be chained together to form a data processing pipeline with automatic updates. Filters are used extensively in Elastix, e.g., for downscaling images or upsampling the B-spline parameters.

5.2. RV-GOMEA

The source code of RV-GOMEA [20] provides an implementation in C usable as a command-line executable. For integration into the Elastix framework, it needs to be implemented as a class according to ITK standards. To this end, the code was first stripped from code relevant only for the publication at the time [20]. This includes bounds checking, built-in objective functions, and specific output for debugging purposes written to files. Next, the code was refactored as a class within the optimizers framework of ITK. Besides this, some minor additions and alterations have been made to either increase performance or ease of use.

5.2.1. Shrinkage

Cholesky factorizations are used to efficiently sample new solutions from the estimated normal distributions, [44]. However, these require a positive definite covariance matrix estimated from the generated solutions within a generation of RV-GOMEA. If the covariance matrix's dimensionality is too high or the population size used for estimating this matrix is too low, the Cholesky factorization is likely to fail. In this case, the algorithm falls back on a univariate factorization using only the diagonal of the covariance matrix, which is undesirable.

Solving DIR using B-spline-based transformation models poses a high-dimensional optimization problem. If it is solved using RV-GOMEA using linkage sets containing many problem variables, large population sizes are necessary to estimate the covariance matrices accurately. Estimating covariance matrices is done using the unbiased sample estimate, which does not minimize the mean-squared error (MSE) [28]. In some cases, using other estimators that introduce some bias in exchange for a lower required population size might make more sense. Unlike the sample estimate, the oracle approximating shrinkage (OAS) estimator [28] minimizes the MSE when samples are normally distributed. It is based on previous shrinkage estimators well suited to high-dimensional problems with small sample sizes.

Within RV-GOMEA, the estimated covariance matrices are subsequently used to sample new solutions. Then, these samples are used in the next generation to estimate new covariance matrices that are shrunk again. Shrinkage is thus applied to the covariance matrices at each generation that was already shrunk in the previous generation. This leads to covariance matrices with increasingly large values for the individual variances of the variables (along the diagonal) and decreasingly small values for the covariances between variables. As such, using such a shrinkage algorithm at each generation will make RV-GOMEA function increasingly more like it is using a univariate linkage model. Besides this, it will also lead to enormous variances over time, leading to proposed transformations with large displacements that map to positions outside of the moving image completely. Naturally, this behavior is not desired, so we do not recommend the application of shrinkage estimators within RV-GOMEA for B-spline-based DIR.

5.2.2. Inspection Output

Some extra features are implemented, which provide additional insights into the optimization process by tracking several metrics. These do not inherently change the functionality of GOMEA but merely allow us to understand how it produced a certain outcome more efficiently. These implemented inspection features include:

- **Distribution Multipliers:** At each generation, the distribution multipliers for each linkage set are written to a file, and an average of these values is computed and included in the iteration output of Elastix so that it can be tracked during optimization.
- **Control points:** The B-spline parameters form a grid of control points that starts in a uniformly aligned setting but, during the optimization process, shifts towards some set that minimizes the objective function. To better understand the trajectory of the control points over time, their average across the population is taken and written to a file for each generation.
- **Cholesky factorization:** To sample new solutions at each generation, a Cholesky factorization of the estimated covariance matrices is used. If this factorization fails due to the covariance matrices not being positive definite, the algorithm reverts to a more trivial factorization using only the diagonal of the covariance matrices. Thus, tracking the success rate of these factorizations for each generation makes sense. For each generation, we aggregate the matrix orders reached by the underlying factorization procedure

as a percentage of the covariance matrix order. The average of these percentages is included in the iteration output of Elastix.

- **Mutual Information:** The linkage between the problem variables can be measured at the end of each generation by calculating their Mutual Information. If enabled, a full Mutual Information matrix is written to a file at the end of each generation.

5.2.3. Optimizations

The original C code makes use of manual memory allocations and stored pointers for the used data structures required in GOMEA. In C++, more modern constructs exist, which increase the overall readability of the code while reducing the potential of introducing bugs. As such, all of the arrays were replaced with vectors from the C++ Standard Template Library, and the matrices were replaced with similar constructs from Eigen [35], which provides optimized matrix operations. A procedure from LAPACK [4] specific to real symmetric positive definite matrices is used for the Cholesky factorization. Similarly, a LAPACK routine specific for lower triangular matrices calculates their inverse.

5.3. Elastix

Elastix aims to improve ITK by providing a modular and user-friendly framework for image registration. Several new optimizers are introduced, and existing ones are enhanced with improvements such as better error handling and integrated support for calculating second-order spatial derivatives (used in regularization terms). In addition, the generic implementation of transformation models allows for the effective use of the compact support of many transformations [48, 49].

As stated earlier in this thesis, Elastix can perform registration in a multi-resolution or hierarchical setting. This allows the user to specify any subsequent stages in which the registration has to occur. These stages can be defined in any way the user seems relevant. However, the fixed and moving images are usually taken from coarse to fine through some pyramid down-sampling scheme to counteract the non-convexity of most registration problems. An overview of possible pyramid schemes available within Elastix is given in Figure 5.2. The registration process (Figure 5.1) can be defined with the following key components:

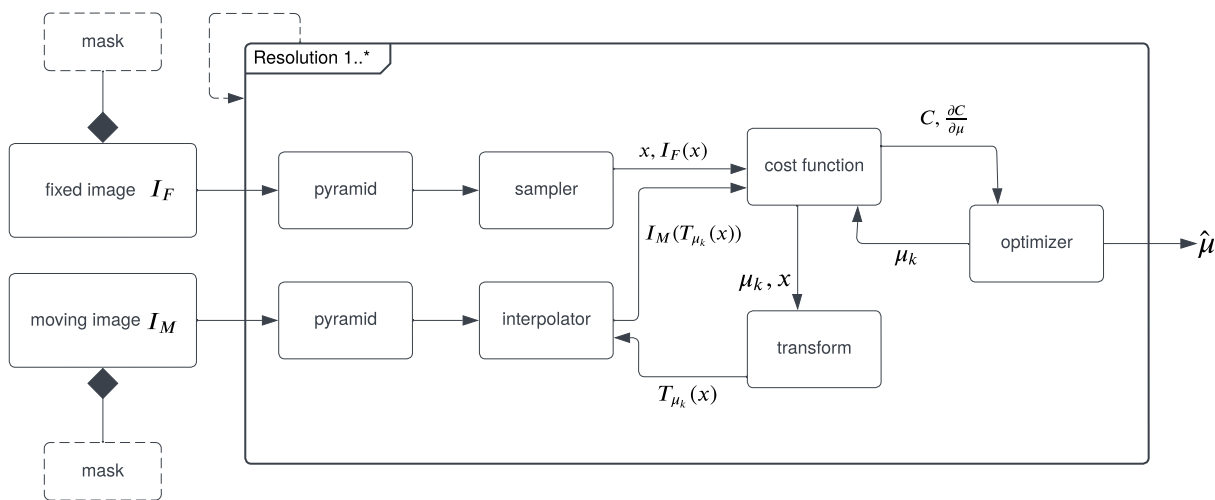


Figure 5.1: Overview of the registration process in Elastix. The dashed lines indicate that either a component or action is optional.

- **Image and mask:** The fixed image I_F and moving image I_M are taken as input for the registration process. Possibly, these include image masks, which indicate which voxels of the images to include in the cost function evaluation.
- **Gaussian pyramid:** To increase the coarseness of either image, a Gaussian pyramid scheme can be used. These can apply a Gaussian blur with or without downsampling. By downsampling, the number of voxels is reduced so that less time is required for costly function evaluations.

- **Image Sampler:** To evaluate the cost function, several sampling strategies can be used: the use of all voxels from the image, a subset of voxels selected on a uniform grid, random sampling of voxels, and random sampling of coordinates in the physical domain with values between voxels being interpolated.
- **Interpolator:** The transformation maps voxels from the fixed image to positions in coordinate space on the moving image. Often, these positions lie between voxels, and thus some form of interpolation is required to evaluate the image intensity $I_M(T_{\mu_k}(x))$ at these positions. Several interpolation methods are available, with varying trade-offs in speed and quality, including nearest neighbor, linear, and n th order B-spline interpolation.
- **Cost function:** The cost function C evaluates a similarity metric or a weighted combination of metrics and penalty terms on the fixed and warped moving image. It can return both the value and derivative w.r.t. μ .
- **Transform:** The transformation model is used to transform the moving image. It is parameterized by μ and provides $T_{\mu_k}(x)$ at iteration k , which the interpolator uses to evaluate the voxel intensity at that location.
- **Optimizer:** The optimizer iterates on the transformation parameters μ with the goal of minimizing the cost function C . After all resolutions are finished, it returns $\hat{\mu}$: the estimated optimal transformation parameters for the given registration problem.

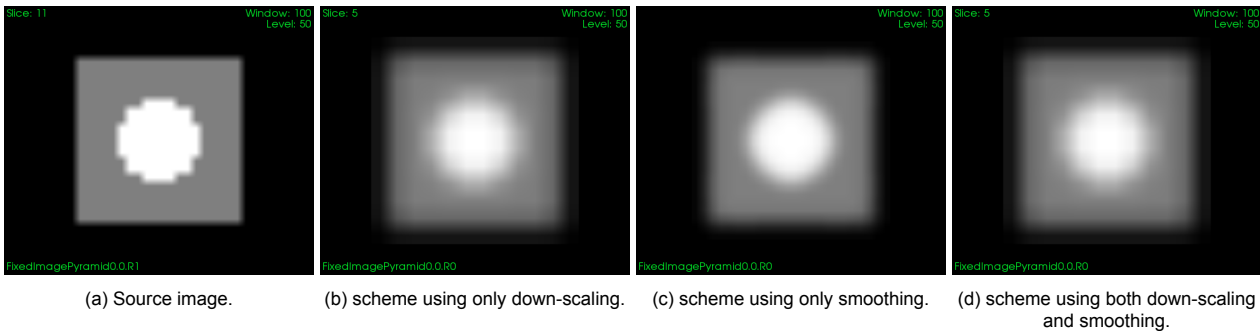


Figure 5.2: application of different pyramid schemes to a source image using Elastix. Down-scaling or smoothing using a Gaussian blur can be applied at various scales at any resolution level. We use the default isotropic schedule for both the re-scaling and smoothing factors.

These components consist of an ITK-style class and an Elastix-style wrapper class which connects it with the general framework (Figure 5.3). To integrate RV-GOMEA as an optimizer within the registration process, we thus have to use ITK-style implementation of RV-GOMEA as discussed in Section 5.2 and extend with an Elastix-style wrapper to make it usable as an optimizer for registration within the framework. The wrapper class of RV-GOMEA is responsible for reading and applying the parameters of RV-GOMEA as specified in a parameter file for each resolution, registering and writing to the iteration output, as well as error handling and logging termination conditions.

5.3.1. Analytic Regularization

To regularize the transformations, Elastix provides several regularization terms which can be used conjointly with a similarity metric as part of a weighted cost function. One of these, the thin-plate bending energy penalty, penalizes sharp deviations in the transformation and is calculated as a summation of partial second-order spatial derivatives w.r.t. the transformation parameters. In Elastix, this term is implemented using numerical differencing and approximates the bending energy by iterating over the fixed image voxels. This technique is generally applicable independent of the used transformation model, but it suffers from reduced computational efficiency. When B-spline-based transformation models are used, this term can be calculated analytically using only the B-spline coefficients as input [75]. This method has been implemented in the 3D volumetric registration tool Plastimatch [77]. Using this implementation, we re-implemented this method as a penalty term within Elastix, again using the class hierarchy shown in Figure 5.3. This required some adaptations of the original implementation due to a difference in the parameterization of the B-spline transformation model.

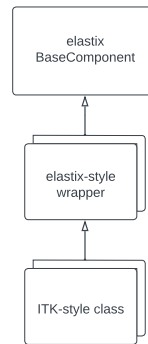


Figure 5.3: The ITK-style classes contain the actual functionality of the components, whereas the Elastix-style classes act as wrappers that "glue" the individual ITK-style components together in the overall framework.

5.3.2. Sobol Sequence

Some of the image samplers within Elastix require using a random number generator (RNG) to sample voxels or coordinates of the fixed image, which defaults to the Mersenne twister [57], a pseudo-random RNG. This generator outputs numbers used to select samples in a seemingly random fashion. If a more uniform but still pseudo-random distribution of samples is desired, other RNGs, such as those using a Sobol sequence [79], can be used. This could improve the predictability of the cost function from iteration to iteration when a sub-sampling strategy is used. As such, we have implemented a Sobol sequence generator within Elastix as part of the relevant image samplers.

5.3.3. Partial Evaluations

Understandably, Elastix does not naturally allow for the application of partial evaluations, as only some of the built-in optimizers can selectively update only a subset of problem variables. To allow for their application, we implement some functionality to the core components of both ITK and Elastix.

At the start of every resolution, the regions of pixels defined by the grid of control points in the B-spline transformation model must be computed. See Section 3.1 for a detailed overview, including visualizations. For each of these regions, an image sampler is constructed of the type defined in the parameter file by the user. In addition, image masks are considered: if the region of pixels falls entirely outside the bounding box of the provided mask, it is no longer considered during registration. Lastly, a mapping from linkage sets to regions of pixels they influence is computed.

As shown in Chapter 4, an objective function can be evaluated partially if it is known to be constructed from k subfunctions. In short, a partial evaluation is performed by taking the previous objective value, subtracting the relevant subfunctions using the old parameter values, and adding the same subfunctions using the updated parameter values. For example, the Sum of Absolute Differences (SAD) (see Equation 4.1) can be evaluated partially by taking the previous sum, subtracting the sum for each changed patch of voxels calculated using the previous parameters, and adding the sum for each changed patch of voxels calculated using the updated parameters. However, in some cases, the evaluation is more involved, requiring a subtraction and addition for multiple operands of each subfunction. As such, we construct a helper class *IntermediateResults* compatible with the binary addition and subtraction operators, which can contain any number of operands. For example, an instantiation of this class could contain the individual summations of the Zero-Normalized Correlation Coefficient (ZNCC) (Section 4.1.3).

Two functions need to be defined and overridden to implement the functionality to evaluate any given metric in Elastix partially. First, *GetValuePartial*, which calculates for a subset of regions of pixels the partial evaluation which is returned as an *IntermediateResults* object (See Figure 5.4). Second, *GetValue*, which takes as input an updated *IntermediateResults* object, and from it calculates the new metric value. With these methods implemented successfully for a metric, GOMEA can partially evaluate solutions during registration.

5.3.4. Missed Voxels

In order to calculate the value of a similarity metric, we interpolate the mapped voxel values of the moving image. However, these mappings may fall outside the moving image when using a deformation model based on B-

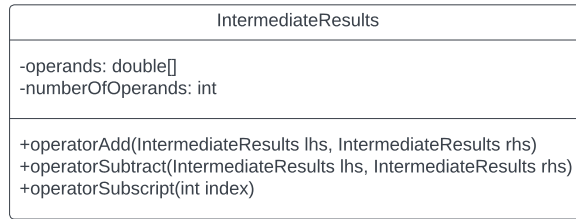


Figure 5.4: UML diagram of IntermediateResults class. It provides public operator methods that allow for the addition and subtraction of individual instances. Specific operands of an instance can be accessed using the subscript operator.

splines. In Elastix, this is handled by only counting the voxels mapped inside the moving image, terminating the registration if the number of missed voxels is greater than 75% of the total number of voxels used in calculating the similarity metric. It is not immediately clear how this would affect the optimization, as it increases the space of possible transformations; e.g., excluding specific voxels from the metric calculation could be more beneficial. In Elastix, these voxels are set to some default value in the final deformed image. This behavior is undesired, but due to the complexity of adding hard constraints for gradient-based optimizers, this seems like a logical solution. Fortunately, most medical image registration problems include masks of the tissue/organ to be registered; the similarity metric is calculated only for the voxels within this mask, making it unlikely that the transformation maps them outside the moving image. Still, it is a limitation of the gradient-based optimizer, which does not exist for GOMEA.

We have implemented two solutions for GOMEA, which guide the optimizer towards solutions with fewer missed voxels. First, it is possible to enable constrained selection, which requires a threshold for the maximum allowed ratio r of missed pixels per solution as input. Any solution with a ratio above this threshold will have a constraint value $c = r$ and 0 otherwise. GOMEA will favor solutions with $c < r$ over solutions with $c \geq r$ without considering their objective values. Next, we have implemented a missed voxel penalty, which considers the maximal difference d_{max} between the voxel values of each image. With the number of missed voxels n_{missed} and the ratio r interpreted as a percentage, we can calculate the penalty P_{missed} for a particular solution x as:

$$P_{missed}(x) = (n_{missed} * d_{max} * d_{max}) * 2^r. \quad (5.1)$$

The factor 2^r is added to increasingly make the penalty severe as it gets more significant to make sure the optimizer is guided away from these solutions.

5.3.5. Fold Constraints

The B-spline control points can move freely during optimization, making it possible for them to move far from their initial position. As a result, it can occur that adjacent control points move past each other in opposite directions, creating an "overlap", which could lead to a fold in the parameterized transformation. Figure 5.5 shows examples of overlaps for a 2D B-spline grid, highlighting the area in which control points are considered to overlap in relation to their neighbors. These overlaps do not necessarily create folds in the transformation, but if we constrain the optimization by not allowing any overlaps, we can ensure there will be no folds.

We implement overlap constraints in Elastix by determining for each control point if it overlaps with any of its neighbors. The total number of folds is calculated and used as a constraint value by GOMEA, if enabled. An overlap is only counted as such in the dimension where the points differ, according to their initial grid positions. For example, two horizontally adjacent points in a 2D grid need only overlap in the horizontal direction; if they overlap in the vertical direction, no overlap is registered. If they differ in both directions, they must overlap in both (see Figure 5.5).

It is possible to use a multi-resolution schedule for the B-spline grid next to schedules for re-scaling and smoothing the moving and fixed image. At the end of each resolution, the parameters must be up-sampled if the subsequent resolution's grid is larger. During this process, the grid of the current resolution is used to calculate the deformation field for the subsequent resolution. This deformation field is then used to estimate the new B-spline parameters using a decomposition filter. Unfortunately, this filter can introduce inaccuracies, so overlaps may be introduced that did not occur in the coarser grid. To remedy this, we implement a repair procedure that simply "fixes" control points that overlap by moving them just outside the invalid region. Multiple passes over all control points are done until no overlap exist or until a threshold of passes is met. The repair

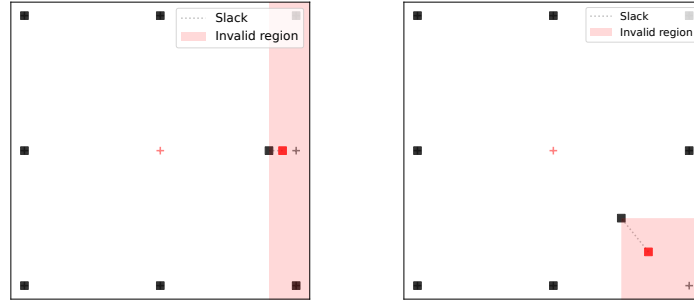


Figure 5.5: Examples of control point overlaps when either two horizontally (left) or diagonally (right) adjacent control points are considered.

procedure is applied to the up-sampled parameters at the start of each resolution if the overlap constraints are enabled.

