Evolvability-guided Optimization of Linear Deformation Setups for Evolutionary Design Optimization

Dissertation

zur Erlangung des akademischen Grades des
Doktors der Naturwissenschaften (Dr. rer. nat.)
an der Technischen Fakultät der Universität Bielefeld

vorgelegt von
Andreas Richter

Bielefeld 2018

Versicherung

Hiermit versichere ich,

- dass mir die geltende Promotionsordnung der Fakultät bekannt ist,
- dass ich die Dissertation selbst angefertigt habe, keine Textabschnitte von Dritten oder eigener Prüfungsarbeiten ohne Kennzeichnung übernommen und alle benutzten Hilfsmittel und Quellen in meiner Arbeit angegeben habe,
- dass Dritte weder unmittelbar noch mittelbar geldwerte Leistungen von mir für Vermittlungstätigkeiten oder für Arbeiten erhalten haben, die im Zusammenhang mit dem Inhalt der vorgelegten Dissertation stehen,
- dass ich die Dissertation noch nicht als Prüfungsarbeit für eine staatliche oder andere wissenschaftliche Prüfung eingereicht habe und
- dass ich keine gleiche, oder in wesentlichen Teilen ähnliche oder eine andere Abhandlung bei einer anderen Hochschule als Dissertation eingereicht habe.

Bielefeld, 2018 ______________________ Andreas Richter

Printed on non-aging paper in compliance with DIN-ISO 9706.

Abstract

This thesis targets efficient solutions for optimal representation setups for evolutionary design optimization problems. The representation maps the abstract parameters of an optimizer to a meaningful variation of the design model, e.g., the shape of a car. Thereby, it determines the convergence speed to and the quality of the final result. Thus, engineers are eager to employ well-tuned representations to achieve high-quality design solutions. But, setting up optimal representations is a cumbersome process because the setup procedure requires detailed knowledge about the objective functions, e.g., a fluid dynamics simulation, and the parameters of the employed representation itself. Thus, we target efficient routines to set up representations automatically to support engineers from their tedious, partly manual work.

Inspired by the concept of evolvability, we present novel quality criteria for the evaluation of linear deformations commonly applied as representations. We define and analyze the criteria variability, regularity, and improvement potential which measure the expected quality and convergence speed of an evolutionary design optimization process based on the linear deformation setup. Moreover, we target the efficient optimization of deformation setups with respect to these three criteria. In dynamic design optimization scenarios a suitable compromise between exploration and exploitation is crucial for efficient solutions. We discuss the construction of optimal compromises for these dynamic scenarios with our criteria because they characterize exploration and exploitation.

As a result an engineer can initialize and adjust the deformation setup for improved convergence speed of a design process and for enhanced quality of the design solutions with our methods.

List of Abbreviations

BFGS	Gradient-based Broyden–Fletcher–Goldfarb–Shanno algorithm.
CMA-ES	Optimization algorithm: Covariance matrix adaptation evolution strategy.
FFD	Free-form deformation.
NSGA-II	Multi-objective optimization algorithm: Non-dominated sorting genetic algorithm - II.
OLS	Orthogonal least squares, sampling strategy.
RBF	Deformation with radial basis function.
SVD	Singular value decomposition.

List of Definitions

Evolvability	Biological concept to characterize exploration, exploitation, and convergence speed which we employ for representations.
Exploitation	The ability to explore information for specialized phenotypes.
Exploration	The ability to generate different phenotypes.
Genotype	Parameter space of an evolutionary algorithm.
Improvement potential	Our mathematical criterion to measure how a representation exploits given information.
Phenotype	The space of objects, e.g., designs, corresponding to certain genotypes.
Regularity	Our mathematical criterion to measure the convergence speed induced by a representation.
Representation	Mapping from genotype to phenotype, i.e., linear deformation which maps parameter variation to design variation.
Variability	Our mathematical criterion to measure exploratory capabilities of a representation.

List of Symbols

$\mathbb{R}^d$	d dimensional space of real numbers.
$\mathbf{D}$	Matrix of handle displacement.
$\mathbf{L}$	Laplacian matrix.
$\mathbf{N}$	Deformation matrix of FFD based on spline function.
$\mathbf{P}$	Matrix of deformation parameters.
$\mathbf{Q}$	Matrix of polynomial coefficients.
$\mathbf{W}$	Matrix of RBF weights.
$\mathbf{X}$	All vertices of a mesh listed in a matrix.
$\mathbf{d}$	Handle displacement as deformation parameter.
$\mathbf{p}$	Deformation parameter, e.g,. a rbf weight/handle or a displacement of a FFD control point.
$\mathbf{q}$	Coefficients of a linear polynomial.
$\mathbf{U}$	Matrix of left singular vectors.
$\mathbf{V}$	Matrix of right singular vectors.
$\mathbf{\Phi}$	Deformation matrix for indirect RBF manipulation.
$\mathbf{\Psi}$	Interpolation matrix for direct RBF manipulation.
$\mathbf{\Sigma}$	Diagonal matrix of singular values.
$\mathbf{o}(\cdot)$	Deformation operator for arbitrary domains.

$\mathbf{w}$	RBF weight as deformation parameter.
$\mathbf{x}$	Vertex of a mesh.
$\mathbf{O}$	Deformation matrix.
$\delta\mathbf{C}$	Displacement matrix for all FFD control points.
$\delta\mathbf{c}$	Displacement of a single FFD control point as deformation parameter.
$\mathbf{u}$	Context dependent: Local coordinates of FFD or singular vector of SVD.
∇	Gradient operator.
Δ	Laplace operator.
$o(\cdot)$	Deformation operator for 1D domains.
$\mathcal{C}$	Unordered set of RBF centers.
$\mathcal{X}$	Unordered set of vertices of a mesh.
$\mathcal{N}(\mathbf{x})$	Set of neighbors of vertex $\mathbf{x}$.
$\partial c_{j,x}$	Partial derivative with respect to the x coordinate of the RBF center $\mathbf{c}_j$.
π	Basis of linear bi-/trivariate polynomials.
σ	Singular value of a matrix.
$N(\cdot)$	Spline basis function.
$\varphi(\cdot)$	Basis function (kernel) of a RBF.
m	Number of vertices of a mesh.
n	Number of parameters of a deformation.