5.3.6. Hybrid Local Search

We combine the explorative operators of GOMEA with the exploitative operators from ASGD by applying gradient descent on a subset of solutions at the end of each generation. This combination of an EA with a gradient-based method is not novel, and has been applied successfully in numerous settings [1, 6, 13, 14, 82]. Several parameters can be set, which dictate how many solutions gradient descent is applied to and for how many iterations per solution. Additionally, a redistribution method and iteration schedule can be set, which dictate how the solutions are chosen and how the number of generations affects the number of applied iterations. More specifically, we can set the following:

- τ_g : A percentage specifying how many solutions of the population gradient descent is to be applied at the end of each generation.
- **Iteration Schedule:** Can be set as either static or logarithmic. When static, a fixed number of iterations i_g of gradient descent is applied to each solution at the end of each generation. When using the logarithmic schedule, several parameters can be set: i_{max} , i_{min} , i_δ , α , and β . With these parameters, the user can define a logarithmic schedule that increasingly applies more iterations to solutions based on the current generation number. If we take x as the current number of generations, we can define the logarithmic schedule function $L(x)$ as:

$$L(x) = \begin{cases} 0 & x < i_\delta \\ i_{max} - (i_{max} - i_{min}) \frac{1}{1 - \frac{\alpha(x-i_\delta)}{\ln \beta}} & x \geq i_\delta \end{cases} \quad (5.2)$$

The default logarithmic schedule is shown in Figure 5.6.

- **Redistribution Method:** The redistribution method specifies how the solutions to which the local search is applied are chosen at the end of each generation. We implement a method that chooses solutions randomly and a method that chooses the best $\lfloor \tau_g n \rfloor$ solutions according to the objective values and constraints, with n the total number of solutions in the population.

5.3.7. OpenMP

Elastix's similarity metrics and penalty terms can be evaluated in parallel using a thread pool implementation from ITK. This implementation, however, requires multiple functions with callbacks and specialized data structures to keep track of the partial evaluations. To simplify this, we replaced it with procedures from OpenMP [27], which includes specific compiler directives that can be used to orchestrate high-level parallelism intuitively. In addition, it abstracts away all the low-level logic of spawning and managing threads and makes it easy to parallelize existing code by adding only a few lines. This has reduced the architectural complexity of the code while increasing its computational efficiency.

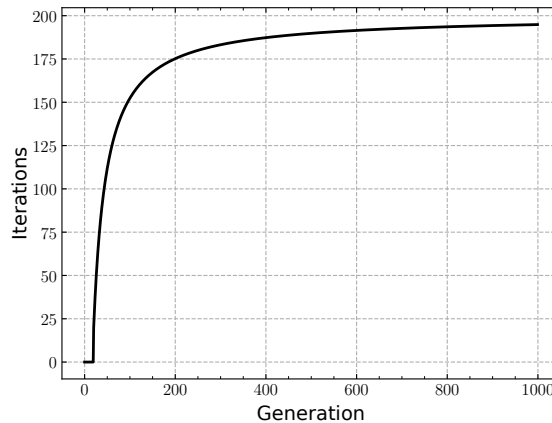


Figure 5.6: Default logarithmic iteration schedule with on the horizontal axis the current number of generations done by GOMEA, and on the vertical axis the number of gradient descent iterations that will be applied to each solution.

5.3.8. Inspection Output

To increase the transparency of the registration process, some minor features are implemented:

- **Image Samples:** When using sub-sampling of voxels to evaluate the cost function, it could help validate that these chosen samples are distributed as expected. Therefore, we have implemented an optional feature for the image samplers, which, if enabled, writes the chosen samples to a file at the end of each iteration.
- **Full Evaluation:** Similarly, when using sub-sampling, it could be helpful if, at the end of the registration process, the cost function is evaluated using all of the voxels instead of a subset. This can make comparing runs with different seeds easier since they are ultimately evaluated similarly. Again, this is implemented and enabled by default.
- **Missed Voxels:** The average percentage of missed voxels across all solutions in an iteration is measured and added as a column to the iteration output when the similarity metrics *MSD* or *NCC* are used.
- **Control Points:** To track the positions of the control points over time, we write their coordinates in physical space to a file at the end of each iteration. This is done for the best solution for that iteration.

5.3.9. Build Process

To build the executable of our implementation, several specific dependencies are required. First, a specific patched version of ITK 5.3 with altered interfaces for partial evaluations. Next, a specific patched version of Plastimatch, with the required alterations to make it compatible with Elastix and partial evaluations. Lastly, there are some additional unpatched and more common dependencies, such as Eigen [35], CBLAS/LAPACK [4], and Boost [74]. To make the build process more accessible, we have implemented it as a superbuild that automatically retrieves and builds ITK, Plastimatch, and Eigen as part of the build process. Due to the wide availability of both CBLAS/LAPACK implementations and Boost, it is up to the user to provide these independently.

6

Analysis

This thesis aims to explore the potential of applying EA-based algorithms for the DIR problem. More specifically, we have implemented RV-GOMEA—from here on referred to as just GOMEA, an evolutionary algorithm capable of exploiting the structure of optimization problems through specialized operators. This chapter analyzes its efficacy in solving synthetic and clinical DIR problems when using common B-spline-based transformation models. First, we provide the general experimental setup in Section 6.1. Subsequently, experiments and results are discussed chronologically in Section 6.2. Lastly, in Section 6.3, we discuss the results of all experiments jointly.

6.1. Experimental Setup

The general experimental setup is outlined in this section by describing the datasets, validation metrics, and specifications of the hardware on which the experiments are run.

6.1.1. Datasets

Synthetic

A synthetic registration problem is constructed using geometric shapes to create a simple test case that is both fast to run (dimensions are small) and easy to understand. The registration consists of shrinking a sphere contained in a fixed-size cube. The total volume consists of $24 \times 24 \times 24$ voxels. The images contain a padding of voxels around the cube to increase the difficulty of finding transformations that correctly map the voxels just in and outside the cube. Landmarks, surface points, and a ground truth Deformation Vector Field (DVF) are generated to validate registration outcomes. The landmarks are placed at specific points on both the surfaces of the cube and sphere; e.g., for the sphere, landmarks are placed at the six positions on its surface where the distance to the edge of the cube is smallest. A visual representation of the synthetic registration problem is given in Figure 6.1. Baseline values for the validation metrics when an identity transform is applied to the moving image are given in Table 6.2.

LEARN

The LEARN Lung CT dataset [43] consists of 30 pairs of intra-patient chest CT scans from the Department of Radiology at the Radboud University Medical Center. With each pair consisting of two scans of a lung after both maximal inspiration and expiration, respectively. The expiration scans do not fully contain the lungs, making the registration more difficult. Additionally, the breathing motions of the patients are large, further increasing the difficulty of finding their correct deformation. The 30 pairs are subdivided into 20 training and 10 test pairs. This thesis only considers patients 1-3, as they include landmarks. See Table 6.1 for pairs of slices of each patient. The resolution of all scans is $192 \times 192 \times 208$ with voxel sizes $1.75 \times 1.75 \times 1.75$ mm. Baseline values for the validation metrics when an identity transform is applied to the moving image are given in Table 6.2.

6.1.2. Validation Metrics

Target Registration Error

The Target Registration Error (TRE) is defined as the mean Euclidean distance between corresponding pairs of landmarks after registration. The transformation is defined as a mapping of points from the fixed image to

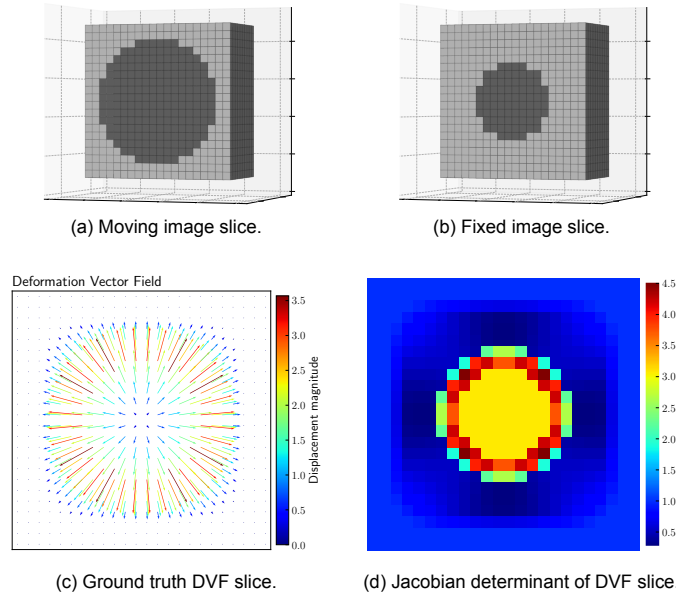


Figure 6.1: Synthetic registration problem: A sphere within a cube is shrunk while the cube stays fixed.

points on the moving image. With the transformation T_μ after registration and the fixed and moving landmark points L_F, L_M , we can calculate the TRE as follows:

$$TRE(T_\mu, L_F, L_M) = \frac{1}{|L_F|} \sum_{i=1}^{|L_F|} |T_\mu(L_F(i)) - L_M(i)|, \quad (6.1)$$

with $L_F(i), L_M(i)$ the i th point of the fixed and moving landmarks respectively.

Average Surface Distance

The average surface distance (ASD) measures the average distance between the surfaces of the fixed and deformed moving image, respectively. This is done by first generating points that approximate the surfaces of shapes in the fixed and moving image. The ASD is then calculated by measuring the average of the distances between the deformed points of the fixed image surfaces with their closest point on the moving image surfaces, as well as the average of the distances between the points on the moving image surfaces with their closest points of the deformed fixed image surfaces. If we take S a set of surface points, and $T_\mu(S)$ the transformed surface points from S using parameters μ , and $\bar{d}(S', S)$ to be the average of distances between points from S' with their closest point in S , we can define the ASD as:

$$ASD(T_\mu, S_M, S_F) = \frac{1}{2} (\bar{d}(T_\mu(S_F(i)), S_M(i)) + \bar{d}(S_M, T_\mu(S_F))). \quad (6.2)$$

Hausdorff Distance

The Hausdorff distance (HD), similar to the ASD, uses surface points to measure the alignment of shapes after registration. Instead of taking the average of the minimum distances, the Hausdorff distance is calculated by taking the maximum of the minimal distances between points for a particular surface. Intuitively, it measures the worst-case distance between two surfaces. If we take $d(x, S) = \min_{y \in S} d(x, y)$ to be the distance from point x to its closest point in Y , and S_M, S_F the sets of surface points, we can define the HD as:

$$HD(T_\mu, S_M, S_F) = \max\left\{ \max_{x \in T_\mu(S_F(i))} d(x, S_M(i)), \max_{y \in S_M(i)} d(y, T_\mu(S_F(i))) \right\}. \quad (6.3)$$







Instance	Moving	Fixed
Patient 1		
Patient 2		
Patient 3		

Table 6.1: Slices of instances 1-3 from the LEARN [43] dataset.

Dice Similarity Coefficient

The Dice similarity coefficient (DSC) measures the spatial overlap between the ground truth segmentation of the fixed image, with the segmentation that follows from the deformation of the moving image. It is defined as:

$$DSC(X, Y) = \frac{2|X \cap Y|}{|X| + |Y|}, \quad (6.4)$$

with, in the image registration case, X and Y as sets of voxels defining the segmentations.

Bending Energy

The bending energy measures the sum of second-order spatial derivatives for transforming the moving image, as defined by the DVF. It does not necessarily measure the overall magnitude of the transformation but rather the smoothness or regularity of the transformation locally. Any sudden changes or sharp turns in a transformation are undesirable, as they are often not anatomically feasible.

The bending energy can be calculated from the DVF by summing over the spatial Hessians in each dimension for each deformation vector. These Hessians are approximated using finite differencing with the vectors of nearby voxels in each direction. If we take $b(v, F)$ to be the bending energy of vector v from the DVF F , we can define the bending energy as:

Metric	SYN	LEARN-1	LEARN-2	LEARN-3
TRE	1.50	20.18	15.60	8.14
ASD	Cube: 0.00	-	-	-
	Sphere: 3.00	-	-	-
DSC	Cube: 0.88	0.84	0.86	0.92
	Sphere: 0.40	-	-	-
\vec{v}_ϵ	0.49	-	-	-

Table 6.2: The baseline values of the validation metrics for both the synthetic registration problem (labeled SYN) and patients 1-3 of the LEARN [43] dataset (labeled LEARN-[1-3]). The baselines are calculated for the identity transform; we can exclude BE , $SDLogJ$, and HD since $BE = SDLogJ = 0.0$ and $HD = ASD$ (for the synthetic problem). The ASD , HD , and \vec{v}_ϵ cannot be calculated for the LEARN cases as no surface points or ground truth DVF is available. The TRE , ASD , and HD are calculated in *mm* using the respective spacing of each instance.

$$E_b(F) = \frac{1}{|F|} \sum_{v \in F} b(v, F), \quad (6.5)$$

$|F|$ is the total vector amount in F , which equals the number of voxels in the fixed image.

DVF Error

The registration of a moving and fixed image is ultimately performed using a Deformation Vector Field (DVF), which maps the voxels of the fixed image domain to the voxels of the moving image domain. It is defined by storing a displacement in each dimension for each voxel of the fixed image and can be used to produce the deformed moving image. The DVF error \vec{v}_ϵ can be calculated if a ground truth DVF for the registration problem is available, i.e., a physically plausible transformation that accurately approximates the fixed image. The DVF error measures the difference between a provided DVF with the ground truth DVF for any registration problem. For example, if we have a registration problem and a corresponding ground truth DVF F_{ref} , we can calculate the error of any given DVF F as:

$$\vec{v}_\epsilon(F) = \frac{1}{|F_{\text{ref}}|} \sqrt{\sum_{v_i \in F} (F_{\text{ref}}(i) - v_i)^2}. \quad (6.6)$$

6.1.3. Hardware

The experiments are executed on the SHARK computing cluster of the Leiden University Medical Center (LUMC). For each experiment, a workload with access to 8 Central Processing Unit (CPU) cores and 6 gigabytes (GB) of Random Access Memory (RAM) is reserved—most execution nodes in the SHARK cluster run on either an Intel E5-2690 or E5-2697.

6.2. Experiments

6.2.1. Single-resolution Comparison to Gradient-based Optimization

Gradient-based optimizers are commonly used in DIR due to their robust performance, and computational efficiency for high-dimensional registration problems [48]. As a baseline comparison, we will compare the performance of GOMEA with the recommended Adaptive Stochastic Gradient Descent (ASGD) optimizer available in Elastix.

Due to its population-based approach, it is expected that GOMEA will outperform ASGD in terms of the objective value; ASGD optimizes a single solution iteratively using stochastic gradient descent, which makes it more prone to local minima traps. The population-based approach of GOMEA has a higher likelihood of “jumping” over these local minima, enabling it to outperform in terms of the objective value. However, this does not necessarily mean that it will outperform in terms of the validation metrics, which are not optimized directly.

Setup We register the synthetic registration problem (Figure 6.1) using no regularization to keep the comparison straightforward. The Mean Squared Difference (MSD) is used as the similarity metric. To analyze the effect

of increasing flexibility in the transformation, mesh sizes of $6 \times 6 \times 6$, $7 \times 7 \times 7$, and $8 \times 8 \times 8$ B-spline control points are used. No scalability analyses are done, as it is not expected that GOMEA will outperform the gradient-based optimizer in terms of computational efficiency due to its population-based approach. GOMEA uses the marginal linkage model \mathcal{F}_{MP} for this experiment. Each run is repeated ten times using different seeds.

Results In Figure 6.2, the convergence of both optimizers regarding the MSD is plotted for each mesh size. For the smaller mesh sizes of $6 \times 6 \times 6$ and $7 \times 7 \times 7$, we can visually see that GOMEA and ASGD perform almost equally. For the largest mesh size, we can see that GOMEA outperforms in terms of the MSD. Intuitively, it makes sense that an increase in the flexibility of the deformation model would make it more likely for a gradient-based optimizer to get stuck in local optima. Suppose there are more degrees of freedom in solving the registration problem. In that case, the landscape of its objective value as a function of the parameters can only become more complex and harder to optimize. From this, we can see that even for a simple synthetic registration problem, GOMEA can provide a marked difference in the objective value when using a B-spline-based deformation model. We additionally calculate the T-Test statistics for the differences in metric values for each mesh size to confirm that GOMEA provides a significant improvement in terms of the metric value also for the smaller mesh sizes ($t(18) = -10.69, p = 2.04 \cdot 10^{-6}$; $t(18) = -6.97, p = 7.31 \cdot 10^{-5}$; $t(18) = -29.73, p = 2.25 \cdot 10^{-10}$ for the mesh sizes $6 \times 6 \times 6$, $7 \times 7 \times 7$, and $8 \times 8 \times 8$ respectively).

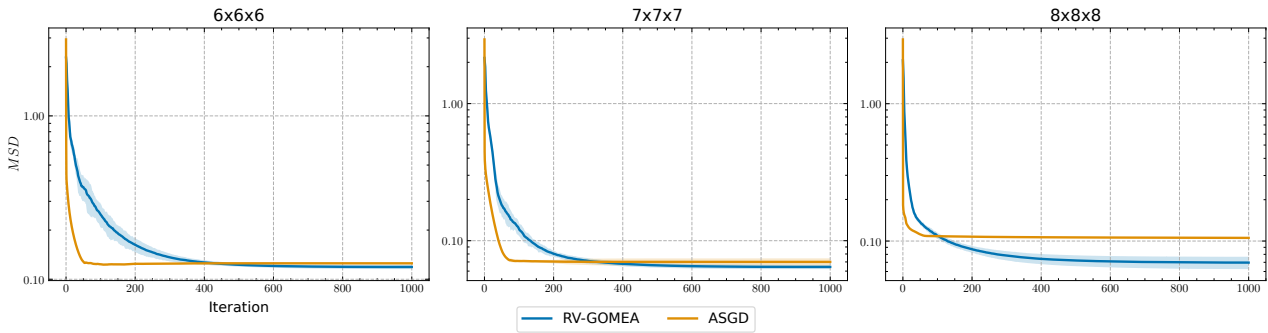


Figure 6.2: Convergence plots for the mesh sizes $6 \times 6 \times 6$, $7 \times 7 \times 7$, and $8 \times 8 \times 8$ when no multi-resolution scheme is used. The objective values for GOMEA and ASGD are plotted jointly using a normalized range of iteration values. For each optimizer and mesh size, the average values are plotted in combination with a 95% confidence interval band.

The results in terms of the validation metrics are shown in Table 6.3. Only for the smallest mesh size, $6 \times 6 \times 6$, we can see that GOMEA outperforms ASGD in all metrics except the DSC_{sphere} . On the other hand, for the larger mesh sizes, ASGD consistently outperforms in all validation metrics except for the DSC_{sphere} . This can be explained by returning to its definition previously defined in Section 6.1.2. From it, we can see that the DSC measures the overlap of segmentations between the target and deformed source image. For the synthetic registration problem, these segmentations can be calculated using a multi-level thresholding algorithm to discretize the images into voxels of 3 distinct values. The MSD, in a sense, can be seen as a more granular version of this metric, as it measures not the overlap but the difference in voxel values on a per-voxel basis; if the difference between two voxel values is small enough, they will get discretized into the same level and provide a positive contribution towards the DSC . Given that GOMEA outperforms ASGD for all mesh sizes in terms of the MSD , it makes sense that it also outperforms in the DSC .

Combined with the validation metrics, the results indicate that a lower value in terms of the MSD does not necessarily improve the registration outcome. A lower value in terms of the MSD will lead to outcomes that look visually appealing. However, the required underlying transformation becomes increasingly complex and anatomically infeasible to get these outcomes. This relates to the overfitting problem often encountered in the field of Machine Learning [42] when a model is used to solve a problem that is more flexible than it needs to be. Our model uses a common B-spline-based deformation model combined with the MSD as the similarity metric. This model, as we now can see, also provides too much flexibility, which can lead to registration outcomes that are overfitting to the MSD. As such, it makes sense that GOMEA outperforms the validation results for the smallest mesh size of $6 \times 6 \times 6$ as it provides fewer degrees of freedom, decreasing the amount of overfitting.