Contents

1 Introduction 1

2 Linear Deformations in Evolutionary Design Optimization 7
2.1 Evolutionary Design Optimization . . . 8
2.2 Radial Basis Functions . . . 12
2.3 Free-Form Deformation . . . 15
2.4 Shell Deformations . . . 16
2.5 Summary . . . 18

3 The General Concept of Evolvability 19
3.1 Complex System Engineering . . . 20
3.2 Evolvability . . . 22
3.2.1 Regularity . . . 23
3.2.2 Variability . . . 24
3.2.3 Improvement Potential . . . 25
3.3 Further Targets of Evolvability . . . 26
3.3.1 Robustness vs. Evolvability . . . 26
3.3.2 Modularity . . . 27
3.3.3 Evolvability of Algorithms . . . 28
3.4 The conflict: Exploration vs. Exploitation . . . 28
3.5 Summary . . . 29

4 Definition of Evolvability Criteria for Linear Deformations 33
4.1 Variability . . . 33
4.2 Regularity . . . 35
4.3 Improvement Potential . . . 37
4.4 Summary . . . 38

5 Evaluation of the Evolvability Criteria 39
5.1 Test Scenario: 1D Function Approximation 40
5.1.1 Results: Variability 42
5.1.2 Results: Regularity 44
5.1.3 Results: Improvement Potential 46
5.2 Test Scenario: 3D Template Fitting 49
5.3 Summary 51

6 Pareto-optimal RBF Centers 53
6.1 Multi-objective Evolutionary Optimization 55
6.2 Single-objective evolutionary Optimization 64
6.3 Heuristic Approaches 65
6.4 Gradient-based Optimization 71
6.5 Summary 76

7 Optimal Preferences for Design Optimization 79
7.1 Static Design Loop 81
7.2 Dynamic Design Loop 87
7.3 Summary 93

8 Extended Analysis of Regularity and Variability 95
8.1 Matrix Orthogonalization for Optimal Regularity 96
8.2 Redefinition of Variability 104
8.3 Summary 109

9 Conclusion 111

Bibliography 115

1 Introduction

The increasing complexity in modern industrial design processes requires advanced optimization methods to come up with novel and high-quality solutions for successful business. For example in automotive product design concurrent development processes are applied to deal with different requirements, e.g., in physical domains such as aerodynamic or structural performance criteria, manufacturing process layout, or according to design features specified by current customer demands. Moreover, these requirements change over time and thus, an efficient development process needs to allow a high degree of flexibility to cope with these dynamic environments.

For the dynamically changing and highly interacting requirements of an industrial design process we are looking for alternatives to the *classical engineering* paradigms, which can only be applied for limited success [MBBY06]. These paradigms focus on specific, independent sub-problems only, for instance optimization of representation parameters, modeling of quality criteria, or the choice of the optimization routine. Moreover, *classical engineering* approaches don't focus on solutions for dynamically changing environments. In contrast, the concept of *complex system engineering* suggests to consider relevant features *simultaneously* because they are strongly linked and they interact with each other. Furthermore, complex system engineering offers concepts to handle changing environments.

Biologically-inspired population-based evolutionary optimization algorithms are recommended for complex systems [MBBY06]. They are robust to local optima, constraints or fitness criteria, can easily be switched, and these criteria can be considered simultaneously. Thus evolutionary algorithms are our preferred choice.

There are several possibilities to tune an evolutionary design optimization process. The most common approach targets the configuration of the applied algorithms by, e.g., employing efficient mutation and selection operators or creating new algorithmic concepts. Other attempts target the fitness models by replacing highly expensive models with surrogates which are easier to evaluate. But, as pointed out by many researchers (e.g., [ITN10, ČLM13]), the representation is a core factor for success in evolutionary optimization algorithms because it defines the design space and how fast an optimizer can search it. Thus, designers are eager to employ efficient representations for these biologically-inspired methods. This motivates our research on concepts and approaches to improve representations and thereby improve the performance of evolutionary design optimization.

The representation defines the mapping from the genotype space to the phenotype space in the evolutionary context. Commonly employed representations for design optimization are linear deformation methods, which map abstract model parameters (as the genotype) to meaningful design variation (as the phenotype). Motivated by the successful application of linear deformations with *radial basis functions* (RBFs) or *free-form deformation* (FFD), e.g., for car, train, or airplane optimization scenarios [SMB12, MOS06, CBG$^+$14, ZYSD16], we employ mainly these methods as representations. In case of RBFs the parameters, which define the genotype–phenotype mapping, are the location of centers and the choice of radial symmetric kernels. The parameters of the FFD representation are a control grid and chosen spline basis functions. Because RBFs and FFD are linear, we express the deformation with a deformation matrix, which encodes the information about the control points (centers or grid) and basis functions (kernels or splines).

In a design optimization cycle the model or optimization parameters, e.g., weights of RBFs or displacements of FFD grids, are optimized for improving the quality of the designs, which is not our focus of research. We target an optimal deformation setup. This consists of choosing an optimal distribution of control points and employing optimal basis functions. Moreover, we target optimal numerical properties of the deformation matrix as a third objective.

The construction of optimal deformation setups/matrices requires suitable quality criteria. The meta-attribute *evolvability* is motivated by complex system engineering [MBBY06]. Keeping the system's evolvability high throughout an optimization cycle on the one hand ensures the adaptation to changing conditions and on the other hand promotes specialization, if needed.

Thus, we employ the concept of evolvability for the evaluation and optimization of deformation setups in evolutionary design optimization like [Men11, LM12]. In Figure 1.1 we sketch a design process utilizing evolvability for an improved automotive design optimization. Based on a car model and information about the current fitness environment (colored roof), the deformation (the center distribution of a RBF deformation operator in this example) is set up for high evolvability. Afterwards, the design is optimized with an optimal setup resulting in superior performance of the design optimization compared to setups of lower quality. During the design optimization information about the possibly changing fitness environment are extracted to adapt the representation, if needed. With an improved deformation setup the design optimization continues leading to high performance during the whole design process. Note that lower ranked deformation setups for current fitness information might become superior later on if, e.g., a drastic change of the fitness occurs. For example if the general shape of the car has to be modified instead of the roof then the second setup in Figure 1.1 would be more promising.