In Figure 6.3, visualizations of specific run results are shown for both optimizers when the largest mesh size is used. From the deformed source images, we can see that GOMEA does provide the best result, as we expected from the values of the MSD and DSC. From the slices of the control points, deformation vector field

Metric	6x6x6		7x7x7		8x8x8	
	GOMEA	ASGD	GOMEA	ASGD	GOMEA	ASGD
TRE	$0.73 \pm 1.1e-1$	$1.04 \pm 1.6e-3$	$2.10 \pm 2.8e-1$	$0.84 \pm 1.5e-1$	$1.97 \pm 1.5e-1$	$1.08 \pm 2.3e-3$
ASD_{cube}	$0.39 \pm 7.1e-3$	$0.52 \pm 4.8e-4$	$0.43 \pm 1.4e-2$	$0.36 \pm 7.3e-3$	$0.57 \pm 2.1e-2$	$0.46 \pm 2.0e-3$
ASD_{sphere}	$0.31 \pm 9.2e-3$	$0.31 \pm 4.2e-5$	$0.28 \pm 1.3e-2$	$0.26 \pm 1.4e-2$	$0.28 \pm 6.1e-3$	$0.31 \pm 5.6e-5$
DSC_{sphere}	$0.92 \pm 3.2e-3$	$0.94 \pm 2.3e-16$	$0.98 \pm 1.9e-3$	$0.97 \pm 5.1e-3$	$0.97 \pm 7.8e-3$	$0.95 \pm 2.7e-3$
E_b	$0.08 \pm 1.1e-2$	$0.20 \pm 7.5e-4$	$0.62 \pm 1.8e-1$	$0.11 \pm 1.6e-2$	$0.76 \pm 6.4e-2$	$0.19 \pm 1.7e-3$
\vec{v}_ϵ	$0.68 \pm 3.4e-2$	$0.70 \pm 5.1e-4$	$1.13 \pm 1.5e-1$	$0.74 \pm 2.4e-2$	$1.11 \pm 3.7e-2$	$0.76 \pm 1.0e-3$

Table 6.3: Validation metrics comparison for mesh sizes 6x6x6, 7x7x7, and 8x8x8 between GOMEA and ASGD on the synthetic instance when no multi-resolution scheme is used. Bold values indicate a statistically significant improvement (p value < 0.05).

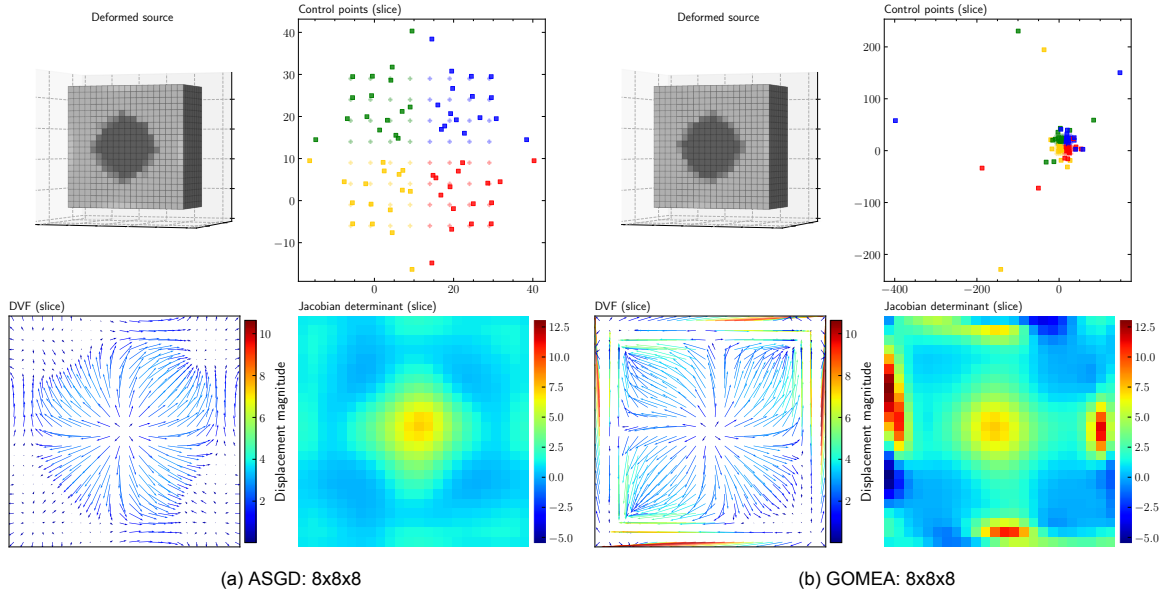


Figure 6.3: Visualization of specific run results from comparing GOMEA with ASGD using a mesh size of 8x8x8 and no multi-resolution scheme.

(DVF), and Jacobian determinant, we see why ASGD outperforms in terms of the validation metrics. Firstly, the final configuration of control points seems a lot more natural and almost symmetric for ASGD. In contrast, the configuration of control points for GOMEA contains huge outliers and overall seems more erratic.

Similarly, the Jacobian determinant shows that the transformation found by ASGD does not contain any folds, whereas the transformation found by GOMEA does (indicated by negative values in the Jacobian determinant). Moreover, from Table 6.3, we find the average DVF error \vec{v}_ϵ to be 0.76 for ASGD and 1.11 for GOMEA, which is quite a significant difference; note that this error is calculated by taking the root mean squared error (RMSE) with the ground truth as presented in Figure 6.1. Furthermore, the bending energy, which measures the overall complexity of the transformation by summing over its second-order derivatives, is also significantly higher for GOMEA; this indicates that the transformation contains many sudden changes and sharp turns, further decreasing the likelihood of its anatomic feasibility. Overall, we can conclude that GOMEA is overfitting on the MSD for the synthetic registration problem, which suggests that the use of a regularization term can prove beneficial. Opposedly, we can also state that ASGD underperforms as it converges prematurely in local minima. In DIR, a multi-resolution scheme is often used to overcome the issue of premature convergence. As such, in our next experiment, we analyze the efficacy of such a scheme when applied to the synthetic registration problem.

6.2.2. Multi-resolution Comparison to Gradient-based Optimization

To decrease the likelihood of the premature convergence local optima, it is common to optimize the registration problem using a multi-resolution scheme [51, 54]. These schemes consist of hierarchical levels which apply an increasing level of down-sampling or a gaussian blur. This creates a hierarchy of images with details ranging from coarse to fine. The registration is then performed top-down, starting by registering the most global image, thereby hopefully avoiding local minima traps. In this experiment, we test this hypothesis for the synthetic registration problem and compare the performance of a gradient-based algorithm with GOMEA.

Setup We register the synthetic registration problem without regularization using mesh sizes of 6x6x6, 7x7x7, and 8x8x8 control points. Again, we use the MSD as the similarity metric, and for GOMEA, the marginal linkage model \mathcal{F}_{MP} . We use a multi-resolution scheme of three levels in which only smoothing is applied according to a linearly decreasing schedule. Each run is repeated ten times using different seeds.

Results A comparison with the single-resolution results in terms of the MSD is given in Table 6.4. For each mesh size, using a multi-resolution scheme significantly improves the found MSD . Convergence plots of the final resolution for each mesh size show that, on average, both optimizers converge on similar values for the MSD (Figure 6.4). Visually, we see a wide confidence interval for GOMEA and a narrow interval for ASGD. This is possibly caused by GOMEA relying much more on stochasticity than ASGD, as new solutions are sampled using estimated normal distributions. For the mesh sizes of 6x6x6 and 8x8x8, the differences in final mean MSD values are not statistically significant, whereas the differences in mean MSD values for the mesh size of 7x7x7 are. The t-test statistics are $t(18) = 2.78e-2, p = 0.98$; $t(18) = -2.34, p = 0.04$; $t(18) = 0.93, p = 0.38$ for the mesh sizes of 6x6x6, 7x7x7, and 8x8x8 respectively.

Metric	6x6x6		7x7x7		8x8x8	
	$R_n = 1$	$R_n = 3$	$R_n = 1$	$R_n = 3$	$R_n = 1$	$R_n = 3$
GOMEA	$0.12 \pm 1.9e-3$	$0.05 \pm 2.5e-3$	$0.06 \pm 1.2e-3$	$0.02 \pm 1.9e-3$	$0.07 \pm 3.8e-3$	$0.02 \pm 7.3e-3$
ASGD	$0.13 \pm 5.9e-6$	$0.05 \pm 9.6e-5$	$0.07 \pm 2.3e-3$	$0.02 \pm 4.3e-6$	$0.11 \pm 2.8e-4$	$0.02 \pm 1.5e-4$

Table 6.4: Comparison of the final optimized MSD values for either $R_n = 1$, i.e. no multi-resolution scheme is used. Or $R_n = 3$, i.e., a multi-resolution scheme of 3 levels is used. Bold values indicate a statistically significant improvement (p value < 0.05).

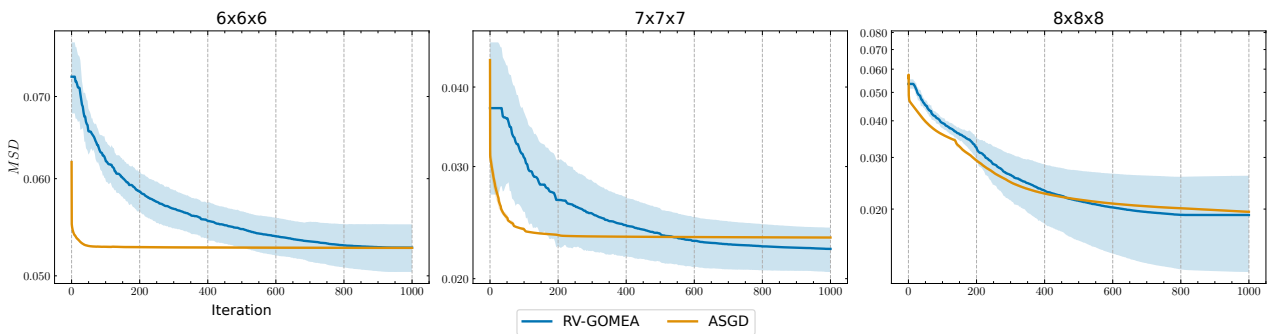


Figure 6.4: Convergence plots of the final resolution when a 3-level multi-resolution scheme is used. Objective values for GOMEA and ASGD are plotted jointly using a normalized range of iteration values. For each optimizer and mesh size, the average values are plotted in combination with a 95% confidence interval band.

In terms of the validation results, Table 6.5 shows that ASGD outperforms GOMEA for all mesh sizes. Notably, the registration results found by ASGD with a mesh size of 7x7x7 are with the smallest ground truth error \vec{v}_ϵ measured thus far. One of these runs is visualized in Figure 6.5 and compared to a run with the same settings optimized using GOMEA. With a multi-resolution scheme, the final configuration of control points for ASGD almost seems perfectly symmetrical. This was already visible when no multi-resolution scheme was used. However, now with the addition of consecutive levels consisting of a decreasing amount of gaussian

blur, this effect seems reinforced. In part, this should also be attributed to the inherent symmetrical nature of the synthetic registration problem, but the effect is interesting nonetheless.

On the other hand, the result found by GOMEA in Figure 6.5 seems worse in all aspects, which is confirmed by and attests to the validity of the validation metrics in Table 6.5. Overall, these results show that using a multi-resolution scheme can improve the registration results significantly when using a gradient-based optimizer. For GOMEA, we see improved results for the MSD but not necessarily for the validation metrics.

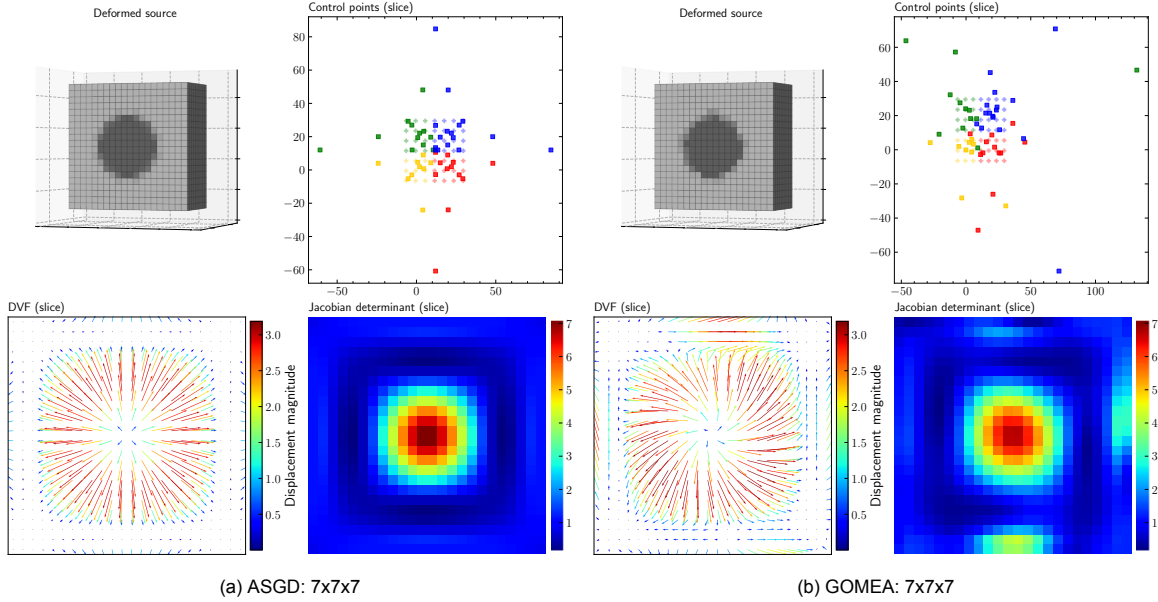


Figure 6.5: Visualization of specific run results from the comparison of GOMEA with ASGD when using a mesh size of $8 \times 8 \times 8$ and a multi-resolution scheme consisting of 3 levels using only a linearly decreasing gaussian blur.

It is also interesting that for the MSD, both optimizers find values that are mostly not significantly different, whereas the validation metrics are. This further demonstrates that a similarity metric alone is not enough to guide GOMEA towards high-quality transformations. From Figure 6.5 and Table 6.5, we can conclude that ASGD is inherently better at finding more regular and natural-looking transformations when applied to the synthetic registration problem, probably due to its gradient-based operation.

Metric	6x6x6		7x7x7		8x8x8	
	GOMEA	ASGD	GOMEA	ASGD	GOMEA	ASGD
TRE	$0.83 \pm 1.7e-1$	$0.44 \pm 1.4e-3$	$1.23 \pm 5.0e-1$	$0.31 \pm 9.9e-4$	$1.92 \pm 2.5e-1$	$0.74 \pm 2.9e-3$
ASD_{cube}	$0.38 \pm 9.3e-3$	$0.39 \pm 1.8e-3$	$0.36 \pm 2.2e-2$	$0.21 \pm 1.6e-3$	$0.45 \pm 9.3e-3$	$0.42 \pm 3.5e-3$
ASD_{sphere}	$0.30 \pm 3.7e-3$	$0.33 \pm 1.0e-4$	$0.26 \pm 2.1e-2$	$0.29 \pm 2.1e-4$	$0.28 \pm 1.2e-2$	$0.27 \pm 6.1e-4$
DSC_{sphere}	$0.92 \pm 4.3e-3$	$0.95 \pm 1.2e-16$	$0.97 \pm 3.9e-3$	$0.98 \pm 1.2e-16$	$0.98 \pm 5.9e-3$	1.00 ± 0.0
E_b	$0.09 \pm 1.6e-2$	$0.05 \pm 1.6e-4$	$0.23 \pm 1.9e-1$	$0.05 \pm 3.4e-5$	$0.69 \pm 1.7e-1$	$0.28 \pm 2.9e-3$
\vec{v}_ϵ	$0.70 \pm 3.7e-2$	$0.53 \pm 1.5e-4$	$0.86 \pm 1.9e-1$	$0.56 \pm 1.4e-4$	$1.15 \pm 1.6e-1$	$0.76 \pm 2.9e-3$

Table 6.5: Validation metrics comparison for mesh sizes $6 \times 6 \times 6$, $7 \times 7 \times 7$, and $8 \times 8 \times 8$ between GOMEA and ASGD on the synthetic instance when a multi-resolution scheme of three levels is used. Bold values indicate a statistically significant improvement (p value < 0.05).

6.2.3. Linkage Models

GOMEA can use a linkage model to exploit the dependencies between specific problem variables during optimization (see Section 3.2.2). In Section 4.2, we defined four specific models appropriate for deformable image registration when using a B-spline-based transformation model.

Setup This experiment aims to compare their performance when given an evaluation budget. The composition of each linkage set determines the number of voxels for which the similarity metric needs to be partially evaluated; thus, it makes sense to limit the optimizations using a budget of total voxel evaluations. We also measure the impact of the number of parameters by using grids of $8 \times 8 \times 8$, $11 \times 11 \times 11$, and $13 \times 13 \times 13$ control points. An increasing number of parameters should affect the full linkage model \mathcal{F}_{Full} negatively, as it will require increasingly bigger population sizes to estimate the dependencies successfully. We expect the univariate model \mathcal{F}_{Uni} to provide the worst objective values when given an evaluation budget due to increased voxel evaluations. \mathcal{F}_{Uni} models each of the parameters as independent, i.e., for each control point, a linkage set exists in each dimension. For the static linkage model \mathcal{F}_{SLT} , we use linkage sets with sizes between 3 and 24. Experiments are repeated ten times using different seeds.

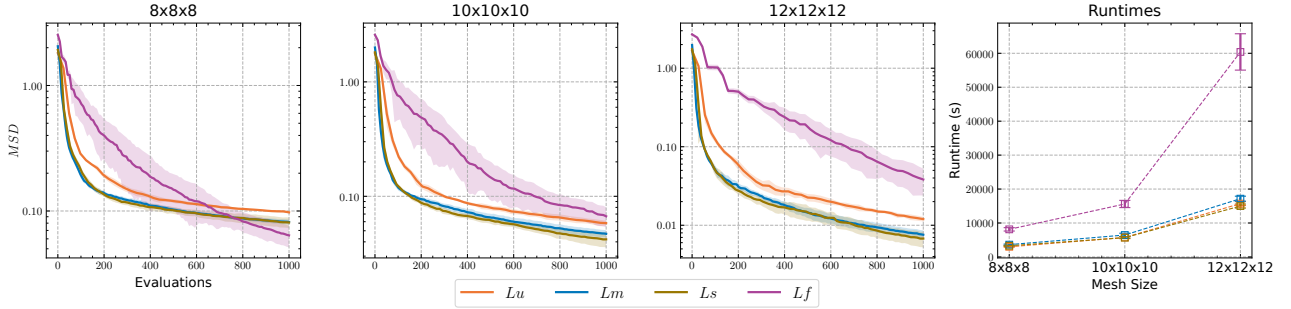


Figure 6.6: Average MSD with error bands plotted against evaluations for runs using various linkage models and grids of $8 \times 8 \times 8$, $10 \times 10 \times 10$, and $12 \times 12 \times 12$ control points (left). We also compare each linkage model's average run-time in seconds and mesh size (right).

Results Figure 6.6 show the results for each mesh size by plotting the MSD against the number of voxel evaluations. We also compare their run times in seconds. The univariate, marginal, and static linkage models seem to perform similarly, with the full linkage model outperforming for the smallest mesh size. However, for the largest mesh size, the full linkage model is outperformed due to the budget of voxel evaluations. Probably due to the large population size required for the Cholesky factorizations [44] of the full covariance matrix—which scales by a factor of $l^{1.5}$ with l the number of parameters [15]. Between the univariate, marginal, and static linkage models, we see a significant improvement for the MSD when using either the marginal or static model. Both model dependencies between the parameters of each control point without considering these independently, which leads to fewer evaluations of the same region of voxels per generation. Moreover, they are capable of exploiting the linkage that may exist between these variables.