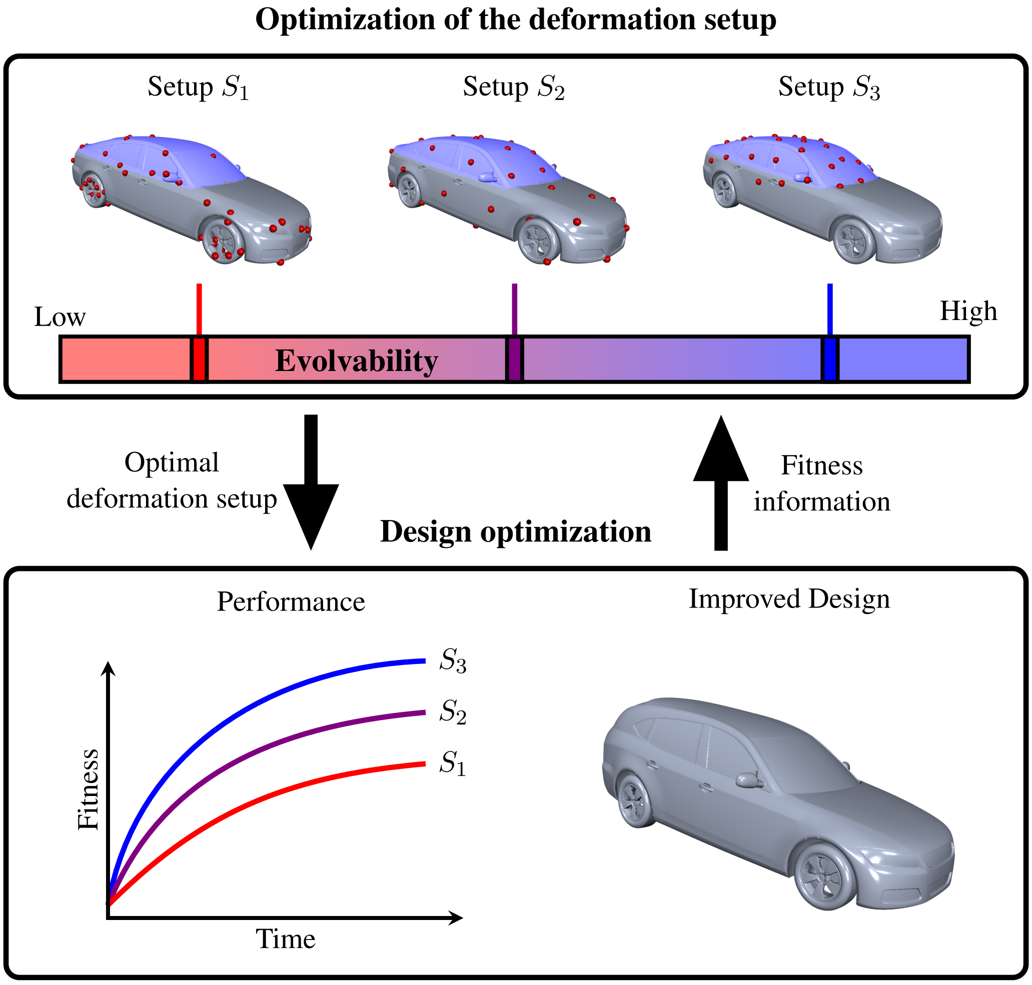


Figure 1.1: Overview of a (automotive) design process. The deformation setup (the centers of a RBF deformation, red dots) is optimized for high evolvability according to given fitness information (colored roof) before the design optimization is started. Employing setups with lower evolvability would result in lower performance of the design optimization. During the design optimization stage information about the possibly changing fitness environment is extracted, which can be utilized for the adaptation of the deformation setup. This would lead to an alternating procedure switching between optimization/adaptation of the deformation setup and design optimization.

The optimization of deformation setups for an enhanced (dynamic) design process requires the solution of two major tasks, which are the main contributions of this thesis:

(1) The suitable definition of evolvability for linear deformations.

(2) The optimization of linear deformation setups for (dynamic) design optimization.

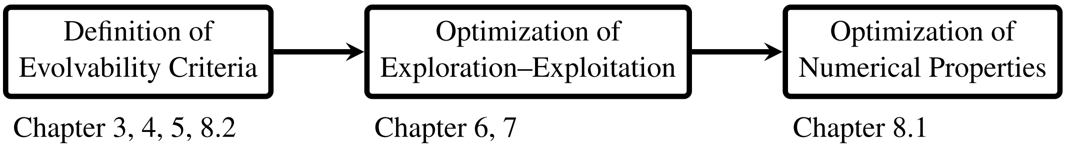

Definition of Evolvability Criteria
Optimization of Exploration–Exploitation
Optimization of Numerical Properties
Chapter 3, 4, 5, 8.2
Chapter 6, 7
Chapter 8.1

Figure 1.2: Workflow of this thesis.

In essence we proceed as shown in Figure 1.2. First, we define quality criteria for linear deformations based on the concept of evolvability. These criteria characterize exploratory and exploitative capabilities as well as numerical properties of the deformation setup. Then we show that the quality criteria indeed estimate the performance of the deformation setups. They characterize the expected quality of solutions as well as the convergence speed of their optimization. Afterwards, we conduct and propose solutions for a Pareto optimization targeting optimal RBF center distributions, which trade off between exploration and exploitation. Based on this optimization of compromise setups we analyze which optimal compromise should be chosen for different dynamic design optimization scenarios. And finally, we optimize numerical properties of the deformation setup.

Another, but minor contribution of this thesis is the comparison between the employed RBF and FFD representations in particular with respect to their evolvability scores to support engineers and designers in the process of choosing between the RBF and FFD methods.

We published our research on evolvability for evolutionary design optimization at major evolutionary conferences, the *congress on evolutionary computation* (CEC) and the *genetic and evolutionary computation conference* (GECCO), and the more specialized *evolutionary multi-objective optimization* conference (EMO):

- Andreas Richter, Mario Botsch, and Stefan Menzel. Evolvability of representations in complex system engineering: a survey. In *Proceedings of IEEE Congress on Evolutionary Computation*, pages 1327–1335, 2015
- Andreas Richter, Jascha Achenbach, Stefan Menzel, and Mario Botsch. Evolvability as a quality criterion for linear deformation representations in evolutionary optimization. In *Proceedings of IEEE Congress on Evolutionary Computation*, pages 901–910, 2016
- Andreas Richter, Jascha Achenbach, Stefan Menzel, and Mario Botsch. Multi-objective representation setups for deformation-based design optimization. In *Proceedings of 9th International Conference on Evolutionary Multi-Criterion Optimization*, pages 514–528, 2017
- Andreas Richter, Stefan Menzel, and Mario Botsch. Preference-guided adaptation of deformation representations for evolutionary design optimization. In *Proceedings of IEEE Congress on Evolutionary Computation*, pages 2110–2119, 2017
- Andreas Richter, Stefan Dresselhaus, Stefan Menzel, and Mario Botsch. Orthogonalization of linear representations for efficient evolutionary design optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 1356–1363, 2018

The second paper [RAMB16] is runner-up for the best student paper award at the CEC 2016.

The remainder of this thesis is structured as follows. In Chapter 2, we start with the discussion of our employed evolutionary framework for design optimization. Moreover, we present the concepts behind the prominent linear deformation methods RBF, FFD, and shell deformation, as examples of linear deformation representations, which we employ for our analysis. This chapter guides us to a general notation for the linear deformation operator, that is expressed as a deformation matrix.

Motivated by the concept of complex system engineering we continue with the discussion of the concept of evolvability, its biological motivations, and already existing (mathematical) realizations in Chapter 3. Moreover, we discuss links to the well-known concepts of *exploration* and *exploitation*. Our conclusion is a three-staged approach consisting of the aspects, *variability*, *regularity*, and *improvement potential*. These three criteria are necessary in order to model a comprehensive realization of evolvability.

We transform the concept of evolvability for deformation matrices as our representations in Chapter 4. We propose easy-to-evaluate mathematical models for variability, regularity, and improvement potential. These criteria quantify the expected convergence speed of an evolutionary optimization and the quality of its results.

The evaluation of two test scenarios, 1D function approximation and 3D template fitting, in Chapter 5 shows that these criteria are indeed suitable to measure the expected performance of a design optimization by evaluation of the deformation setup. This enables an optimization of the deformation setup for performance improvement.