Metric	Linkage Model			
	\mathcal{F}_{Uni}	\mathcal{F}_{MP}	\mathcal{F}_{SLT}	\mathcal{F}_{Full}
TRE	1.81 ± 0.03	1.45 ± 0.09	1.64 ± 0.17	3.31 ± 0.31
ASD_{cube}	0.64 ± 0.01	0.49 ± 0.02	0.49 ± 0.01	0.74 ± 0.03
ASD_{sphere}	0.29 ± 0.01	0.28 ± 0.01	0.27 ± 0.01	0.46 ± 0.04
DSC_{sphere}	0.94 ± 0.00	0.96 ± 0.01	0.97 ± 0.01	0.98 ± 0.00
E_b	0.50 ± 0.02	0.39 ± 0.03	0.45 ± 0.04	1.98 ± 0.31
\vec{v}_ϵ	0.58 ± 0.01	0.59 ± 0.05	0.61 ± 0.06	1.75 ± 0.16

Table 6.6: Validation metrics if we use GOMEA with various linkage models and a mesh size of $8 \times 8 \times 8$. Bold values indicate a (joint) statistically significant improvement over all other models. The average surface distance (ASD) and Hausdorff distance (HD) are given for the sphere and cube separately.

We have included the validation metrics for the lowest mesh size of $8 \times 8 \times 8$ in Table 6.6. We omitted the results from the larger mesh sizes for brevity; the validation results are similar across all mesh sizes. The marginal linkage model \mathcal{F}_{MP} performs best overall. The full linkage model \mathcal{F}_{Full} finds the lowest MSD value

within the evaluation budget for this mesh size but has the worst value for each validation metric. Similar to the results from Section 6.2.1, this suggests that the model is over-fitting in some way, either because of its increased population size or the exploitation of dependencies between all problem variables at once. Overall, the marginal model seems the most practical and cost-efficient; it is computationally scalable regarding the number of parameters and can find decent solutions in terms of the objective value and validation metrics. The static model performs reasonably well but will have the same problems as the full linkage model when the maximum linkage set size increases.

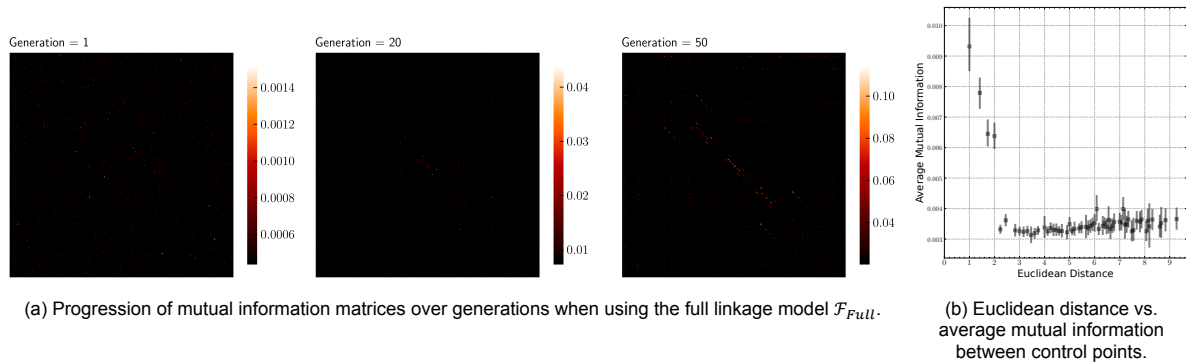


Figure 6.7: The linkage between control points visualized using heatmaps of the mutual information for specific generations. Also, the relationship between the euclidean distance and mutual information of control points—here, the euclidean distances are binned, and standard confidence intervals are shown.

Figure 6.7 additionally gives some insight into the linkage between problem variables for the synthetic registration problem. After some generations, a clear pattern emerges in the heatmaps with dependencies along the diagonal. The plot on the right (b) shows these dependencies mainly exist between adjacent control points. Moreover, the dependencies are mostly visible for control points around the center of the grid; this makes sense, as these are most influential in transforming the voxels inside the cube.

6.2.4. Missed Voxels

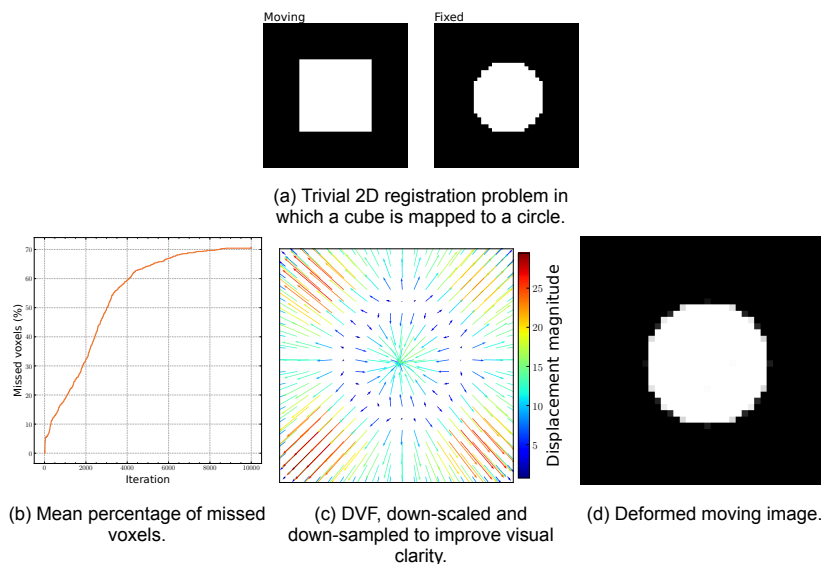


Figure 6.8: Visualization of a registration for a trivial 2D problem using ASGD as the optimizer. The optimizer converges on a transformation that maps voxels from each cube corner to positions outside the moving image.

In order to calculate the value of a similarity metric, such as the MSD, we need to interpolate the mapped voxel values from the moving image. However, these mappings may fall outside the moving image when using a deformation model based on B-splines. In Elastix, this is handled by only counting the voxels that are mapped inside of the moving image, terminating the registration if the number of missed voxels is greater than 75% of

the total number of voxels set to be used in calculating the similarity metric. It is not immediately clear how this would affect the optimization, as it can only increase the space of possible transformations; it could be more beneficial to exclude specific voxels from the metric calculation.

We created a trivial 2D synthetic registration problem consisting of a cube mapped to a circle for demonstration purposes. The problem is registered using ASGD on a mesh size of $7 \times 7 \times 7$, which results in a transformation with 70% missed pixels that may visually look correct but ultimately "solves" the problem by mapping voxels that differ in value to positions outside the moving image, see Figure 6.8. Of course, this is a stylized example designed to elicit this behavior. Still, it does show that missed voxels can have an undesired impact on the registration outcome. Therefore, we implement both a missed voxel threshold and penalty to guide GOMEA towards solutions with reduced or no missed voxels; for implementation details, see Section 5.3.4.

Setup We test the implemented methods from Section 5.3.4 by registering the synthetic registration problem when either constrained selection with various percentage thresholds $\in \{0.0, 0.5, 2, 5, 10\}$ or the missed pixel penalty is used. Registrations are done using GOMEA, a mesh size of $7 \times 7 \times 7$, without any multi-resolution scheme or regularization. We use the MSD as the similarity metric. Each run is repeated ten times with different seeds.

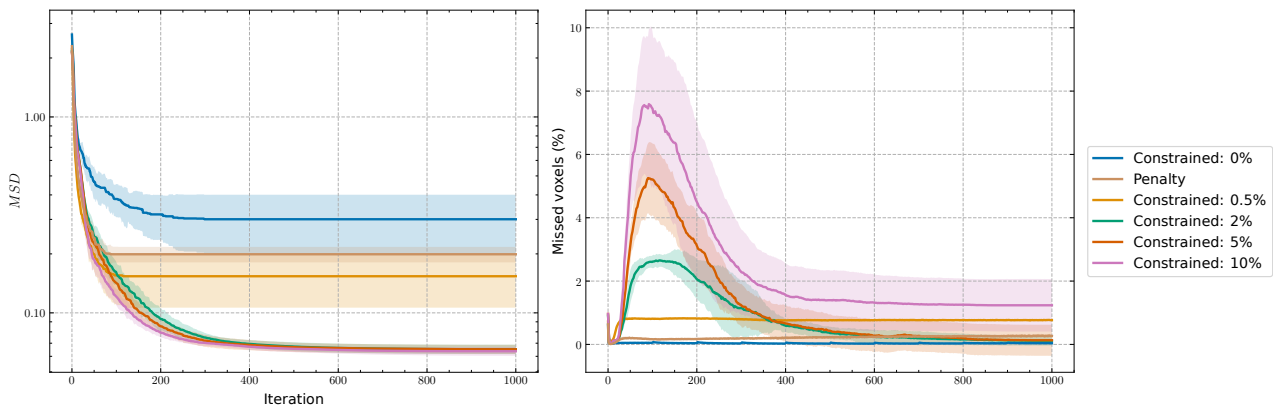


Figure 6.9: Mean iteration output of the MSD and missed voxels percentage for runs using GOMEA on the synthetic registration problem. A mesh size of $7 \times 7 \times 7$ with either constrained selection or a missed voxel penalty is used.

Results Figure 6.9 shows convergence plots for the MSD and the average missed voxel percentage. The outcomes clearly show that by using increased thresholds for the constrained selection, we can improve outcomes in terms of the MSD . It suggests that the optimizer exploits the additional freedom given by larger thresholds to find solutions with lower objective values. The convergence plots show an initial impulse in the average of missed voxels during the early stages of each run, indicating that some of the increased search space is exploited there. After this initial impulse, each run seems to converge to some equilibrium at a value lower than the chosen threshold. This suggests that the lower metric values are not simply achieved by the decreased number of voxels used to calculate it; if this were true, missed voxel percentages for each run would converge to each of their respective thresholds. Instead, they converge to some lower percentage but with significantly lower MSD values.

Figure 6.10 shows the difference in registration outcomes between two runs with different missed pixels thresholds. We can see that when we use a lower threshold, the optimizer struggles more with finding suitable mappings for the voxels of the sphere and around the cube. This suggests that with a larger margin for missed pixels, the optimizer can find better registration outcomes with fewer mismatches in voxel mappings. The artifacts around the cube indicate that the differences mainly come from the outer control points. By giving these control points more freedom to move farther away from their initial positions at the start, the optimizer can converge to equilibria where voxels just in and outside the cube are mapped correctly. The inner control points can move around more freely if the outer control points are sufficiently far away, ultimately leading to configurations that result in better registration outcomes. The DVFs also show some interesting differences: the lower threshold has some cyclical characteristics, whereas the higher threshold is straighter and more regular.

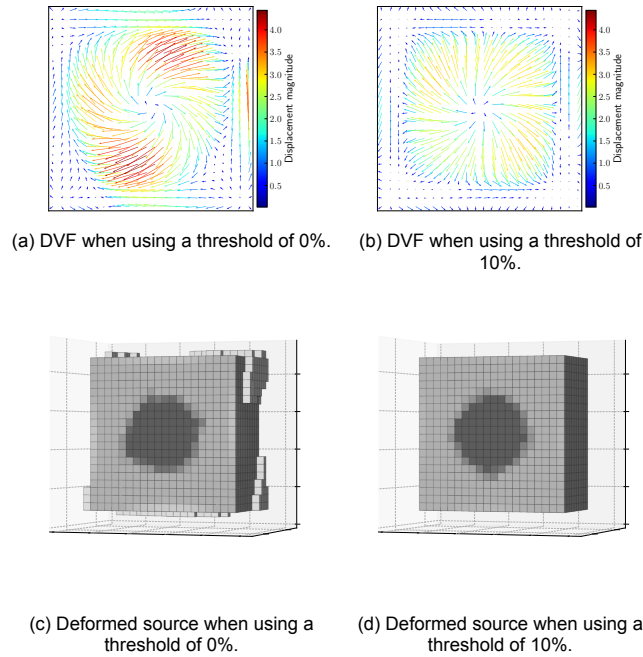


Figure 6.10: Deformation Vector Fields (DVF) and registration outcomes when using either a missed voxels threshold of 0% or 10%.

For the lower threshold, the inner control points are cramped by the outer control points and, in turn, forced into a more cyclic pattern.

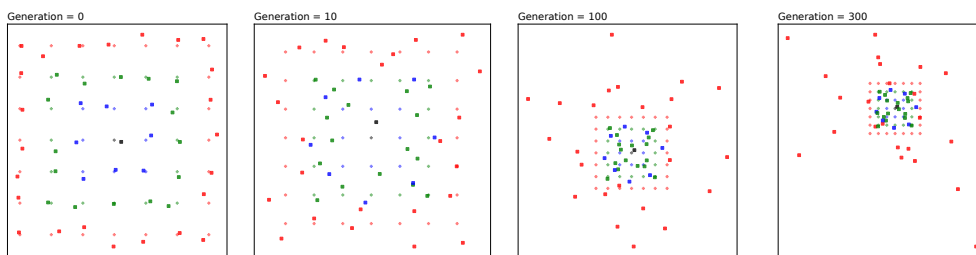


Figure 6.11: Slices of the control point configurations at the end of specific generations when using no constrained selection or missed voxel penalty. All other settings are the same as for the runs from Figure 6.9.

To gain additional insight into the optimizer’s behavior, we save the configuration of control points for the best solution from the population at the end of each generation. Figure 6.11 shows configuration slices for four specific generations, illustrating their positions’ progression over time. We use a different coloring scheme in which the control points are grouped by their minimal distance to the center in either direction to differentiate the outer and inner control points. We can label these by their level ordered inwards to outwards— i.e., the center control point is level 0, and the outermost control points are level 3. As expected, the plots show the level 3 points moving from their initial position towards some final configuration. More interestingly, we can see that some level 2 control points converge towards the center, whereas the level 1 control points move past them in the other direction. If the control points move sufficiently past each other in reverse directions, they can create folds locally in the transformation, which is undesired behavior in a clinical context. Here, the outer control points are sufficiently spaced outwards so that no folds exist. Nevertheless, it shows that it could be beneficial to further constrain the optimizer by not allowing control points to overlap.

Overall, it has become clear that the missed voxels can provide substantial improvements by increasing the space in which optimizers can configure the control points. It will vary depending on the registration problem; if an image mask is used—which is the case in most clinical contexts, missed voxels will not be an issue. Still, the additional freedom for the outer control points could prove beneficial even if an image mask is used. Therefore, we suggest not using a missed voxel constraint or penalty for most registration problems.

6.2.5. Fold Constraints

We can additionally constrain the registration by not allowing local overlaps regarding the control points. From Figure 6.11, we can see that this happens when the inner control points at two consecutive levels move past each other in opposite directions (the blue and green control points). These overlaps do not necessarily create folds in the transformation, but there will be none if there are no overlapping control points. It could be the case that these overlaps are ultimately necessary to find adequate registrations due to the nature of B-spline-based transformation models. In any case, it will be interesting to see the impact of such a constraint on the optimizer. We expect it to lead to more regular transformations, as it removes the possibility of a negative spatial Jacobian value at any position in the transformation. We also expect it leads to worse outcomes in terms of the similarity metric—and, in turn, the visual outcome of the registration, as it strictly decreases the space of possible transformations. Nevertheless, the increased regularity of the underlying transformations could lead to an improvement in the validation metrics.

Setup We measure the impact of constraining the control points for registrations of the synthetic registration problem using mesh sizes of $7 \times 7 \times 7$, $8 \times 8 \times 8$, and $9 \times 9 \times 9$. The results are compared to registrations without constraints; in both cases, no regularization or multi-resolution scheme is used. We use the MSD as the similarity metric, and each run is repeated ten times using different seeds.

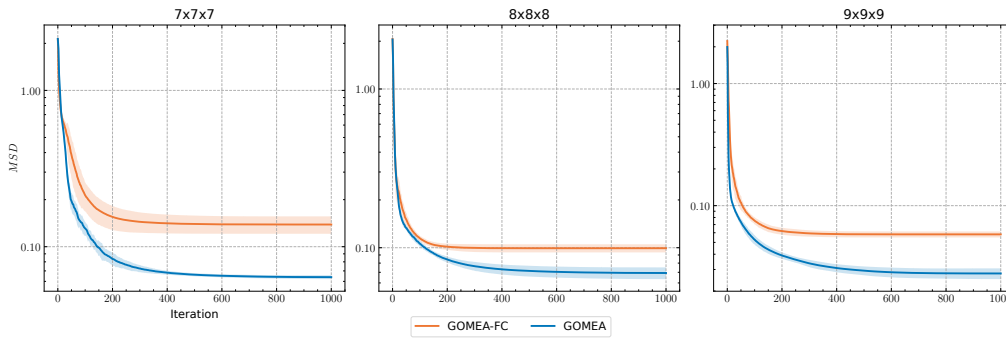


Figure 6.12: Comparison of the MSD convergence for the mesh sizes $7 \times 7 \times 7$, $8 \times 8 \times 8$, and $9 \times 9 \times 9$ using GOMEA with and without fold constraints—GOMEA with constraints is referred to as GOMEA-FC. No regularization or other constraints are used.

Mesh Size	Method	\vec{v}_ϵ	TRE	ASD_{cube}	ASD_{sphere}	E_b
$7 \times 7 \times 7$	GOMEA	0.77 ± 0.24	1.89 ± 0.59	0.42 ± 0.02	0.28 ± 0.01	0.57 ± 0.27
	GOMEA-FC	1.24 ± 0.04	2.80 ± 0.07	0.40 ± 0.01	0.44 ± 0.04	0.38 ± 0.04
$8 \times 8 \times 8$	GOMEA	0.84 ± 0.04	1.96 ± 0.08	0.58 ± 0.02	0.28 ± 0.01	0.80 ± 0.05
	GOMEA-FC	0.34 ± 0.06	0.80 ± 0.12	0.40 ± 0.00	0.29 ± 0.00	0.12 ± 0.02
$9 \times 9 \times 9$	GOMEA	1.58 ± 0.07	4.23 ± 0.23	0.59 ± 0.02	0.36 ± 0.02	3.12 ± 0.16
	GOMEA-FC	0.82 ± 0.03	1.49 ± 0.08	0.42 ± 0.01	0.28 ± 0.01	0.47 ± 0.04

Table 6.7: Comparison of validation results for the mesh sizes $7 \times 7 \times 7$, $8 \times 8 \times 8$, and $9 \times 9 \times 9$ for the synthetic registration problem between GOMEA and GOMEA-FC. No regularization or missed voxel constraints are used.

Results Figure 6.12 shows how the constraints impact the optimization of the similarity metric. We see that GOMEA-FC converges to solutions with significantly worse MSD values. This is expected since the constraints heavily restrict the search space by not allowing adjacent control points to overlap. We can say that the search space with constraints is a strict subset of the space without constraints, which means that GOMEA-FC can never outperform GOMEA in terms of the similarity metric. However, it can outperform in terms of transformation quality. This becomes clear from the validation results in Table 6.7, which show that GOMEA-FC outperforms

for the mesh sizes $8 \times 8 \times 8$ and $9 \times 9 \times 9$. The smaller bending energies show that the constraints inherently regularize the transformation by keeping the control points from overlapping. It limits the possible transformation complexity but does enforce more natural transformations with improved results for the ground truth error, target registration error, and average surface distances. The underperformance for the smallest mesh size is likely due to the larger spacing between the control points. Increasing the mesh size decreases the spacing between control points and the region of voxels they influence. This increased freedom left unconstrained leads to worse outcomes for the validation metrics, likely due to overfitting. If we constrain the optimizer for these larger mesh sizes, it can find transformations of higher quality that adequately transform the source using lower bending energy.

Figure 6.13 shows a visual comparison between two runs using either GOMEA or GOMEA-FC with a mesh size of $8 \times 8 \times 8$. When constrained, the optimizer finds solutions with more naturally aligned control points similar to configurations found by ASGD. The Jacobian determinant and DVF are more regular, containing less abrupt and sharp changes with significantly smaller magnitudes. Moreover, the negative values in the Jacobian determinant for GOMEA indicate that folds exist in the transformation. However, these mainly lie around the edges of the transformation and do not impact the cube or sphere. Still, these results confirm that constraining the control points eliminates the possibility of folds in the transformation. Whenever we know this to be a hard requirement, we can restrict the search space by imposing overlap constraints.

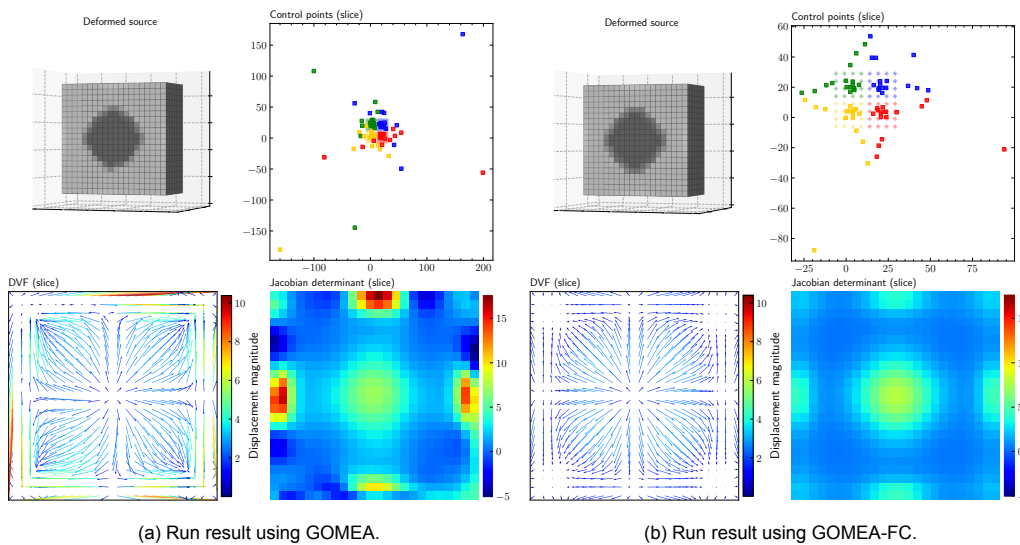


Figure 6.13: Visual comparison of two runs using either GOMEA or GOMEA-FC to optimize the synthetic registration problem with no regularization or missed voxel constraints/penalty and a mesh size of $8 \times 8 \times 8$.

6.2.6. Hybridization using Gradient-based Local Search

From our experiments thus far, we can argue for either GOMEA or ASGD to optimize registration problems. ASGD is fast, naturally provides more regular results, and is less prone to overfitting. GOMEA, on the other hand, is objectively better at optimizing the similarity metric but is much slower in its operation. If left unconstrained, GOMEA will likely find physically implausible transformations with high bending energy, possibly including folds and sharp changes. Therefore, it makes sense to combine both methods to leverage the global exploration of EAs, such as GOMEA, with the efficient local exploitation of a gradient-based method, such as ASGD. We can exploit each basin of interest more efficiently across the search space by optimizing solutions using a local gradient-based search method. By maintaining a population of solutions across the search space, we decrease the likelihood of converging to local optima while improving the convergence rate by requiring fewer generations to iteratively improve the solutions towards the "bottom" of each basin. We have implemented a basic hybridization in Elastix of GOMEA with local search (GOMEA-LS) with two redistribution methods and two iteration schedules (see Section 5.3.6).

Setup We estimate the effect these parameters have on registration outcomes by conducting a standard grid search for $\tau_g \in \{0.0, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3\}$ and $i_g \in \{5, 10, 20, 30, 50, 100, 200, 500\}$ when a mesh size of

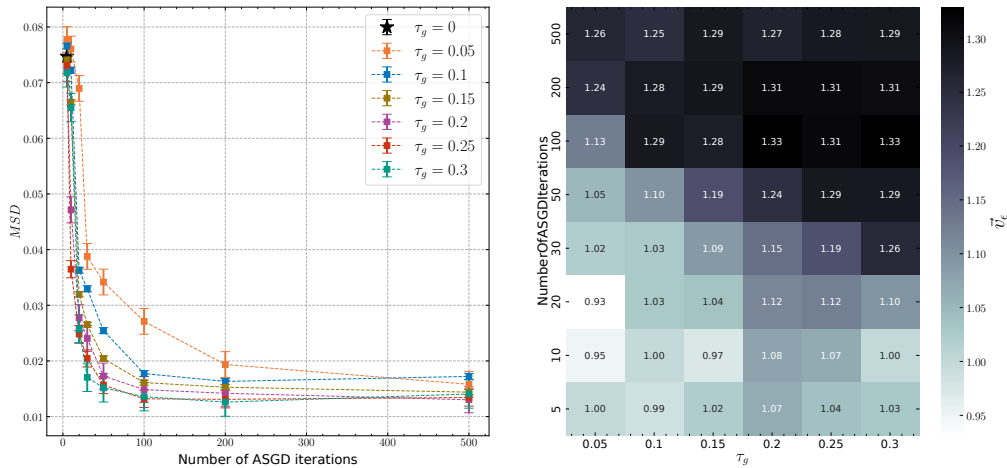


Figure 6.14: MSD plotted against the number of ASGD iterations for each τ_g (left) and a heatmap with the average ground truth error \vec{v}_ϵ for each combination of parameter values (right). For reference, the result found by GOMEA ($\tau_g = 0$) has an average \vec{v}_ϵ of 1.02.

8x8x8 and a budget of $5e10$ pixel evaluations are used for the synthetic registration problem. We use the static iteration schedule and random redistribution method—i.e., gradient descent is applied to a random subset of solutions for a fixed number of iterations at the end of each generation.

Additionally, we compare GOMEA and GOMEA-LS for the mesh sizes 6x6x6, 7x7x7, and 8x8x8 in terms of the ground truth error \vec{v}_ϵ , TRE , and bending energy E_b . Registrations are performed without constraints or regularization and are repeated ten times with different seeds. For GOMEA-LS, we use $\tau_g = 0.1$ and $i_g = 20$. For the calculation of the \vec{v}_ϵ and E_b , we use a mask to only consider the region of the ground truth deformation vector field (DVF) with non-zero entries. We ignore the outliers around the edges of the transformation as these have minimal to no impact on the transformation of the cube or sphere itself.

Results Figure 6.14 shows a negative correlation between the number of ASGD iterations i_g and the objective value. With increased iterations, the optimizer finds lower values for the MSD by aggressively exploiting the local search method. The heatmap shows that these improved results are likely overfitting since they are worse regarding the ground truth error. We must remember that no regularization was used, so these results do not necessarily suggest that using more iterations at the end of each generation is not a good idea. It only shows that it increases the likelihood of optimizing the similarity metric more effectively. More interestingly, the heatmap shows an area of improved \vec{v}_ϵ values around the center when a moderate amount of ASGD iterations is applied to a small portion of the population at the end of each generation.

Mesh Size	Method	\vec{v}_ϵ	TRE	E_b
6x6x6	GOMEA	0.34 ± 0.05	0.73 ± 0.10	0.08 ± 0.01
	GOMEA-LS	0.19 ± 0.01	0.51 ± 0.02	0.06 ± 0.00
7x7x7	GOMEA	0.85 ± 0.14	2.10 ± 0.27	0.62 ± 0.17
	GOMEA-LS	0.31 ± 0.12	0.67 ± 0.29	0.08 ± 0.02
8x8x8	GOMEA	0.81 ± 0.04	1.97 ± 0.14	0.76 ± 0.06
	GOMEA-LS	0.71 ± 0.09	1.46 ± 0.15	0.64 ± 0.15

Table 6.8: Comparison of validation results for the mesh sizes 6x6x6, 7x7x7, and 8x8x8 for the synthetic registration problem between GOMEA and GOMEA-LS ($\tau_g = 0.1$, $i_g = 20$). No regularization or constraints are used.

Table 6.8 shows that hybridization positively impacts the validation metrics with significant improvements for all mesh sizes. The difference is most notable when using a mesh size of 7x7x7. A visual comparison of run results between the two methods for this mesh size is shown in Figure 6.15. From it, we see from both

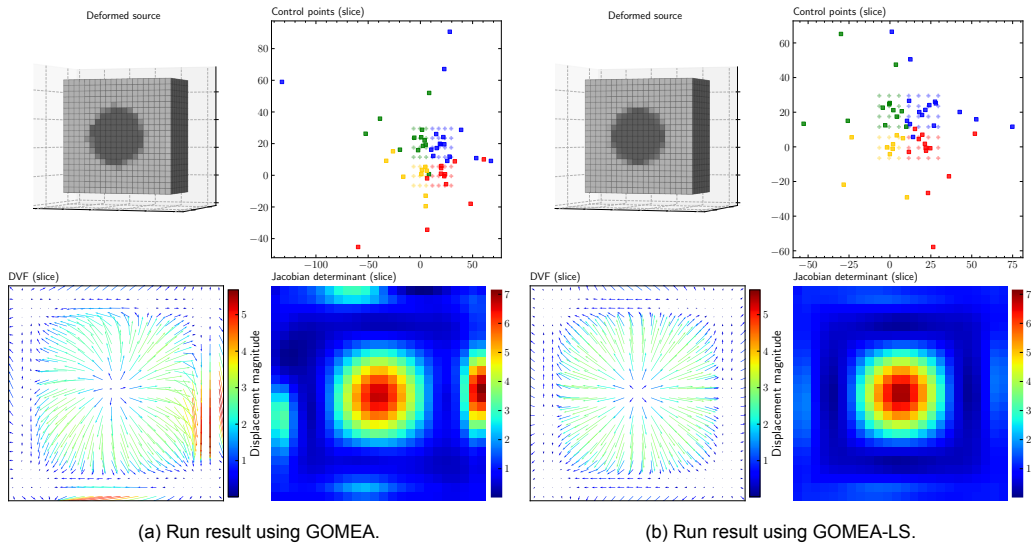


Figure 6.15: Visual comparison of two runs when using either GOMEA or GOMEA-LS to optimize the synthetic registration problem with no regularization or constraints and a mesh size of $7 \times 7 \times 7$.

the DVF and Jacobian determinant slices that the transformation found by GOMEA-LS has more regularity and smoothness.

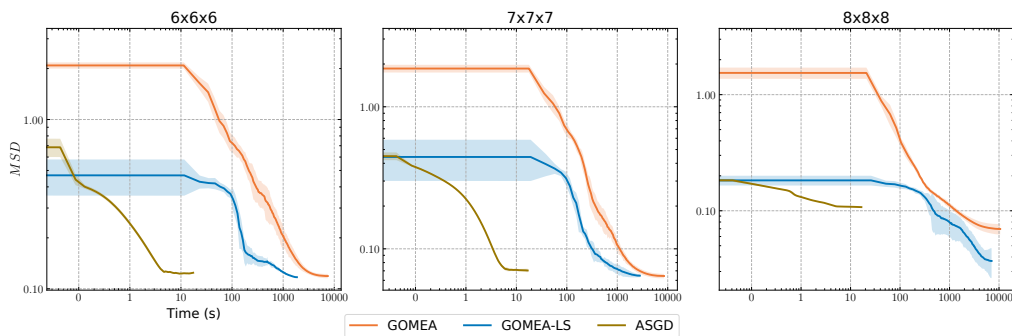


Figure 6.16: Comparison of the convergence rates for the mesh sizes of $6 \times 6 \times 6$, $7 \times 7 \times 7$, and $8 \times 8 \times 8$ between GOMEA and GOMEA-LS ($\tau_g = 0.1$, $i_g = 20$) measured by the MSD over time in seconds.

Lastly, we show the impact on the convergence rate by visualizing the metric value as a function of time for each mesh size. Figure 6.16 shows that the hybrid method achieves quicker convergence rates for each mesh size. We can conclude that the hybrid method should be considered when solving other registration problems. It can find solutions with lower objective values that are also better regarding the validation metrics and is quicker to converge. The recommended parameter settings will depend on the registration problem, but we recommend starting with small values for τ_g and i_g , e.g., $\tau_g \leq 0.1$ and $i_g \leq 50$; setting these too aggressively might lead to premature convergence or overfitting.

6.2.7. Regularization Weight

Regularization is used in deformable image registration to prevent overfitting and to ensure the transformation between the two images is smooth and physically plausible. The bending energy is a common regularization term that is used to penalize large deformations and to encourage the transformation to be smooth and continuous. This is particularly important when working with medical images, as large, unrealistic deformations can lead to inaccurate results and severe consequences in clinical applications. In Elastix, the bending energy can be included in a weighted combination with any similarity metric—weights for each term will have to be specified. As such, we still have a single-objective optimization problem, but now with an infinite number of possible weight combinations to choose from, which only adds to the complexity of these problems.

By increasing the weight for the bending energy term, we limit the space of possible transformations by

penalizing those which include large deformations. We have already seen that gradient-based optimizers are more prone to premature convergence in local optima due to their exploitative approach. It would be interesting to see how regularization affects this phenomenon, as it might decrease the number of local optima through its restrictive effect on the search space. Oppositely, the bending energy could also increase the number of local optima due to its added complexity in the objective function. GOMEA has shown to be more capable at avoiding local optima, which might mean it is also less sensitive to the chosen regularization weight. A known issue is that it is hard to a priori estimate the weights that adequately solve the registration problem. If the optimizer can find these solutions given a broader range of weights, it becomes less of a guessing game for practitioners.

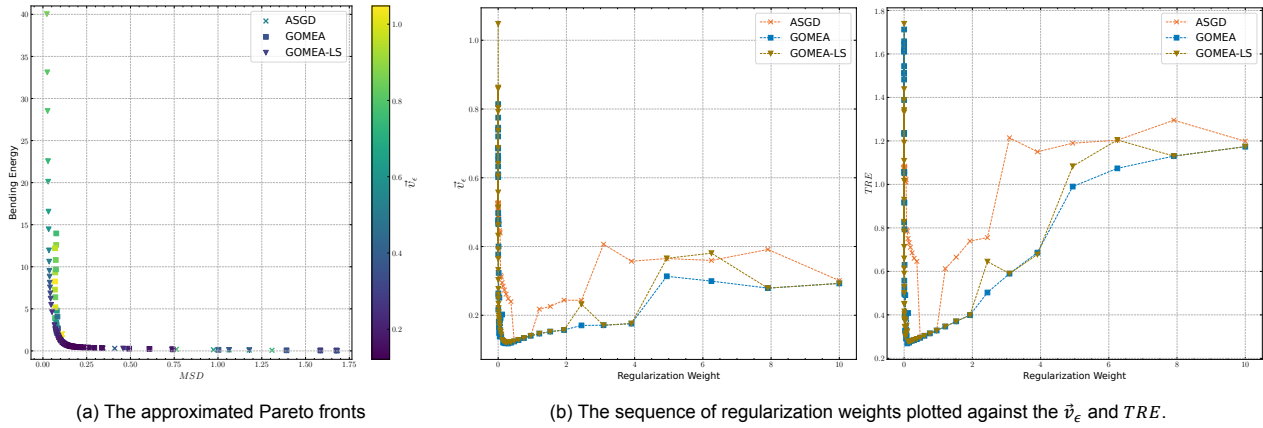


Figure 6.17: Approximated Pareto fronts and validation results when using either GOMEA, GOMEA-LS, or ASGD for the synthetic registration problem using a mesh size of $8 \times 8 \times 8$. To approximate the fronts, we optimize the problem using a range of regularization weights for the bending energy. The color scheme shows the ground truth error \bar{v}_ϵ achieved by each run.

Setup We use a geometrically increasing sequence of weights $\in [0.0, 10.0]$ to measure the impact of regularization on registration outcomes for the synthetic registration problem. Both optimizers are given the same sequence, and a mesh size of $8 \times 8 \times 8$ is used. We use a weighted combination of the analytic bending energy and the MSD .

Results Figure 6.17 visualizes the approximated Pareto fronts for both optimizers. From this figure, we can see that GOMEA’s results span a much broader range of the bending energy than ASGD. If we plot the fronts jointly, we can see that they look very similar. The most significant difference comes from the left side of the front, where GOMEA achieves smaller values for the MSD but with decreasingly smooth transformations and higher ground truth errors. These transformations are likely overfitting to the registration problem; however, they show that GOMEA is more capable as an optimizer as it finds a range of solutions that dominates ASGD.

Due to the overlap of the approximated fronts, it does not immediately become clear what optimizer outperforms in terms of transformation quality. Figure 6.17 shows more precisely what both optimizers achieved regarding the ground truth error and TRE . We see that GOMEA dominates the results of ASGD for both metrics. To achieve comparably good results with ASGD, we need to set the regularization weight to a rather small range of values, whereas, with GOMEA, we can permit a much broader range of weights. Consequently, if we have no prior knowledge of the registration problem and what weights might be suitable, it would be better to use GOMEA to optimize it using a series of weights since it is more likely we will find a good result. However, ASGD operates magnitudes faster than GOMEA; in settings where time is a constraint, it might still make more sense to use ASGD.

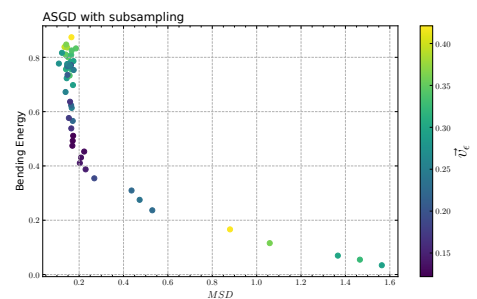


Figure 6.18: An approximation to the Pareto front when sub-sampling is used when optimizing the synthetic registration problem with ASGD. We use the same sequence of weights as in Figure 6.17.

Lastly, we note that the approximated Pareto fronts are somewhat atypical in their shape, as they come close

to looking like a smooth curve; we expect more variance in the range of values towards the top of the front, where registrations are least constrained by the penalty term. The synthetic registration problem is possibly simple enough so that both optimizers can converge to near-optimal solutions across the front for each combination of weights. Otherwise, it could also be explained by the fact that we do not use sub-sampling but instead include all of the image voxels when calculating the objective function. If we use sub-sampling, we get a front as shown in Figure 6.18. With sub-sampling, a new set of voxels is chosen for each iteration. The set of voxels for the last iteration will determine the final output for the bending energy and MSD . As such, if we use a different seed for each chosen regularization weight, we will end up with a different set of voxels at the end of each run, leading to increased variance in the results.

6.2.8. LEARN

The synthetic registration problem provides an easy-to-understand and computationally tractable registration problem. Moreover, by creating it ourselves, we can specify landmarks, surface points, and a ground truth DVF, allowing us to validate the results quantitatively. Still, this thesis aims to find out if the application of EAs in DIR is feasible in practice. That is, the application of GOMEA—with prior modifications, e.g., fold constraints or hybrid local search—to real-world clinical registration problems. To this end, we compare results between variants of GOMEA and ASGD when applied to three cases of the LEARN challenge. In this challenge, pairs of CT scans for each patient are taken for both maximally inspired and expired lungs. By registering these pairs of images, the large lung motion can be estimated and used by practitioners for diagnosis (e.g., COPD, see Section 1).

Setup We register cases 1-3 of the LEARN challenge using a mesh size of $8 \times 8 \times 8$ with and without regularization. To optimize the registration problems, we use a weighted combination of the Zero-Normalized Correlation Coefficient (ZNCC) (see Section 4.1.3) with the bending energy. Due to the increased computational load, we use a small set of regularization weights $\{0.0001, 0.001, 0.01, 0.1\}$; initial sample runs indicate that these weights are sufficient to provide a diverse set of results across all three cases. A mask of the expired lungs is used to limit the calculation of the similarity of only those voxels between the fixed and deformed moving image. We compare the following methods: ASGD, GOMEA, GOMEA-LS, GOMEA-FC, and use an evaluation budget of $3e11$ voxels for each run. A multi-resolution scheme of three levels with down-sampling and smoothing is used, with the final level at half the resolution of the original image and no smoothing. We repeat each run ten times using different seeds. For GOMEA-LS, we use the static iteration schedule and random redistribution method.

To validate the results, provided landmarks are used to calculate the TRE . We also calculate the DSC between the transformed moving image mask and the fixed image mask, indicating how much the lungs overlap after registration. To measure the regularity of the transformation, we calculate the bending energy BE . Finally, to measure the diversity and quality of the approximated Pareto fronts, we calculate the dominated Hypervolume (HV), as defined in Section 3.3.2.