The discovered conflicting nature among the criteria regularity and improvement potential that is closely related to the conflict between exploration and exploitation, emphasizes the construction of Pareto optimal setups. Because we motivate RBF deformations as the methods of our choice, we analyze the optimization of the RBF center distribution in Chapter 6. We conduct and compare a multi-objective optimization with a weighted single-objective optimization for Pareto optimal solutions and evaluate the quality of the efficient heuristic, Lloyd and orthogonal least squares sampling, as well as a deterministic gradient-based approach. Our approach of combining the two heuristics results in enhanced algorithms for the construction of optimal center distributions with respect to a chosen preference between regularity and improvement potential.

In Chapter 7, we target the optimal choice among the previously constructed compromises for dynamic environments. The common assumption to focus on exploration for a strong change in the environment and to employ exploitation for static dynamics is supported by our results. We will show that a 50-50 focus between exploration and exploitation works well for noisy or imprecise information.

In the last part of the thesis, we review our work, especially the definition of regularity and variability in Chapter 8. We propose an orthogonalization approach to tune any deformation setups for optimal regularity and thereby for optimal convergence speed. We discuss the previously drawn connection to exploration (of Chapter 6). And we propose an alternative definition for variability, which is better suited to characterize exploratory capabilities than the previously proposed variability measure. The results of the previously conducted preference analyses are still valid for the new variability definition which we show in additional tests.

Finally, we summarize our findings and look at future research towards evolvability-optimal deformations in Chapter 9. In essence, we support a designer or engineer with enhanced methods to optimize linear deformation setups and thereby improve the performance of an evolutionary design optimization process. Only little information has to be provided by users to apply our automatic approaches in a black-box manner for, e.g, optimal RBF deformation setups.

2 Linear Deformations in Evolutionary Design Optimization

Biologically-inspired evolutionary optimization routines [BFM00, MS08, Wei09] are commonly applied for real-world design optimization problems like the optimization of turbine blades or wings [MOS05, LSN15, KG15], car or train models [SMB12, ZYSD16]. Evolutionary algorithms offer the required robustness and flexibility to handle noise and varying objectives (called fitness functions). Hence, in our application, automotive shape optimization, we apply an evolutionary optimization scheme, which we describe in the first section of this chapter. In particular we motivate and employ the *Covariance Matrix Adaptation - Evolution Strategy* (CMA-ES, [Han06, AH12]).

In the automotive shape optimization the design model to be optimized is typically represented by a surface polygon mesh, where the m mesh vertices $\mathbf{x}_1, \ldots, \mathbf{x}_m$ represent points on the discrete surface, which are connected by polygonal faces (usually triangles or quads). The vertex positions $\mathbf{x}_i$ could in theory be used as optimization parameters in an evolutionary optimization. However, for non-trivial models the complexity of the model easily exceeds 1 million vertices, thus making the direct optimization of vertex positions intractable.

However, even for highly complex shapes the actual shape deformations applied during optimization are rather simple, low-frequency functions, which can therefore be controlled by a small number of parameters. The representation is therefore a shape deformation operator $\mathbf{o}$, which maps deformation parameters (genotypes) to shape variations (phenotypes), which are then evaluated by the fitness function (for an overview see Figure 2.1). Deformation techniques frequently employed in design optimization are radial basis functions (RBFs [Wen04, SMB15]) and free-form deformation (FFD [SP86, HHK92]). In addition, we apply thin shell models, which are used to describe surface deformation in computational mechanics or computer graphics [BS08, BKP$^+$10]. These chosen deformations are linear in the parameters, hence we express them in matrix notation on the discrete shapes. Thus, the deformation operators result in deformation matrices **O** as representations. In the second part we therefore describe the mathematical framework of the employed linear deformations and the general matrix framework.

The deformation matrix has to be initialized before an evolutionary design optimization process starts and during the optimization it can be adapted for enhanced performance. This initialization or adaptation, respectively, is an own optimization process, which we call *setting up the deformation* to clearly distinguish this process from the design optimization process itself. Consequently, we call the deformation matrix *deformation setup*.

Note, because we evaluate methods to optimize deformation setups in test scenarios of different dimensionality we utilize a general notation here and give details when we describe the scenarios in Chapter 5. Visualized examples in this section are in $\mathbb{R}^3$ and hold for arbitrary dimension, too.

2.1 Evolutionary Design Optimization

The biologically-inspired, randomized search principles of evolutionary optimization methods induce robustness towards local optima and thereby allow a global search, such that these algorithms are often denoted as global optimization. For basic literature and general concepts we refer to [BFM00, Wei09]. The populations, which evolutionary algorithms are based on, offer the potential of tracking different individuals (sets of parameters) and thereby they enable finding equally fit but different solutions. This offers a high amount of flexibility especially in changing fitness environments. The algorithms are free of analytic constraints for the mathematical models of the fitness environments, such that they can be employed for any, even discrete, fitness function. In contrast, e.g., gradient solvers require smooth and differentiable functions. However, evolutionary algorithms are able to follow a fitness trend like a fitness gradient. Moreover, they are capable of tracking different fitness functions at the same time.

But these advantages come at a cost. The randomized search requires many fitness evaluations to recognize a fitness trend. Especially computational expensive functions, like aerodynamic simulations for determining drag performance, result in a slow optimization process. Furthermore, the randomized search lacks guarantees for convergence which makes termination criteria difficult to model. One simply does not know if a computed solution is (locally) optimal or if it can be further improved but the algorithm requires more iterations to do so. Nonetheless, the flexibility and robustness as advantages of evolutionary computation motivate its application for design optimization tasks.

In principle these algorithms follow biological motivated processes, which we describe shortly. First, the representation as the mapping between genotype (parameter space) and phenotype (design space) is set up for an initial population of typical randomly generated phenotypes. This population is initialized as the parent generation. Second, the mutation and/or crossover operator vary the genotype of each individual partly random, which results in the new offspring. Then this set of new genotypes is mapped back to a new set of phenotypes in the third step according

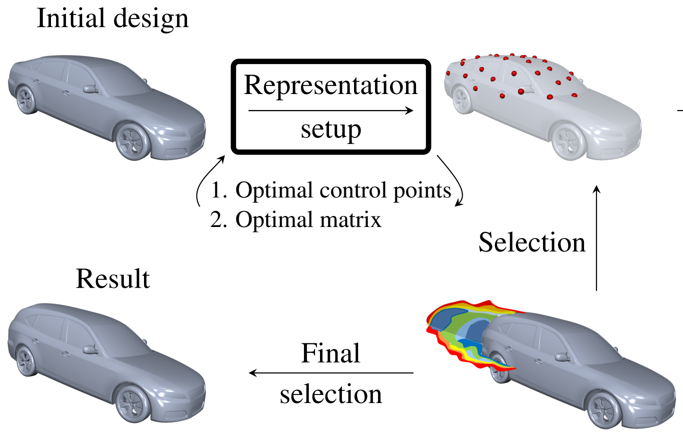


Figure 2.1: Overview of evolutionary automotive design optimization: Our focus is on the optimal initial representation setup, i.e., an optimal genotype→phenotype mapping, which is determined by the distribution of control points (e.g., the RBF kernels shown in red) and defined by the numerical properties of the resulting deformation matrix.

to the initially defined representation. Fourth, these newly generated phenotypes are evaluated by a fitness functions (e.g. aerodynamic drag simulation). Finally, the phenotypes (including or excluding the parent generation) are ranked and if a termination criterion is met an optimal result is selected. Otherwise, the optimization process continues with a possibly randomized selection of the ranked phenotypes as new parents whose genotypes are varied again by mutation and/or crossover. For a dynamic optimization process, where e.g., the fitness function may change (e.g., by inducing a change to the angle of attack in the aerodynamic drag simulation), the representation can be reinitialized according to extracted knowledge during the fitness evaluation stage. In Figure 2.1 we exemplarily sketch a design process without a dynamic interaction. As mentioned before, our focus is on the optimal setup of the representation, i.e., the optimal location of control points and optimal numerical properties of the deformation matrix of linear deformations.