Results Figure 6.19 shows approximated Pareto fronts for each patient and method used. Runs per regularization weight and method are aggregated by taking the average of the $ZNCC$, BE , and TRE . Across all patients, we see that the solutions found by GOMEA-LS are both non-dominated and diverse, likely due to the explorative capabilities of GOMEA in combination with the exploitative capabilities of ASGD. We see that GOMEA finds similar fronts but with worse values for the $ZNCC$, which reaffirms that the combined approach of GOMEA-LS has merit. Solutions found by ASGD and GOMEA-FC are quite concentrated but are more regular and with lower values for the BE . For GOMEA-FC, this is expected, as the additional constraints restrict the search space by not allowing the control points to overlap; the additional regularization from the penalty term probably has minimal impact due to the already strong regularization effects from the constraints. For ASGD, it seems that even with a multi-resolution approach, it converges to solutions with significantly lower similarity scores than GOMEA-LS.

Not taking into account TRE and DSC , we can evaluate the performance of these methods by trying to understand the process from a practitioner's perspective. For any deformable image registration problem, it is often not known *a priori* what kind of registration is required. In some cases, a significant degree of deformation is required, while in others, small, subtle deformations are sufficient. As such, the problem is inherently multi-objective and should be considered as such when evaluating the performance of tested methods. The goal in multi-objective optimization is to find sets of solutions that approximate the Pareto set, a set of solutions with objective values both well-spread along and close to the Pareto front. Given such a set, a practitioner

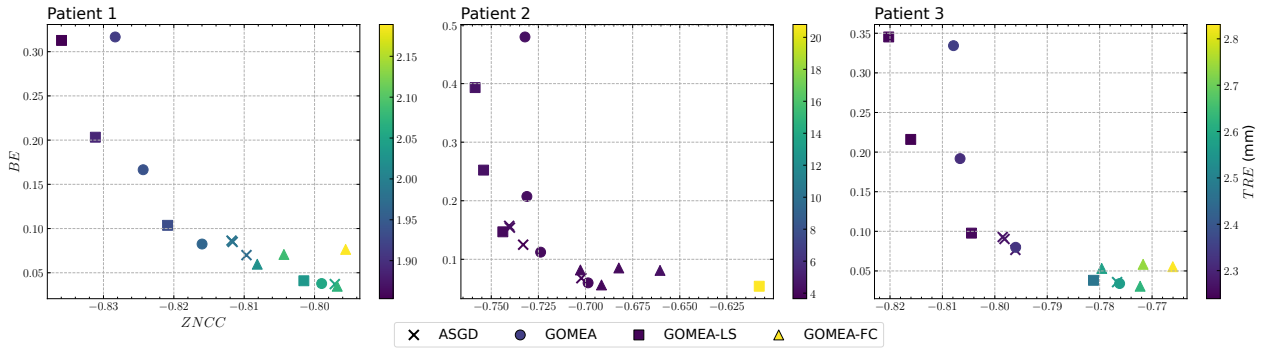


Figure 6.19: Approximated Pareto fronts for patients 1-3 of the LEARN dataset by ASGD, GOMEA, GOMEA-LS, and GOMEA-FC when using a mesh size of $8 \times 8 \times 8$, a 3-level multi-resolution scheme and regularization weights $\in \{0.0001, 0.001, 0.01, 0.1\}$. We plot the $ZNCC$ on the horizontal axes and the BE on the vertical axes. The TRE in mm is embedded as the color of each point.

Patient	Method	TRE	DSC	BE	HV
Patient 1	ASGD	1.97 ± 0.01	0.94 ± 0.00	0.07 ± 0.00	0.83
	GOMEA	1.94 ± 0.02	0.94 ± 0.00	0.17 ± 0.01	0.88
	GOMEA-LS	1.89 ± 0.01	0.93 ± 0.00	0.20 ± 0.01	0.89
	GOMEA-FC	2.02 ± 0.03	0.94 ± 0.00	0.06 ± 0.00	0.83
Patient 2	ASGD	3.65 ± 0.01	0.91 ± 0.00	0.12 ± 0.00	1.71
	GOMEA	4.00 ± 0.13	0.90 ± 0.00	0.06 ± 0.00	1.71
	GOMEA-LS	4.11 ± 0.04	0.90 ± 0.00	0.15 ± 0.00	1.75
	GOMEA-FC	4.03 ± 0.12	0.90 ± 0.00	0.06 ± 0.00	1.55
Patient 3	ASGD	2.27 ± 0.00	0.94 ± 0.00	0.09 ± 0.00	1.17
	GOMEA	2.33 ± 0.04	0.94 ± 0.00	0.08 ± 0.00	1.21
	GOMEA-LS	2.25 ± 0.00	0.94 ± 0.00	0.10 ± 0.00	1.24
	GOMEA-FC	2.58 ± 0.07	0.94 ± 0.00	0.05 ± 0.00	1.1

Table 6.9: Validation metrics for patients 1-3 of the LEARN [43] dataset when ASGD, GOMEA, GOMEA-LS, and GOMEA-FC is used with a mesh size of $8 \times 8 \times 8$, a multi-resolution scheme and regularization weights $\in \{0.0001, 0.001, 0.01, 0.1\}$. For each method, we only show the average values for the weight which provided the best results (except for the HV). A bold value indicates a (joint) statistically significant improvement over all other methods for a specific metric and patient. The HV s are scaled up for improved readability.

can decide on their preference regarding the trade-off between the objectives. Qualitatively, we can see from Figure 6.19 that GOMEA-LS approximates sets of solutions that are most diverse and with the highest $ZNCC$. Quantitatively, the quality of approximated sets can be measured using the HV , which refers to the total volume dominated in objective space by the solutions in the set w.r.t. to a reference point. The HV for each method and patient is shown in Table 6.9.

In Table 6.9, we also compare the best results for each method by inversely scaling the TRE by the DSC (i.e., $\frac{TRE}{DSC}$). In terms of the TRE , we see comparable results for all methods, with ASGD outperforming slightly for patient 2 and GOMEA-LS for patients 1 and 3. GOMEA-FC finds the most regular transformations with the lowest bending energies. Figure 6.20 shows the inversely scaled TRE distribution for all patients, methods, and regularization weights. Most interestingly, we see a significant outlier for patient 2 and GOMEA-LS when the largest regularization weight is used. From Figure 6.19, it becomes clear that this is due to GOMEA-LS converging on solutions with the lowest bending energy but with significantly lower $ZNCC$. This is not necessarily undesired behavior; it only further shows that GOMEA-LS is better at optimizing the objective function and provides sets of solutions with a broader range of trade-offs.

The bending energy measures the regularity of transformations as the sum of second-order spatial derivatives. For patient 1, we see that GOMEA and GOMEA-LS find solutions with competitive TRE and DSC scores but with transformations that contain considerably higher bending energies (by a factor of ≈ 3). To understand

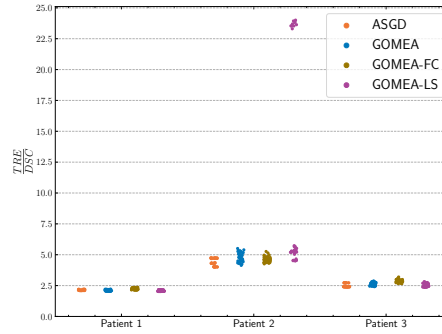
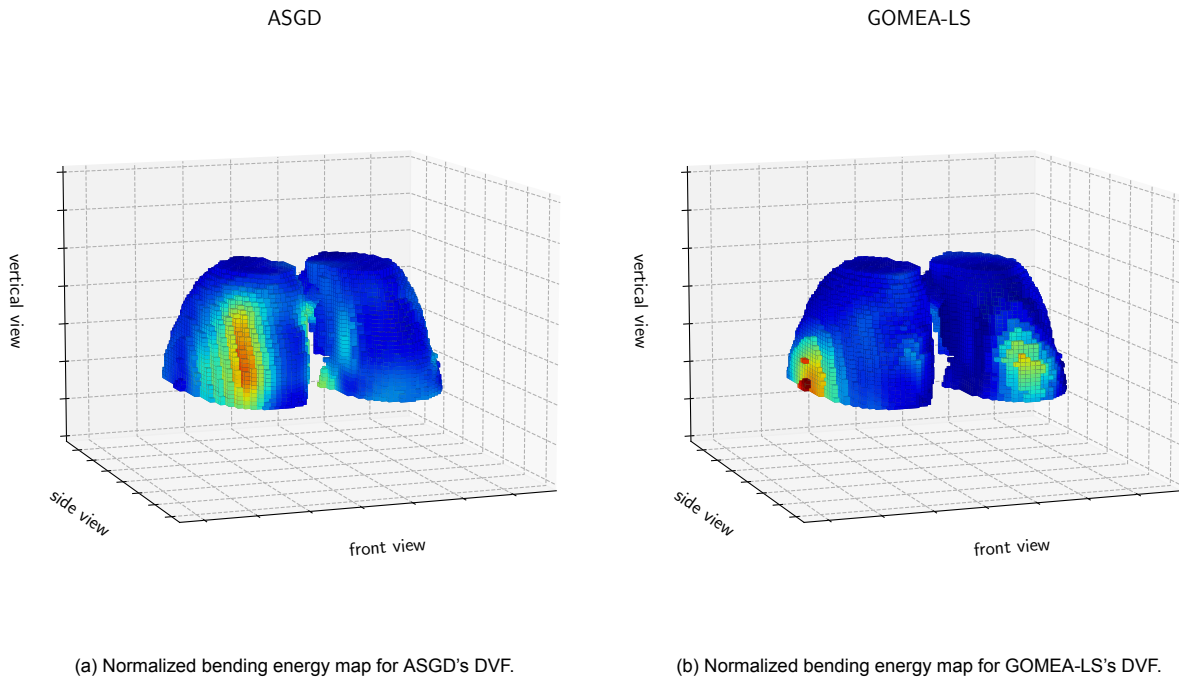


Figure 6.20: Distribution of the TRE inversely scaled by the DSC for each patient, method and used regularization weight.



(a) Normalized bending energy map for ASGD's DVF.

(b) Normalized bending energy map for GOMEA-LS's DVF.

Figure 6.21: Normalized bending energy maps for both ASGD and GOMEA-LS for registrations of Patient 1 when using a regularization weight of 0.001. Only voxels within the lung mask of the fixed image are shown.

the differences, we calculate the bending energy per voxel using the DVF and normalize by dividing it with the maximum bending energy across all voxels within the lung mask. Figure 6.21 compares these normalized energies mapped onto the voxels within the lungs for both ASGD and GOMEA when a regularization weight of 0.001 is used. From it, we see several areas in which the intensities of the bending energy differ. For example, GOMEA-LS has a concentration of high bending energies at the left lower side of the lung. In comparison, ASGD has a concentration of higher bending energies at a similar slice of voxels but a bit more inward towards the center of the body. Overall, the distribution of bending energies seems smoother for ASGD with gradual increments up until extrema. On the other hand, the mapped bending energies of GOMEA-LS are more concentrated, as we see two prominent regions with high bending energies and most other voxels being almost the same shade of dark blue. We have seen this behavior for ASGD and GOMEA across all experiments thus far. The transformations found by ASGD are generally smoother and have fewer sudden changes. In contrast, GOMEA can find more specific transformations that are better fit to the similarity metric but often of higher bending energy.

We compare the registration results in more detail by visualizing the DVF and re-sampled voxels for two slices with the most significant differences in bending energies between ASGD and GOMEA-LS. First, a horizontal slice taken at the lowest part of the lung is compared in Figure 6.22. The top row visualizes the difference in the DVFs using deformed uniform meshes, whereas the bottom row visualizes the difference in the regis-

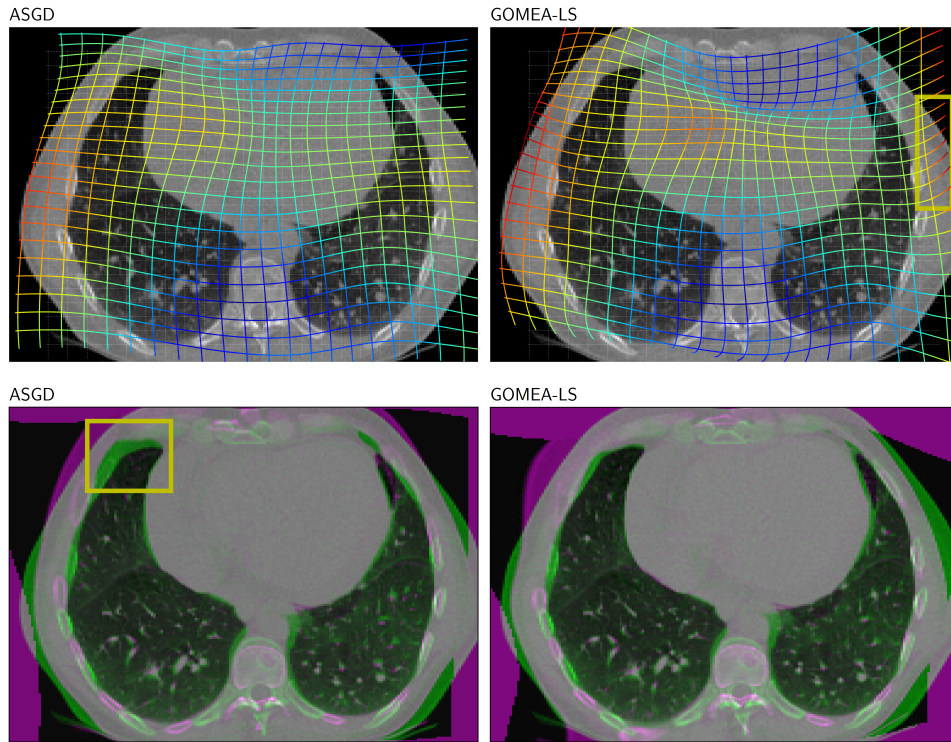


Figure 6.22: Deformed uniform meshes and color difference plots for two registration results by ASGD and GOMEA-LS for patient 1 of the LEARN dataset. The highlighted yellow rectangles indicate areas of interest. In the color difference plots, the green color displays the fixed image, and the purple color the deformed moving image. Only the voxels within the fixed image mask are used during registration. However, after registration, the moving image is re-sampled using all voxels, which can lead to inaccurate results outside of the mask.

tration outcome using a color difference plot between the deformed and fixed images. Important to note: only the voxels within the fixed image mask are used during registration, but after registration, the moving image is re-sampled using all voxels, which leads to erroneous results outside of the mask in the re-sampled images.

The deformed meshes show that the transformation of ASGD is more smooth and more regular, whereas the mesh of GOMEA-LS contains more local variability and overall seems more flexible. The highlighted area in the top-right corner contains the voxels with the large bending energies found for GOMEA-LS in Figure 6.21. The deformation does show a sudden curve upward; however, the bending energy is calculated in all three dimensions—the contribution in the vertical direction is left out. In terms of the registration outcomes, we see relatively similar results. A notable difference is in the highlighted area (left), where the movement around the edges of the lungs is better estimated by GOMEA-LS.

In Figure 6.23, we compare the same registration results from ASGD and GOMEA-LS for a vertical slice as seen from the side of the lungs. In the bottom area of the deformed mesh, ASGD’s DVF contained considerable bending energies compared to the rest of the deformations according to the normalized map from Figure 6.21. In this area, we do see deformations with the most significant magnitude in this slice, but not necessarily a large amount of bending—again, some of this could be explained by the deformation in the horizontal direction. Also, ASGD generally finds transformations that are more smooth and more regular, as stated before. Similar to the previous slice, we see from the highlighted area in Figure 6.23 that GOMEA-LS more accurately estimates the range of motion around the edges of the lungs.

Overall, we show that GOMEA(-LS/FC) is competitive to ASGD on the tested clinical cases from the LEARN [43] dataset. For patients 1 and 3, GOMEA-LS achieved the best registration results in terms of the *DSC* and *TRE*. For patient 2, ASGD achieved the best result. However, given that the problem is inherently multi-objective, GOMEA-LS has outperformed overall, as it delivered the best results regarding the diversity and quality of the approximation set. In terms of computational efficiency, ASGD is magnitudes faster and capable of performing many registrations with different regularization weights when GOMEA performs a single registration, even though this does not mean it can achieve similar image similarities. Lastly, this experiment has been somewhat limited, as we only use a single mesh size and a reasonably small set of weights in the interest of run times. To validate and expand on our findings, more experiments with larger mesh sizes and finer sets of

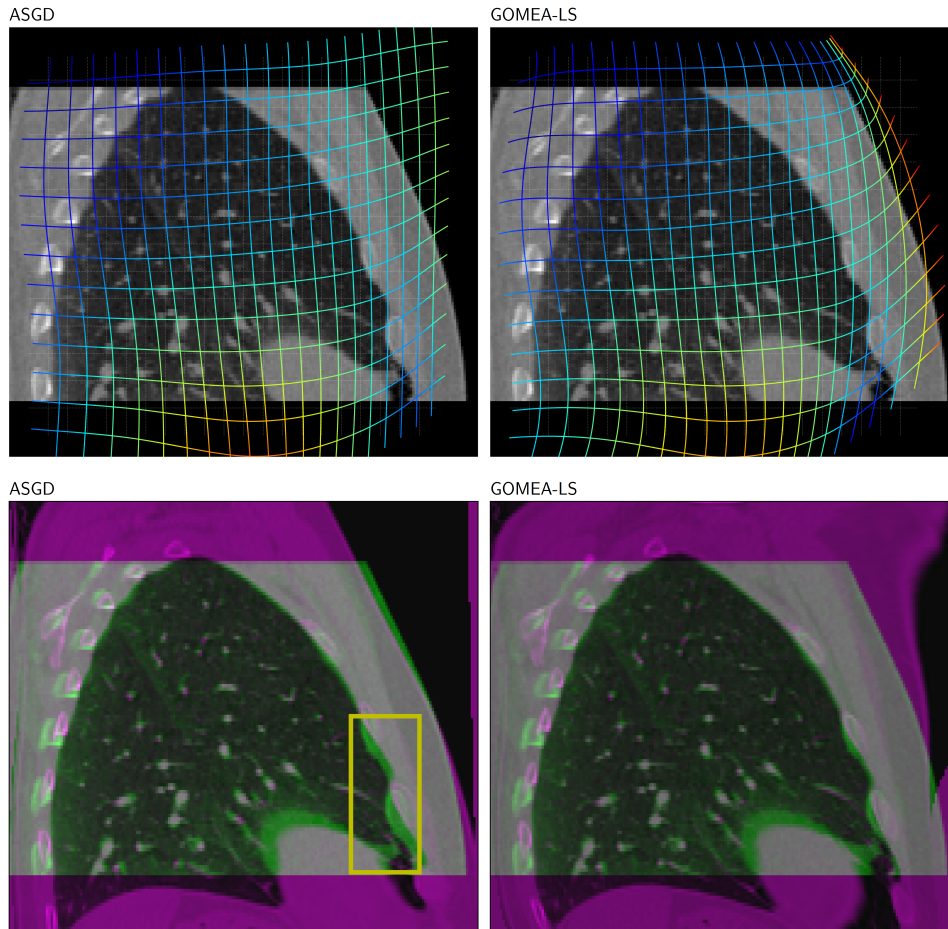


Figure 6.23: Deformed uniform meshes and color difference plots for two registration results by ASGD and GOMEA-LS for patient 1 of the LEARN dataset. The highlighted yellow rectangles indicate areas of interest. The magnitude of the deformation is visualized using a color scheme with the same scaling for both methods. In the difference plots, the green color displays the fixed image, and the purple color the deformed moving image.

weights are required.

6.3. Discussion

In this chapter, we analyze and evaluate the performance of GOMEA when applied to synthetic and clinical DIR problems. In our initial experiments on the synthetic registration problem, we find that GOMEA outperforms ASGD in the objective value but struggles to find solutions with equal transformation quality. Without a regularization term, GOMEA tends to overfit on the similarity metric, producing solutions that look good visually but with anatomically unlikely transformations. Multi-resolution schemes are often used in DIR to overcome local minima traps. When such a scheme is applied to the synthetic registration problem, we observe that GOMEA no longer outperforms ASGD in either the similarity metric or the quality of the transformation. However, when applied to more complex cases of the LEARN [43] dataset, we see that ASGD converges prematurely and cannot approximate Pareto fronts as well as GOMEA can. Overall, ASGD finds reasonable solutions to the tested registration problems, but quite specific tuning of the multi-resolution scheme and regularization weight is often necessary to achieve those solutions.