Among the different algorithms, e.g., genetic algorithms, or evolution strategies (to name a few possible choices), we chose the *Covariance matrix adaptation – evolution strategy* (CMA-ES) [Han06, AH12] because of its efficient search. The algorithm is recommended especially for expensive fitness functions like an aerodynamic simulation. The reason of its success lies in the modification of the mutation and recombination operator with the covariance matrix of the population. Thereby, a population is pushed more rapidly towards a promising direction for fitness improvements. For large problems (more than 1000 parameters) or problems where specific domain knowledge can be utilized, the CMA-ES might not be the right tool [Han06, AH12].

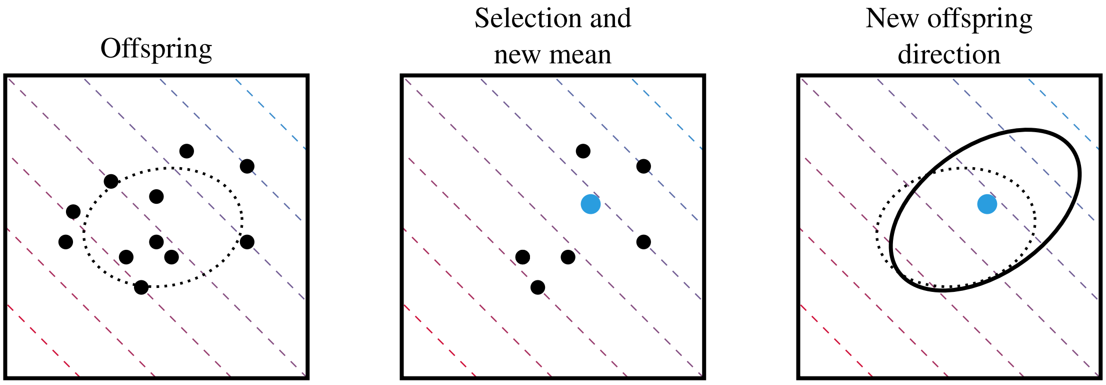


Figure 2.2: The procedure of CMA-ES: Offspring is sampled according to a covariance matrix (left). Candidates are select and their mean is computed (center). The covariance matrix is updated given the new mean and the selected candidates (right). As a consequence the fitness (depicted as the dashed lines) of the population is improved.

In Figure 2.2 we depict the principle procedure of CMA-ES for the maximization of a linear function exemplarily. The dashed lines are the function's level sets. First, offspring is generated according to a given covariance matrix (dashed ellipse) which can be seen as a mutation operator. Then, candidates are selected of this offspring. Next, a new mean with these candidates is calculated. This can be interpreted as a recombination step. Finally, the covariance matrix is updated according to the new mean and the selected candidates. This matrix is employed to sample new offspring in the next iteration. For the mathematical details and formulas, e.g., mutation and selection operators and probabilities, we refer to [Han06, AH12].

As basic parameters of $(\mu(\overset{+}{,})\lambda)$ CMA-ES a designer only has to chose the number of the parents μ, the number of generated offspring λ, and the selection scheme $(\overset{+}{,})$, where $(+)$ indicated a selection among the parents and the offspring, and (,) indicates selection among the offspring only. A typical employed operator for selection is non-elitist selection, i.e., the fittest individual might not be selected as offspring. Furthermore, the CMA-ES sets an initial step size and adapts it during an optimization run. But we set the initial step size in some examples manually for improved performance. As a realization of the CMA-ES to evaluate linear deformations in design optimization we use the shark 3.0 library [IHMG08].

Our focus is on the definition of evolvability-based quality criteria for deformation setups and their optimization. For the optimization of these setups we employ the NSGA-II (elitist Non-Dominated Sorting Genetic Algorithm-II, [DPAM02]) because this algorithm optimizes multiple criteria very efficient. The algorithm follows the typical selection–crossover–mutation scheme with typical crossover and mutation operators (elitism, binary crossover, real-parameter mutation [SD94]). The selection operator is the main difference to alternative multi-objective algorithms. We explain the operator briefly and refer to [DPAM02] for details.

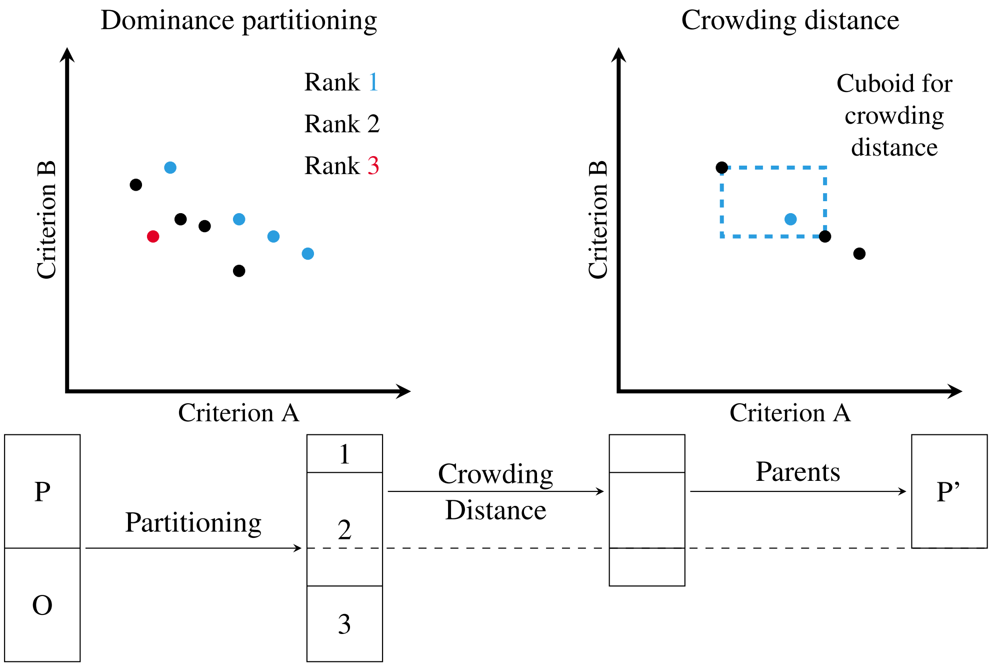


Figure 2.3: The selection scheme of NSGA-II: First, the dominance partitioning is computed for the parent (P) and offspring (O) generation. Second, the crowding distance is computed for the individuals. The new parent generation (P') is generated according the partitioning with binary tournament selection. If the individuals of a particular rank are to many for the size of the parent generation then selection according to the crowding distance is applied.