We use the synthetic registration problem to explore GOMEA's operation on registration problems when using a B-spline-based transformation model, hoping to understand better how it works and what improvements can be made. We test various linkage models (see Section 4.2) and find that the marginal and static models outperform in all aspects. Furthermore, the full linkage model can be considered when using a small mesh size, but for larger mesh sizes, it becomes computationally infeasible.

We also test various constraints to see if GOMEA can be guided towards more regular solutions without

using a regularization term. Both the missed voxel threshold and penalty hinder optimization by limiting the outer control points from expanding too far outside the area of the fixed image, resulting in worse outcomes in terms of similarity and transformation quality. Overlap constraints on the control points provide a benefit by inherently regularizing the transformation. In the synthetic registration problem, this leads to improved quality of transformations when the chosen mesh size is sufficiently large. However, when applied to clinical cases, the constraints are too limiting for the mesh size used. Further experiments with larger mesh sizes are necessary to determine if overlap constraints work well in practice.

Lastly, we combine ASGD and GOMEA by applying gradient descent to a subset of solutions at the end of each generation, resulting in improved registration results for both the synthetic and clinical registration problems. Our results suggest that the explorative population-based approach of GOMEA is a good fit with the exploitative gradient-based approach of ASGD.

To evaluate the efficacy of our implemented methods in practice, we run registrations on the first three clinical cases from the LEARN challenge. Preliminary findings indicate that GOMEA(-LS/FC) is capable of registration results competitive with ASGD in terms of the *TRE* and *DSC*, with improved results for patients 1 and 3. Moreover, we see that the sets of solutions found by GOMEA-LS are more diverse and of higher quality. The results suggest that using GOMEA-LS on clinical registration problems can provide significant benefits over ASGD when considering the problem is inherently multi-objective. However, due to the computational complexity of the current GOMEA implementation in Elastix, a moderately small mesh size of $8 \times 8 \times 8$ was used for the clinical cases. More experiments with larger mesh sizes and a finer set of regularization weights are required to validate our findings.

Conclusion and Future Work

In this thesis, we analyze the performance of hybrid evolutionary algorithms when applied to deformable image registration using common B-spline-based methods. To this end, we implement the Real-Valued Gene-pool Optimal Mixing Evolutionary Algorithm (RV-GOMEA) [20] and specific variants in Elastix [49], with support for the partial evaluation of common similarity metrics and penalty terms. The implemented methods are tested on both synthetic and clinical registration problems. In this chapter, we summarize our findings, reflect on the research topics, and give recommendations for future work.

7.1. Conclusion

Several contributions are made in this work. First, we provide an implementation of (RV-)GOMEA [20] within Elastix [49]. We show how partial evaluations can be applied to any objective function using a B-spline-based transformation model to drastically improve the computational efficiency of the registration process. For three commonly used similarity metrics and penalty terms, we provide formal derivations and implementations for partial evaluations during optimization. We also implement several versions of GOMEA with specific alterations or additions based on initial findings from experiments on a synthetic registration problem. Furthermore, we implement fold constraints so that control points from the B-spline grid cannot overlap in GOMEA-FC, to improve the regularity of the transformation without using a regularization term. In GOMEA-LS, we implement an operator that applies gradient-based local search to a subset of solutions at the end of each generation to combine the explorative capabilities of GOMEA with the exploitative qualities of gradient-based optimization. Lastly, we integrate an analytic implementation of the bending energy [75] to improve the existing implementation in Elastix, which computes the bending energy by estimating the spatial Hessian at each voxel in all dimensions.

To gain an understanding of Adaptive Stochastic Gradient Descent's (ASGD) and GOMEA's performance on different Deformable Image Registration (DIR) problems, we create a simple 3D synthetic registration problem in which a sphere within a cube shrinks. We add surface points, landmarks, and a ground truth DVF so that we can validate the transformation quality after each registration. We find that GOMEA is more capable of optimizing the similarity metric but often overfits if no regularization term is used. However, if a regularization term is used, we find that GOMEA-LS outperforms in the approximated Pareto front, but most notably in the transformation quality, as it dominates the results of ASGD for the majority of used regularization weights. Subsequently, we test several linkage models and find that both the marginal linkage model \mathcal{F}_{MP} and static linkage model \mathcal{F}_{SLT} outperform. Furthermore, in Elastix, voxels mapped outside of the moving image are omitted in the calculation of similarity metrics, which can lead to locally undefined transformations. To measure the effect of this formulation on registration quality, we implement a missed voxel threshold and penalty that can be used during registration to limit the amount of missed voxels per solution. We find that such a threshold or penalty can significantly inhibit the registration by too heavily restricting the outer control points from the B-spline model. In contrast, the constraints introduced in GOMEA-FC lead to significantly improved regularity of transformations; with a sufficiently large mesh size, it can outperform GOMEA in transformation quality. Lastly, applying gradient descent in GOMEA-LS significantly improves convergence rates, found objective function values, and transformation qualities when applied to the synthetic registration problem.

Given a better understanding of GOMEA's and ASGD's operation within the context of DIR, we validate

the implemented methods using clinical registration problems from the LEARN [43] dataset. We state that the DIR problem is inherently multi-objective, as it is often not known *a priori* how much deformation is required. Therefore, multiple registrations are done using a set of regularization weights to choose a suitable solution *a posteriori*. Viewed as such, GOMEA-LS outperforms on all three tested patients as it approximates solution sets that are close to and well spread across the Pareto front. Quantitatively, this is confirmed by the fact that for each patient, GOMEA-LS dominates the most Hypervolume in terms of the similarity metric and penalty term. We additionally compare the best solutions found by each method and find mixed results regarding the transformation quality. For patients 1 and 3, GOMEA-LS outperforms in the *TRE* and *DSC*. However, for patient 2, ASGD outperforms. From visualizations of the DVF and registration outcomes, we find that GOMEA-LS is capable of deforming local patches more flexibly, which could be the reason for its improved similarity scores. Overall, we conclude that GOMEA-LS provides significant benefits over ASGD when applied to the LEARN dataset.

In closing, the application of hybrid evolutionary algorithms is certainly to be considered in the context of DIR when using common B-spline-based transformation models. More specifically, the application of GOMEA, an evolutionary algorithm capable of exploiting the structure of optimization problems through explicit mappings of dependencies between problem variables. We show that with these mappings, partial evaluations can be applied to common similarity metrics and penalty terms, improving the registration’s computational efficiency. Moreover, we combine the population-based approach of GOMEA with the exploitative capabilities of ASGD in GOMEA-LS and find that it improves convergence rates and similarity scores. Additionally, given the conflicting trade-offs in DIR, we find that GOMEA-LS approximates sets of higher quality solutions better spread along the Pareto front compared to ASGD. Overall, GOMEA-LS has proven to be effective at optimizing the metrics of the DIR problem; consequently, its application should be done with caution to avoid overfitting to these optimization metrics. Moreover, in time-critical scenarios, GOMEA’s slower operation may negatively impact registration performance compared to the faster ASGD. To conclude, we see promising future possibilities for hybrid evolutionary algorithms in the context of DIR. However, more research is needed before they can be applied in critical clinical settings.

7.2. Future Work

While we hope this thesis provides valuable insights and clear interpretative results, we find many avenues for further research. This section broadly aggregates these areas and provides recommendations for future work.

Dynamic Resource Allocation In Section 6.2.6, we analyze the impact of the hybridization of GOMEA with gradient descent-based local search (GOMEA-LS) when applied to DIR problems. Due to time constraints, we implement this functionality with limited options regarding the iteration schedule and redistribution method. For example, suppose the static iteration schedule is used. In that case, gradient descent is applied to a fixed percentage of the population for a fixed number of iterations at the end of each generation. It would be more efficient if we determine a schedule dynamically based on the benefits of applying local search at any point during the registration. As such, we recommend investigating the application of a resource allocation scheme for the distribution of resources in GOMEA-LS. Recent work implements such a scheme to dynamically allocate resources between a multi-objective variant of GOMEA with UHV-Adam, a method capable of optimizing multi-objective problems using the gradient of the Uncrowded Hypervolume (UHV) (see Section 3.3.2), and provides analysis on possible reward functions and distribution schedules [36].

Constrained Differential Optimization We introduced GOMEA-FC, which uses overlap constraints on the B-spline control points to regularize the registration without using a penalty term. Due to the constraints, it is not straightforward to combine this approach with a gradient-based local search operator as implemented in GOMEA-LS. To include constraints in gradient-based optimization, we can use penalty-based approaches that replace the constraints in constrained optimization problems with various penalty terms. For example, a penalty term adding $\pm\infty$ to infeasible solutions. An exhaustive overview of various penalties is given in [88].

Similarly, Lagrange multipliers [9] can convert the constrained optimization problem into an unconstrained optimization problem [33]. The Basic Differential Multiplier Method [66] is one example of an approach that uses Lagrange multipliers to solve constrained optimization problems using gradient descent. Lastly, the Resultant Gradient Descent Algorithm (RGDA) [29] does not transform the constrained optimization problem but iteratively moves solutions out of infeasible regions using gradients w.r.t. the individual constraints. The Gradient-Based Genetic Algorithm (GGA) [29] uses RGDA in a hybrid evolutionary approach to solve constrained optimization

problems. It can be interesting to apply one of these methods within GOMEA-FC to determine if it improves the convergence rate and registration results for DIR problems.

Multi-objective Optimization In Section 2.4, we argue for applying multi-objective optimization for DIR problems due to the trade-off between image similarity and deformation magnitude. DIR problems are inherently multi-objective, but in this thesis, we implement and apply a single-objective optimizer and reformulate the multi-objective optimization problem by weighting the individual objectives. Most medical registration toolboxes—including Elastix—use this formulation as it allows for gradient-based optimization of the weighted objective function. As such, we first implement the single-objective optimizer RV-GOMEA in Elastix and evaluate its performance at solving DIR problems. A logical next step would be to implement the Multi-Objective Real-Valued Gene-pool Optimal Mixing Evolutionary Algorithm (MO-RV-GOMEA) [21] in Elastix as well. Moreover, similar to GOMEA-LS, an implementation, and hybridization with UHV-Adam [32] could provide MO-RV-GOMEA with an efficient local search operator that improves convergence rates towards the Pareto front.

Hardware Acceleration Our implementation of RV-GOMEA in Elastix is multi-threaded but solely runs on the CPU. For registration problems with large resolutions (e.g., LEARN) with decently sized B-spline grids (e.g., 8x8x8), individual runs can take up to 2 days to complete. In a practical sense, this is not ideal as it increases the feedback loop for researchers and practitioners. Recent work has drastically improved the run times of evolutionary optimization algorithms by utilizing the GPU and parallelization of independent subfunctions [22]. Such an approach can be used in Elastix to drastically improve the computational efficiency of RV-GOMEA when applied to DIR problems. For example, implementing a dual-dynamic triangulated grid shows speedup factors of up to 111 when applied to various 2D DIR problems [25]. Such speedups would improve the practicality of RV-GOMEA within Elastix by reducing run times significantly.

Further Linkage Models Analysis In Section 6.2.3, we analyze registration results for the synthetic registration problem for various linkage models. We find that linkage mainly exists between adjacent control points and that the marginal and static linkage models outperform. However, for other registration problems, we might find different outcomes due to a difference in linkage structure and magnitude. As such, we recommend a further analysis of linkage models when they are used to solve more complex and real-world registration problems (e.g., the LEARN dataset).

Concatenation of Transformations In Elastix, it is possible to specify a grid schedule for the B-spline parameters in a multi-resolution setting. A smaller grid size is often used for the first resolutions to capture the large deformations and to avoid local minima traps. The parameters from the previous resolution are up-sampled using a decomposition of the parameterized deformation field. This decomposition can lead to inaccuracies and possibly introduces overlaps in the control point configuration of the up-sampled grid. An alternative method is possible in which subsequent B-spline transformations are concatenated—i.e., at a specific resolution, the deformation is given by concatenating the previous deformations with the current deformation.

Interestingly, a study found that combining such a concatenated transformation and specific constraints on the control point displacements can lead to diffeomorphic—meaning smooth and invertible—registrations [73]. Ensuring diffeomorphism can improve the regularity and physical plausibility of the transformation, which is desirable if it is known that similar structures exist in both images, and no large deformations are required. Concatenating B-spline deformations only introduces new possibilities and simultaneously removes the possibly inaccurate up-sampling step from the registration.

Bibliography

- [1] Tanja Alderliesten, Peter A. N. Bosman, and Wiro J. Niessen. Towards a Real-Time Minimally-Invasive Vascular Intervention Simulation System. *IEEE Transactions on Medical Imaging*, 26(1):128–132, January 2007. ISSN 0278-0062. doi: 10.1109/TMI.2006.886814.
- [2] Tanja Alderliesten, Jan-Jakob Sonke, and Peter A. N. Bosman. Multi-objective optimization for deformable image registration: Proof of concept. In *Medical Imaging 2012: Image Processing*, volume 8314, pages 594–600. SPIE, February 2012. doi: 10.1117/12.911268.
- [3] Tanja Alderliesten, Jan-Jakob Sonke, and Peter A. N. Bosman. Deformable image registration by multi-objective optimization using a dual-dynamic transformation model to account for large anatomical differences. In Sebastien Ourselin and David R. Haynor, editors, *SPIE Medical Imaging*, page 866910, Lake Buena Vista (Orlando Area), Florida, USA, March 2013. doi: 10.1117/12.2006783.
- [4] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen. *LAPACK Users' Guide*. Society for Industrial and Applied Mathematics, Philadelphia, PA, third edition, 1999. ISBN 0-89871-447-8 (paperback).
- [5] Georgios Andreadis, Peter A. N. Bosman, and Tanja Alderliesten. Multi-objective dual simplex-mesh based deformable image registration for 3D medical images - proof of concept. In *Medical Imaging 2022: Image Processing*, volume 12032, pages 744–750. SPIE, April 2022. doi: 10.1117/12.2605498.
- [6] Aliasghar Arab and Alireza Alfi. An adaptive gradient descent-based local search in memetic algorithm applied to optimal controller design. *Information Sciences*, 299:117–142, April 2015. ISSN 0020-0255. doi: 10.1016/j.ins.2014.11.051.
- [7] Mordecai Avriel. *Nonlinear Programming: Analysis and Methods*. Courier Corporation, January 2003. ISBN 978-0-486-43227-4.
- [8] Guha Balakrishnan, Amy Zhao, Mert R. Sabuncu, John Guttag, and Adrian V. Dalca. VoxelMorph: A Learning Framework for Deformable Medical Image Registration. *IEEE Transactions on Medical Imaging*, 38(8):1788–1800, August 2019. ISSN 1558-254X. doi: 10.1109/TMI.2019.2897538.
- [9] Brian Beavis and Ian Dobbs. *Optimisation and Stability Theory for Economic Analysis*. Cambridge University Press, February 1990. ISBN 978-0-521-33605-5.
- [10] Luiza Bondar, Mischa S. Hoogeman, Eliana M. Vásquez Osorio, and Ben J. M. Heijmen. A symmetric non-rigid registration method to handle large organ deformations in cervical cancer patients. *Medical Physics*, 37(7):3760–3772, July 2010. ISSN 0094-2405. doi: 10.1118/1.3443436.
- [11] C. D. Boor. B(asic)-Spline Basics. *WISCONSIN UNIV-MADISON MATHEMATICS RESEARCH CENTER*, 1986.
- [12] De Boor and C. SUBROUTINE PACKAGE FOR CALCULATING WITH B-SPLINES. Technical Report LA-4728, Los Alamos National Lab. (LANL), Los Alamos, NM (United States), January 1971.
- [13] Peter A. N. Bosman. On Gradients and Hybrid Evolutionary Algorithms for Real-Valued Multiobjective Optimization. *IEEE Transactions on Evolutionary Computation*, 16(1):51–69, February 2012. ISSN 1941-0026. doi: 10.1109/TEVC.2010.2051445.
- [14] Peter A N Bosman and Dirk Thierens. Exploiting Gradient Information in Continuous Iterated Density Estimation Evolutionary Algorithms. *BNAIC'01 Sponsors*, page 8, 2001.
- [15] Peter A. N. Bosman, Jörn Grahl, and Dirk Thierens. Benchmarking Parameter-Free AMaLGaM on Functions With and Without Noise. *Evolutionary Computation*, 21(3):445–469, September 2013. ISSN 1063-6560, 1530-9304. doi: 10.1162/EVCO_a_00094.

- [16] Peter A.N. Bosman. The anticipated mean shift and cluster registration in mixture-based EDAs for multi-objective optimization. In *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation*, GECCO '10, pages 351–358, New York, NY, USA, July 2010. Association for Computing Machinery. ISBN 978-1-4503-0072-8. doi: 10.1145/1830483.1830549.
- [17] Peter A.N. Bosman and Tanja Alderliesten. Incremental gaussian model-building in multi-objective EDAs with an application to deformable image registration. In *Proceedings of the Fourteenth International Conference on Genetic and Evolutionary Computation Conference - GECCO '12*, page 241, Philadelphia, Pennsylvania, USA, 2012. ACM Press. ISBN 978-1-4503-1177-9. doi: 10.1145/2330163.2330199.
- [18] Peter A.N. Bosman and Dirk Thierens. Linkage neighbors, optimal mixing and forced improvements in genetic algorithms. In *Proceedings of the 14th Annual Conference on Genetic and Evolutionary Computation*, GECCO '12, pages 585–592, New York, NY, USA, July 2012. Association for Computing Machinery. ISBN 978-1-4503-1177-9. doi: 10.1145/2330163.2330247.
- [19] Anton Bouter, Tanja Alderliesten, and Peter A. N. Bosman. A novel model-based evolutionary algorithm for multi-objective deformable image registration with content mismatch and large deformations: Benchmarking efficiency and quality. In *Medical Imaging 2017: Image Processing*, volume 10133, pages 304–311. SPIE, February 2017. doi: 10.1117/12.2254144.
- [20] Anton Bouter, Tanja Alderliesten, Cees Witteveen, and Peter A. N. Bosman. Exploiting linkage information in real-valued optimization with the real-valued gene-pool optimal mixing evolutionary algorithm. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 705–712, Berlin Germany, July 2017. ACM. ISBN 978-1-4503-4920-8. doi: 10.1145/3071178.3071272.
- [21] Anton Bouter, Ngoc Hoang Luong, Cees Witteveen, Tanja Alderliesten, and Peter A. N. Bosman. The multi-objective real-valued gene-pool optimal mixing evolutionary algorithm. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO '17, pages 537–544, New York, NY, USA, July 2017. Association for Computing Machinery. ISBN 978-1-4503-4920-8. doi: 10.1145/3071178.3071274.
- [22] Anton Bouter, Tanja Alderliesten, Arjan Bel, Cees Witteveen, and Peter A. N. Bosman. Large-scale parallelization of partial evaluations in evolutionary algorithms for real-world problems. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 1199–1206, Kyoto Japan, July 2018. ACM. ISBN 978-1-4503-5618-3. doi: 10.1145/3205455.3205610.
- [23] Anton Bouter, Tanja Alderliesten, and Peter Bosman. Achieving highly scalable evolutionary real-valued optimization by exploiting partial evaluations. *Evolutionary Computation*, pages 1–27, June 2020. ISSN 1063-6560. doi: 10.1162/evco_a_00275.
- [24] Anton Bouter, Stefanus C. Maree, Tanja Alderliesten, and Peter A. N. Bosman. Leveraging conditional linkage models in gray-box optimization with the real-valued gene-pool optimal mixing evolutionary algorithm. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference*, GECCO '20, pages 603–611, New York, NY, USA, June 2020. Association for Computing Machinery. ISBN 978-1-4503-7128-5. doi: 10.1145/3377930.3390225.
- [25] Anton Bouter, Tanja Alderliesten, and Peter A.N. Bosman. GPU-Accelerated Parallel Gene-pool Optimal Mixing Applied to Multi-Objective Deformable Image Registration. In *2021 IEEE Congress on Evolutionary Computation (CEC)*, pages 2539–2548, June 2021. doi: 10.1109/CEC45853.2021.9504840.
- [26] K. K. Brock, M. B. Sharpe, L. A. Dawson, S. M. Kim, and D. A. Jaffray. Accuracy of finite element model-based multi-organ deformable image registration. *Medical Physics*, 32(6Part1):1647–1659, 2005. ISSN 2473-4209. doi: 10.1118/1.1915012.
- [27] Rohit Chandra, Leo Dagum, David Kohr, Ramesh Menon, Dror Maydan, and Jeff McDonald. *Parallel Programming in OpenMP*. Morgan kaufmann, 2001.
- [28] Yilun Chen, Ami Wiesel, Yonina C. Eldar, and Alfred O. Hero. Shrinkage Algorithms for MMSE Covariance Estimation. *IEEE Transactions on Signal Processing*, 58(10):5016–5029, October 2010. ISSN 1941-0476. doi: 10.1109/TSP.2010.2053029.