In a fist step the dominance partitioning is computed to assign a rank to each individual of the parent and offspring generation. Rank 1 is assigned to all non-dominated individuals. Then these individuals are temporarily ignored and rank 2 is assigned to the new non-dominated individuals. Further ranks are computed analogue (Figure 2.3, left).

In a second step a fitness value, i.e., the crowding distance, is computed for each individual. It is defined as the average side length of the individual's cuboid, which is bounded by its neighbors as depicted with the dashed box in Figure 2.3 (right).

Finally, binary tournament selection according to the rank is applied for the new parent generation. If a set of individuals with identical rank would exceed the size of the parent generation, tournament selection according to the crowding distance is applied for this set (Figure 2.3, bottom).

In essence we employ the CMA-ES to solve design optimization problems for the evaluation of our proposed evolvability-based quality criteria. In contrast, we use NSGA-II for the optimization of deformation setups with respect to these criteria.

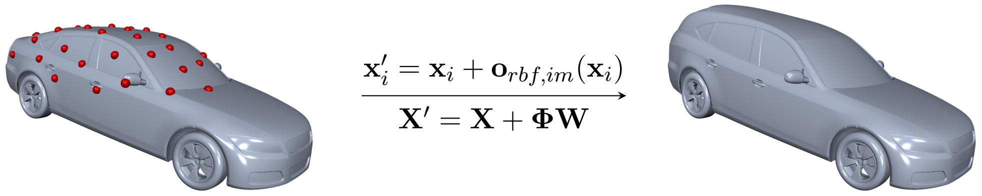

$\mathbf{x}'_i = \mathbf{x}_i + \mathbf{o}_{rbf,im}(\mathbf{x}_i)$
$\mathbf{X}' = \mathbf{X} + \mathbf{\Phi W}$

Figure 2.4: The linear RBF deformation $\mathbf{o}_{rbf}$ transforms the initial model $\mathbf{X}$ to $\mathbf{X}'$ by translating each vertex $\mathbf{x}_i$ of $\mathbf{X}$ by the displacement $\mathbf{o}_{rbf}(\mathbf{x}_i)$, which depends on chosen weights $\mathbf{W} = (\mathbf{w}_1^{\mathrm{T}}, \ldots, \mathbf{w}_n^{\mathrm{T}})^{\mathrm{T}}$. In general each weight $\mathbf{w}_j$ cannot be interpreted geometrically. We call this approach *indirect manipulation (im)*. The red dots depict the centers $\mathbf{c}_j$ of the RBFs.

2.2 Radial Basis Functions

A deformation with RBFs is a kernel-based deformation, for which radial symmetric basis functions (the kernels) are distributed at RBF centers. A designer can freely chose the amount n, the shape, and the location of the kernels which gives an enormous flexibility in contrast to other deformation methods like FFD or shell deformation.

The initial design $\mathbf{X} = (\mathbf{x}_1^{\mathrm{T}}, \ldots, \mathbf{x}_m^{\mathrm{T}})^{\mathrm{T}}$ is deformed into a shape variant $\mathbf{X}' = (\mathbf{x}_1'^{\mathrm{T}}, \ldots, \mathbf{x}_m'^{\mathrm{T}})^{\mathrm{T}}$ by adding to each $\mathbf{x}_i$ the displacement $\mathbf{o}(\mathbf{x}_i)$, which for RBFs has the form

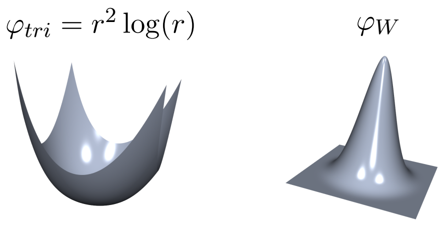
$$\mathbf{o}(\mathbf{x}) \;=\; \sum_{j=1}^{n} \varphi(\|\mathbf{c}_j - \mathbf{x}\|)\; \mathbf{w}_j \;\;=:\; \sum_{j=1}^{n} \varphi_j(\mathbf{x})\; \mathbf{w}_j\,. \tag{2.1}$$

Here, $\varphi_j(\mathbf{x}) = \varphi(\|\mathbf{c}_j - \mathbf{x}\|)$ denotes the j-th scalar-valued radial basis function, which is centered at $\mathbf{c}_j \in \mathbb{R}^{d_1}$ and weighted by $\mathbf{w}_j \in \mathbb{R}^{d_2}$ (see Figure 2.4). Hence, the deformation $\mathbf{o}$ maps $\mathbb{R}^{d_1}$ to $\mathbb{R}^{d_2}$. We analyze deformations from $\mathbb{R}^2$ to $\mathbb{R}$ and from $\mathbb{R}^3$ to $\mathbb{R}^3$ in our tests later on.

The choice of the kernel function $\varphi\colon \mathbb{R} \to \mathbb{R}$ has a significant influence on the resulting deformation and the computation complexity [SMB12]. We employ and analyze globally-supported triharmonic thin-plate splines, φ_{tri}, as well as compactly-supported Wendland functions, φ_W, with support radii s varying from rather local to more global:

$$\varphi_{tri}(r) = \begin{cases} r^2 \log(r) & \text{for 2D domains,} \\ r^3 & \text{for 3D domains.} \end{cases}$$

$$\varphi_W(r) = \begin{cases} \left(1 - \frac{r}{s}\right)^4 \left(\frac{4r}{s} + 1\right) & \text{for } r < s, \\ 0 & \text{otherwise.} \end{cases}$$



Exemplary visualization of kernels

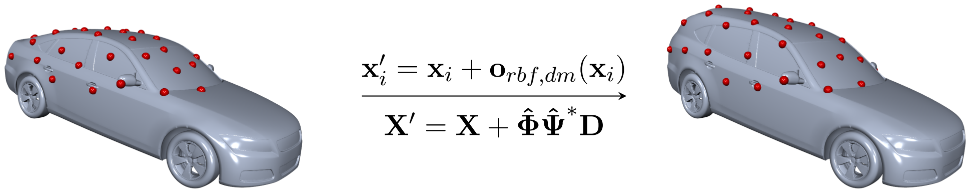

$\mathbf{x}'_i = \mathbf{x}_i + \mathbf{o}_{rbf,dm}(\mathbf{x}_i)$
$\mathbf{X}' = \mathbf{X} + \hat{\mathbf{\Phi}}\hat{\mathbf{\Psi}}^*\mathbf{D}$

Figure 2.5: The linear RBF deformation $\mathbf{o}_{rbf}$ transforms the initial model $\mathbf{X}$ to $\mathbf{X}'$ by translating each vertex $\mathbf{x}_i$ of $\mathbf{X}$ by the displacement $\mathbf{o}_{rbf}(\mathbf{x}_i)$. The RBF-weights are solved such that the deformation exactly prescribes chosen displacements $\mathbf{d}_j$ at the centers $\mathbf{c}_j$, which we call *direct manipulation (dm)*. In this setting the centers are interpreted as handles.

The RBF deformation (and thus the deformed shape) is linear in the RBF weights $\mathbf{w}_j$. If we write the initial and deformed shapes as matrices $\mathbf{X} = (\mathbf{x}_1^{\mathrm{T}}, \ldots, \mathbf{x}_m^{\mathrm{T}})^{\mathrm{T}}$ and $\mathbf{X}' = (\mathbf{x}_1'^{\mathrm{T}}, \ldots, \mathbf{x}_m'^{\mathrm{T}})^{\mathrm{T}}$, respectively, we can write the shape deformation in matrix notation

$$\mathbf{X}' = \mathbf{X} + \mathbf{\Phi}\mathbf{W} \tag{2.2}$$

using a $(m \times n)$ RBF matrix $(\mathbf{\Phi})_{i,j} = \varphi_j(\mathbf{x}_i)$ and the RBF weights $\mathbf{W} = (\mathbf{w}_1^{\mathrm{T}}, \ldots, \mathbf{w}_n^{\mathrm{T}})^{\mathrm{T}}$.

The global triharmonic kernel is a fundamental solution of the 3rd order Laplacian in $\mathbb{R}^3$ resulting in a smooth deformation at the cost of a dense deformation matrix $\mathbf{\Phi}$ [SMB14, BKP$^+$10]. In contrast the Wendland kernels lack such a physical motivation. However, they result in smooth deformations [Wen04] and they lead to an efficiently solvable sparse linear system. We utilize these two kernels as representatives for global and compact kernels, respectively.

In the above setting, the deformation $\mathbf{o}$ would be controlled by manipulating the "abstract" RBF weights $\mathbf{w}_j$, which may result in unintuitive deformations especially for global kernels employed in a manual, designer driven manipulation scenario. However, it has been shown in the context of free-form deformation that so-called *direct manipulation* is more intuitive for the human designer [HHK92] as well as more efficient in an evolutionary optimization [MOS06], due to the more direct and stronger causal relation between optimization parameters and the resulting shape deformation. In the RBF setting, a direct manipulation is controlled by distributing handles on the object's surface to allow for direct control by specifying displacements $\mathbf{d}_i \in \mathbb{R}^{d_2}$ there. For the interpolation of these displacements we place the RBF centers $\mathbf{c}_i$ on the handles, such that the following interpolation problem has to be solved for the weights:

$$\mathbf{d}_i \;=\; \mathbf{o}(\mathbf{c}_i) \;=\; \sum_{j=1}^{n} \mathbf{w}_j\, \varphi_j(\mathbf{c}_i) \;.$$

In general the solvability of this linear system is a serious problem as pointed out in [Sch07]. A positive definite deformation operator is key to guarantee solvability. For the Wendland kernels this condition is automatically fulfilled, but for the triharmonic kernel a linear polynomial π is added [Wen04, Sch07]. To allow a proper comparison of the deformation mehtods we add the linear polynomial for Wendland kernels for direct manipulation, too. The resulting additional degrees of freedom are removed by additional homogeneous equations (*) leading to the deformation

$$\begin{aligned} \mathbf{o}(\mathbf{x}) \;&= \sum_{j=1}^{n} \mathbf{w}_j \, \varphi_j(\mathbf{x}) + \sum_{k=1}^{d_1+1} \mathbf{q}_k \pi_k(\mathbf{x}) \\ &\text{subject to} \quad \mathbf{d}_i = \sum_{j=1}^{n} \mathbf{w}_j \, \varphi_j(\mathbf{c}_i) \quad \text{and} \quad \sum_{j=1}^{n} \mathbf{w}_j \, \pi_k(\mathbf{c}_j) = \mathbf{0} \, (*) \, . \end{aligned} \tag{2.3}$$

Here, $\{\pi_k\}_{k=1,\ldots,d_1+1}$ spans the space of linear (trivariate) polynomials, e.g., $\pi_1(\mathbf{x}) = 1, \pi_2(\mathbf{x}) = x, \pi_3(\mathbf{x}) = y, \pi_4(\mathbf{x}) = z$ for $\mathbf{x} = (x, y, z)$. Moreover, the polynomial guarantees linear precision of the direct manipulation approach. We could simply add this feature to indirect manipulation, too. But this would increase the number of parameters and thereby it would distort a proper comparison of both methods. Finally, for direct manipulation the following linear system has to be solved for the weights $\mathbf{w}_j$ and polynomial coefficients $\mathbf{q}_k$:

$$\begin{pmatrix} \boldsymbol{\Psi} & \boldsymbol{\pi} \\ \boldsymbol{\pi}^{\mathrm{T}} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{W} \\ \mathbf{Q} \end{pmatrix} = \begin{pmatrix} \mathbf{D} \\ \mathbf{0} \end{pmatrix} = \begin{pmatrix} \mathbf{I} \\ \mathbf{0} \end{pmatrix} \mathbf{D}, \tag{2.4}$$

with $\mathbf{D} = (\mathbf{d}_1^{\mathrm{T}}, \ldots, \mathbf{d}_n^{\mathrm{T}})^{\mathrm{T}}$, $(\boldsymbol{\Psi})_{i,j} = \varphi_j(\mathbf{c}_i)$, $(\boldsymbol{\pi})_{j,k} = \pi_k(\mathbf{c}_j)$, and the $((d_1 + 1) \times d_2)$ matrix $\mathbf{Q} = (\mathbf{q}_1^{\mathrm{T}}, \ldots, \mathbf{q}_{d_1+1}^{\mathrm{T}})^{\mathrm{T}}$. Equation (2.3) and (2.4) lead to the matrix representation of direct RBF deformation:

$$\mathbf{X}' = \mathbf{X} + \hat{\boldsymbol{\Phi}} \hat{\boldsymbol{\Psi}}^* \, \mathbf{D} \tag{2.5}$$

with

$$\hat{\boldsymbol{\Phi}} = \begin{pmatrix} \boldsymbol{\Phi} & \boldsymbol{\Pi} \end{pmatrix}, \quad (\boldsymbol{\Pi})_{i,k} = \pi_k(\mathbf{x}_i) \, , \quad \text{and} \quad \hat{\boldsymbol{\Psi}}^* = \begin{pmatrix} \boldsymbol{\Psi} & \boldsymbol{\pi} \\ \boldsymbol{\pi}^{\mathrm{T}} & \mathbf{0} \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{I} \\ \mathbf{0} \end{pmatrix} .$$

Because an interpolation problem has to be solved for the handle displacements the matrix notation and computation of *direct manipulation* (Equation (2.5)) is more complicated than *indirect manipulation* (Equation (2.2)). However, both approaches can be written in unified matrix notation because of their linearity:

$$\mathbf{X}' = \mathbf{X} + \mathbf{O} \, \mathbf{P} \, , \tag{2.6}$$

with a deformation matrix $\mathbf{O}$, being $\boldsymbol{\Phi}$ or $\hat{\boldsymbol{\Phi}} \hat{\boldsymbol{\Psi}}^*$, and deformation parameters $\mathbf{P}$, being either weights $\mathbf{W}$ or handle displacements $\mathbf{D}$. Note that the relation between indirect and direct manipulation becomes clear by omitting the polynomial term such that direct manipulation is given by $\boldsymbol{\Phi} \boldsymbol{\Psi}^{-1}$, the product between the indirect deformation matrix and the inverse interpolation matrix. In this setting $\boldsymbol{\Psi}^{-1}$ can be interpreted as a preconditioning matrix, which indeed improves the condition number of the deformation (which we show in Chapter 5).