- [29] Gianni D'Angelo and Francesco Palmieri. GGA: A modified genetic algorithm with gradient-based local search for solving constrained optimization problems. *Information Sciences*, 547:136–162, February 2021. ISSN 00200255. doi: 10.1016/j.ins.2020.08.040.
- [30] Bob D de Vos, Floris F Berendsen, Max A Viergever, Hessam Sokooti, Marius Staring, and Ivana Išgum. A deep learning framework for unsupervised affine and deformable image registration. *Medical Image Analysis*, page 16, 2019.
- [31] Kalyanmoy Deb. Multi-objective Optimisation Using Evolutionary Algorithms: An Introduction. In Lihui Wang, Amos H. C. Ng, and Kalyanmoy Deb, editors, *Multi-Objective Evolutionary Optimisation for Product Design and Manufacturing*, pages 3–34. Springer, London, 2011. ISBN 978-0-85729-652-8. doi: 10.1007/978-0-85729-652-8_1.
- [32] Timo M. Deist, Stefanus C. Maree, Tanja Alderliesten, and Peter A. N. Bosman. Multi-objective Optimization by Uncrowded Hypervolume Gradient Ascent. In Thomas Bäck, Mike Preuss, André Deutz, Hao Wang, Carola Doerr, Michael Emmerich, and Heike Trautmann, editors, *Parallel Problem Solving from Nature – PPSN XVI*, Lecture Notes in Computer Science, pages 186–200, Cham, 2020. Springer International Publishing. ISBN 978-3-030-58115-2. doi: 10.1007/978-3-030-58115-2_13.
- [33] Neil K. Dhingra, Sei Zhen Khong, and Mihailo R. Jovanović. The proximal augmented Lagrangian method for nonsmooth composite optimization. *IEEE Transactions on Automatic Control*, 64(7):2861–2868, July 2019. ISSN 0018-9286, 1558-2523, 2334-3303. doi: 10.1109/TAC.2018.2867589.
- [34] Ilan Gronau and Shlomo Moran. Optimal implementations of UPGMA and other common clustering algorithms. *Information Processing Letters*, 104(6):205–210, December 2007. ISSN 00200190. doi: 10.1016/j.ipl.2007.07.002.
- [35] Gaël Guennebaud, Benoît Jacob, et al. Eigen v3, 2010.
- [36] Damy M. F. Ha, Timo M. Deist, and Peter A. N. Bosman. Hybridizing Hypervolume-Based Evolutionary Algorithms and Gradient Descent by Dynamic Resource Allocation. In Günter Rudolph, Anna V. Kononova, Hernán Aguirre, Pascal Kerschke, Gabriela Ochoa, and Tea Tušar, editors, *Parallel Problem Solving from Nature – PPSN XVII*, Lecture Notes in Computer Science, pages 179–192, Cham, 2022. Springer International Publishing. ISBN 978-3-031-14721-0. doi: 10.1007/978-3-031-14721-0_13.
- [37] Haili Chui and A. Rangarajan. A new algorithm for non-rigid point matching. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No. PR00662)*, volume 2, pages 44–51, Hilton Head Island, SC, USA, 2000. IEEE Comput. Soc. ISBN 978-0-7695-0662-3. doi: 10.1109/CVPR.2000.854733.
- [38] Lasse Hansen and Mattias P. Heinrich. GraphRegNet: Deep Graph Regularisation Networks on Sparse Keypoints for Dense Registration of 3D Lung CTs. *IEEE Transactions on Medical Imaging*, 40(9):2246–2257, September 2021. ISSN 1558-254X. doi: 10.1109/TMI.2021.3073986.
- [39] Nikolaus Hansen and Andreas Ostermeier. Completely Derandomized Self-Adaptation in Evolution Strategies. *Evolutionary Computation*, 9(2):159–195, June 2001. ISSN 1063-6560, 1530-9304. doi: 10.1162/106365601750190398.
- [40] Nikolaus Hansen, Anne Auger, Raymond Ros, Steffen Finck, and Petr Pošík. Comparing results of 31 algorithms from the black-box optimization benchmarking BBOB-2009. In *Proceedings of the 12th Annual Conference Comp on Genetic and Evolutionary Computation - GECCO '10*, page 1689, Portland, Oregon, USA, 2010. ACM Press. ISBN 978-1-4503-0073-5. doi: 10.1145/1830761.1830790.
- [41] Grant Haskins, Uwe Kruger, and Pingkun Yan. Deep learning in medical image registration: A survey. *Machine Vision and Applications*, 31(1-2):8, February 2020. ISSN 0932-8092, 1432-1769. doi: 10.1007/s00138-020-01060-x.
- [42] Douglas M. Hawkins. The Problem of Overfitting. *Journal of Chemical Information and Computer Sciences*, 44(1):1–12, January 2004. ISSN 0095-2338. doi: 10.1021/ci0342472.

- [43] Alessa Hering, Keelin Murphy, and Bram van Ginneken. Learn2Reg Challenge: CT Lung Registration - Training Data, May 2020.
- [44] Nicholas J. Higham. Cholesky factorization. *Wiley Interdisciplinary Reviews: Computational Statistics*, 1(2):251–254, September 2009. ISSN 1939-5108. doi: 10.1002/wics.18.
- [45] Derek L. G. Hill, Philipp G. Batchelor, Mark Holden, and David J. Hawkes. Medical image registration. *Physics in Medicine & Biology*, 46(3):R1, March 2001. ISSN 0031-9155. doi: 10.1088/0031-9155/46/3/201.
- [46] Mohammad Hussein, Adeyemi Akintonde, Jamie McClelland, Richard Speight, and Catharine H Clark. Clinical use, challenges, and barriers to implementation of deformable image registration in radiotherapy – the need for guidance and QA tools. *The British Journal of Radiology*, 94(1122):20210001, June 2021. ISSN 0007-1285, 1748-880X. doi: 10.1259/bjr.20210001.
- [47] Luis Ibanez, William Schroeder, Lydia Ng, and Josh Cates. The ITK Software Guide: The Insight Segmentation and Registration Toolkit. *Kitware Inc*, January 2005.
- [48] Stefan Klein, Marius Staring, and Josien P. W. 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, December 2007. ISSN 1941-0042. doi: 10.1109/TIP.2007.909412.
- [49] Stefan Klein, Marius Staring, Keelin Murphy, Max A. Viergever, and Josien P. W. Pluim. Elastix: A Toolbox for Intensity-Based Medical Image Registration. *IEEE Transactions on Medical Imaging*, 29(1):196–205, January 2010. ISSN 1558-254X. doi: 10.1109/TMI.2009.2035616.
- [50] J. Kybic and M. Unser. Fast parametric elastic image registration. *IEEE Transactions on Image Processing*, 12(11):1427–1442, November 2003. ISSN 1941-0042. doi: 10.1109/TIP.2003.813139.
- [51] Hava Lester and Simon R. Arridge. A survey of hierarchical non-linear medical image registration. *Pattern Recognition*, 32(1):129–149, January 1999. ISSN 0031-3203. doi: 10.1016/S0031-3203(98)00095-8.
- [52] Ngoc Hoang Luong, Han La Poutre, and Peter A.N. Bosman. Multi-objective gene-pool optimal mixing evolutionary algorithms. In *Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation*, pages 357–364, Vancouver BC Canada, July 2014. ACM. ISBN 978-1-4503-2662-9. doi: 10.1145/2576768.2598261.
- [53] Tom Lyche and Knut Morken. Spline methods draft. *Department of Informatics, Center of Mathematics for Applications, University of Oslo, Oslo*, pages 3–8, 2008.
- [54] Frederik Maes, Dirk Vandermeulen, and Paul Suetens. Comparative evaluation of multiresolution optimization strategies for multimodality image registration by maximization of mutual information. *Medical Image Analysis*, 3(4):373–386, December 1999. ISSN 1361-8415. doi: 10.1016/S1361-8415(99)80030-9.
- [55] J. B. Antoine Maintz and Max A. Viergever. A survey of medical image registration. *Medical Image Analysis*, 2(1):1–36, March 1998. ISSN 1361-8415. doi: 10.1016/S1361-8415(01)80026-8.
- [56] S. C. Marea, T. Alderliesten, and P. A. N. Bosman. Uncrowded Hypervolume-based Multi-objective Optimization with Gene-pool Optimal Mixing. *arXiv:2004.05068 [cs]*, April 2020.
- [57] Makoto Matsumoto and Takuji Nishimura. Mersenne twister: A 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Transactions on Modeling and Computer Simulation*, 8(1):3–30, January 1998. ISSN 1049-3301, 1558-1195. doi: 10.1145/272991.272995.
- [58] Matthew McCormick, Xiaoxiao Liu, Luis Ibanez, Julien Jomier, and Charles Marion. ITK: Enabling reproducible research and open science. *Frontiers in Neuroinformatics*, 8, 2014. ISSN 1662-5196.
- [59] Marc Modat, M. Jorge Cardoso, Pankaj Daga, David Cash, Nick C. Fox, and Sébastien Ourselin. Inverse-Consistent Symmetric Free Form Deformation. In Benoît M. Dawant, Gary E. Christensen, J. Michael Fitzpatrick, and Daniel Rueckert, editors, *Biomedical Image Registration*, Lecture Notes in Computer Science, pages 79–88, Berlin, Heidelberg, 2012. Springer. ISBN 978-3-642-31340-0. doi: 10.1007/978-3-642-31340-0_9.

- [60] Seungjong Oh and Siyong Kim. Deformable image registration in radiation therapy. *Radiation Oncology Journal*, 35(2):101–111, June 2017. ISSN 2234-1900. doi: 10.3857/roj.2017.00325.
- [61] Chiara Paganelli, Giorgia Meschini, Silvia Molinelli, Marco Riboldi, and Guido Baroni. “Patient-specific validation of deformable image registration in radiation therapy: Overview and caveats”. *Medical Physics*, 45(10):e908–e922, 2018. ISSN 2473-4209. doi: 10.1002/mp.13162.
- [62] Kleopatra Pirpinia, Peter A. N. Bosman, Jan-Jakob Sonke, Marcel van Herk, and Tanja Alderliesten. On the usefulness of gradient information in multi-objective deformable image registration using a B-spline-based dual-dynamic transformation model: Comparison of three optimization algorithms. In *Medical Imaging 2015: Image Processing*, volume 9413, pages 866–875. SPIE, March 2015. doi: 10.1117/12.2082172.
- [63] Kleopatra Pirpinia, Peter A. N. Bosman, Jan-Jakob Sonke, Marcel van Herk, and Tanja Alderliesten. A first step toward uncovering the truth about weight tuning in deformable image registration. In *Medical Imaging 2016: Image Processing*, volume 9784, pages 1099–1107. SPIE, March 2016. doi: 10.1117/12.2216370.
- [64] Kleopatra Pirpinia, Peter A N Bosman, Claudette E Loo, Gonneke Winter-Warnars, Natasja N Y Janssen, Astrid N Scholten, Jan-Jakob Sonke, Marcel van Herk, and Tanja Alderliesten. The feasibility of manual parameter tuning for deformable breast MR image registration from a multi-objective optimization perspective. *Physics in Medicine & Biology*, 62(14):5723–5743, June 2017. ISSN 1361-6560. doi: 10.1088/1361-6560/aa6edc.
- [65] Kleopatra Pirpinia, Peter A. N. Bosman, Jan-Jakob Sonke, Marcel van Herk, and Tanja Alderliesten. Evolutionary Machine Learning for Multi-Objective Class Solutions in Medical Deformable Image Registration. *Algorithms*, 12(5):99, May 2019. doi: 10.3390/a12050099.
- [66] John C Platt and Alan H Barr. Constrained Differential Optimization. *American Institute of Physics*, 1988.
- [67] Josien P.W. Pluim, J.B. Antoine Maintz, and Max A. Viergever. Interpolation Artefacts in Mutual Information-Based Image Registration. *Computer Vision and Image Understanding*, 77(2):211–232, February 2000. ISSN 10773142. doi: 10.1006/cviu.1999.0816.
- [68] Josien P.W. Pluim, Sascha E.A. Muenzing, Koen A.J. Eppenhof, and Keelin Murphy. The truth is hard to make: Validation of medical image registration. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pages 2294–2300, Cancun, December 2016. IEEE. ISBN 978-1-5090-4847-2. doi: 10.1109/ICPR.2016.7899978.
- [69] J.P.W. Pluim, J.B.A. Maintz, and M.A. Viergever. Mutual-information-based registration of medical images: A survey. *IEEE Transactions on Medical Imaging*, 22(8):986–1004, August 2003. ISSN 1558-254X. doi: 10.1109/TMI.2003.815867.
- [70] Claudia E. Rodriguez-Carranza and Murray H. Loew. Weighted and deterministic entropy measure for image registration using mutual information. In *Medical Imaging 1998: Image Processing*, volume 3338, pages 155–166. SPIE, June 1998. doi: 10.1117/12.310854.
- [71] Marc-Michel Rohé, Manasi Datar, Tobias Heimann, Maxime Sermesant, and Xavier Pennec. SVF-Net: Learning Deformable Image Registration Using Shape Matching. In Maxime Descoteaux, Lena Maier-Hein, Alfred Franz, Pierre Jannin, D. Louis Collins, and Simon Duchesne, editors, *Medical Image Computing and Computer Assisted Intervention - MICCAI 2017*, volume 10433, pages 266–274. Springer International Publishing, Cham, 2017. ISBN 978-3-319-66181-0 978-3-319-66182-7. doi: 10.1007/978-3-319-66182-7_31.
- [72] D. Rueckert, L.I. Sonoda, C. Hayes, D.L.G. Hill, M.O. Leach, and D.J. Hawkes. Nonrigid registration using free-form deformations: Application to breast MR images. *IEEE Transactions on Medical Imaging*, 18(8): 712–721, August 1999. ISSN 1558-254X. doi: 10.1109/42.796284.
- [73] Daniel Rueckert, Paul Aljabar, Rolf A. Heckemann, Joseph V. Hajnal, and Alexander Hammers. Diffeomorphic Registration Using B-Splines. In Rasmus Larsen, Mads Nielsen, and Jon Sporring, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2006*, Lecture Notes in Computer Science, pages 702–709, Berlin, Heidelberg, 2006. Springer. ISBN 978-3-540-44728-3. doi: 10.1007/11866763_86.

- [74] Boris Schäling. *The Boost C++ Libraries*. XML Press, 2014. ISBN 978-1-937434-36-6 1-937434-36-2.
- [75] James Shackelford, Qi Yang, Ana Lourenço, Nadya Shusharina, Nagarajan Kandasamy, and Gregory Sharp. Analytic Regularization of Uniform Cubic B-spline Deformation Fields. In *Medical Image Computing and Computer-Assisted Intervention : MICCAI ... International Conference on Medical Image Computing and Computer-Assisted Intervention*, volume 15, pages 122–9, October 2012. ISBN 978-3-642-33417-7. doi: 10.1007/978-3-642-33418-4_16.
- [76] C. E. Shannon. Prediction and Entropy of Printed English. *Bell System Technical Journal*, 30(1):50–64, 1951. ISSN 1538-7305. doi: 10.1002/j.1538-7305.1951.tb01366.x.
- [77] G. Sharp, R. LI, John Wolfgang, G. Chen, Marta Peroni, Maria Spadea, Shinichiro Mori, J. Zhang, James Shackelford, and Nagarajan Kandasamy. PLASTIMATCH– AN OPEN SOURCE SOFTWARE SUITE FOR RADIOTHERAPY IMAGE PROCESSING. May 2010.
- [78] Gregory Sharp, Marta Peroni, Rui Li, James Shackelford, and Nagarajan Kandasamy. Evaluation of Plastimatch B-Spline Registration on the EMPIRE10 Data Set. *Medical Physics*, page 10, 2010.
- [79] Ilya M. Sobol', Danil Asotsky, Alexander Kreinin, and Sergei Kucherenko. Construction and Comparison of High-Dimensional Sobol' Generators. *Wilmott*, 2011(56):64–79, November 2011. ISSN 15406962. doi: 10.1002/wilm.10056.
- [80] Aristeidis Sotiras, Christos Davatzikos, and Nikos Paragios. Deformable Medical Image Registration: A Survey. *IEEE Transactions on Medical Imaging*, 32(7):1153–1190, July 2013. ISSN 1558-254X. doi: 10.1109/TMI.2013.2265603.
- [81] C. Studholme, D. L. G. Hill, and D. J. Hawkes. An overlap invariant entropy measure of 3D medical image alignment. *Pattern Recognition*, 32(1):71–86, January 1999. ISSN 0031-3203. doi: 10.1016/S0031-3203(98)00091-0.
- [82] Min-Jea Tahk, Hyun-Wook Woo, and Moon-Su Park. A hybrid optimization method of evolutionary and gradient search. *Engineering Optimization*, 39(1):87–104, January 2007. ISSN 0305-215X, 1029-0273. doi: 10.1080/03052150600957314.
- [83] Christine Tanner, Firat Ozdemir, Romy Profanter, Valeriy Vishnevsky, Ender Konukoglu, and Orcun Goksel. Generative Adversarial Networks for MR-CT Deformable Image Registration. *arXiv:1807.07349 [cs]*, July 2018.
- [84] Nicholas J. Tustison, Brian A. Avants, and James C. Gee. Improved FFD B-Spline Image Registration. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8, October 2007. doi: 10.1109/ICCV.2007.4409161.
- [85] M. Unser. Splines: A perfect fit for signal and image processing. *IEEE Signal Processing Magazine*, 16(6):22–38, November 1999. ISSN 1558-0792. doi: 10.1109/79.799930.
- [86] Ola Westrand and Stina Svensson. The ANACONDA algorithm for deformable image registration in radiotherapy. *Medical Physics*, 42(1):40–53, January 2015. ISSN 2473-4209. doi: 10.1118/1.4894702.
- [87] Zhenlin Xu and Marc Niethammer. DeepAtlas: Joint Semi-supervised Learning of Image Registration and Segmentation. In Dinggang Shen, Tianming Liu, Terry M. Peters, Lawrence H. Staib, Caroline Essert, Sean Zhou, Pew-Thian Yap, and Ali Khan, editors, *Medical Image Computing and Computer Assisted Intervention – MICCAI 2019*, Lecture Notes in Computer Science, pages 420–429, Cham, 2019. Springer International Publishing. ISBN 978-3-030-32245-8. doi: 10.1007/978-3-030-32245-8_47.
- [88] Özgür Yeniay. Penalty Function Methods for Constrained Optimization with Genetic Algorithms. *Mathematical and Computational Applications*, 10(1):45–56, April 2005. ISSN 2297-8747. doi: 10.3390/mca10010045.
- [89] Xiaoran Zhang, Hexiang Dong, Di Gao, and Xiao Zhao. A Comparative Study for Non-rigid Image Registration and Rigid Image Registration. *arXiv:2001.03831 [cs, eess]*, January 2020.