To setup up an RBF deformation a designer simply has to chose the number of centers, their location, the kernel function and indirect or direct manipulation. This deformation offers a high amount of flexibility because the stated parameters can be chosen freely.

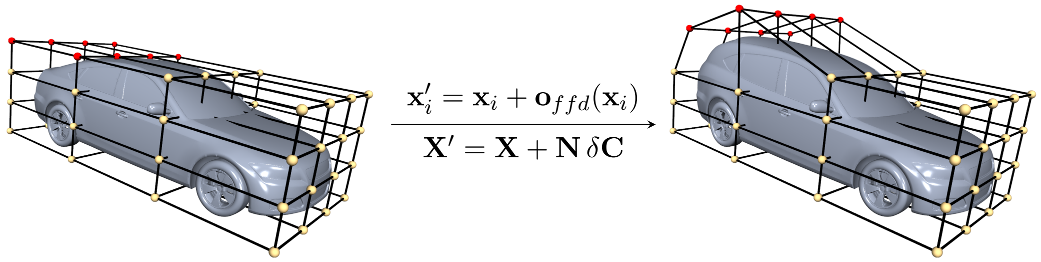

$\mathbf{x}'_i = \mathbf{x}_i + \mathbf{o}_{ffd}(\mathbf{x}_i)$
$\mathbf{X}' = \mathbf{X} + \mathbf{N}\,\delta\mathbf{C}$

Figure 2.6: For linear FFD the initial model $\mathbf{X}$ is embedded into a control grid and displacements of the grid points $\delta\mathbf{C}$ result in the deformed model $\mathbf{X}'$. Each grid point is connected to spline basis functions, which form the deformation matrix $\mathbf{N}$.

2.3 Free-Form Deformation

In contrast to kernel-based RBF deformation Free-Form Deformation (FFD) embeds the object in a lattice of $n_1 \cdot n_2 \cdot n_3$ control points (or control grid) and displacements of the control points control the deformation of the object (Figure 2.6). These displacements $\delta\mathbf{c}$ of the grid points are linked to tensor-product spline functions $N : \mathbb{R} \to \mathbb{R}$. The deformation $\mathbf{o} : \mathbb{R}^{d_1} \to \mathbb{R}^{d_2}$ is written as:

$$\mathbf{o}(\mathbf{x}) = \mathbf{o}(x, y, z) = \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \sum_{k=1}^{n_3} \delta\mathbf{c}_{ijk} N_i(u(x))\, N_j(v(y))\, N_k(w(z)) \tag{2.7}$$

with $\mathbf{u}(\mathbf{x}) = (u(x)\,, v(y)\,, w(z))$ as the local coordinates of $\mathbf{x}$ with respect to the grid and n_i as the number of control points in one particular dimension [PT96]. Note for the parametrization of 2D domains the z coordinate, its local coordinate w, the control points $\delta\mathbf{c}_{\cdot\cdot k}$ and the corresponding basis N_k are omitted (and analogue for any dimension of the domain). We define

$$\delta\mathbf{c}_l := \delta\mathbf{c}_{ijk},\;\; N_l(\mathbf{x}) := N_l(\mathbf{u}(\mathbf{x})) = N_i(u(x))\, N_j(v(x))\, N_k(w(x)) \quad \text{and} \quad n := n_1 \cdot n_2 \cdot n_3$$

to write Equation (2.7) as (equivalent to [BKP+10]):

$$\mathbf{o}(\mathbf{x}) = \sum_{l=1}^{n} \delta\mathbf{c}_l N_l(\mathbf{x}) \;.$$

Here, the linear structure of FFD becomes clear and we write the deformation in unified matrix notation (Equation (2.6)) as:

$$\mathbf{X}' = \mathbf{X} + \mathbf{N}\,\delta\mathbf{C} \tag{2.8}$$

with $\delta\mathbf{C} = (\delta\mathbf{c}_1^{\mathrm{T}}, \ldots, \delta\mathbf{c}_n^{\mathrm{T}})^{\mathrm{T}}$ and the $m \times n$ deformation matrix $(\mathbf{N})_{i,l} = N_l(\mathbf{x}_i)$.

Although we are free to chose any spline N we utilize cubic B-splines functions with a uniform knot vector because they are well established in state-of-the-art design optimization [SMB12, ZYSD16]. To evaluate the spline N at a point $\mathbf{x}_i$ we need its local coordinates $\mathbf{u}_i = \mathbf{u}(\mathbf{x}_i)$, which have to be computed before a deformation is performed. This is a non-linear problem in $\mathbf{u}_i$ for arbitrary control grids for which we minimize:

$$\min_{\mathbf{u}_i} \left\| \mathbf{x}_i - \sum_{l=1}^{n} N_l(\mathbf{u}_i)\, \mathbf{c}_l \right\|^2 .$$

We minimize these differences between the points $\mathbf{x}_i$ and their representation in the FFD system (following [MJ96]) with Newton's method [Kel03] and an implementation from [Dre17].

The number of grid points, their location, and the type of the spline basis have to be chosen for setting up FFD. This is conceptually equivalent to RBF deformation. But, in contrast FFD does not offer the flexibility that RBF deformation does. The number of grid points n is restricted by the dimensions of the grid, i.e., n can only be a product of the number of points in x,y (z) direction. Moreover, feasibility of the control grid has to be guaranteed, i.e., the grid has to be free of self-intersections. Hence, the distribution and optimization of grid points requires additional constraints for automatic procedures. Furthermore, the degree of the utilized spline is a lower bound for number of grid points in the particular dimension. E.g. cubic splines require at least 4 control points per dimension. Because typically applied grids have a higher resolution this is only a minor drawback. But, from a numerical point of view FFD is more costly than RBF. Not only does the recursive evaluation of a spline requires more effort than the evaluation of a polynomial kernel function, but also the computation of local FFD coordinates is an optimization process itself, which is not required for RBFs. However, FFD's advantage is the sparsity of the deformation matrix and the intuitive control of the deformation for manual deformation approaches.

Equivalent to RBFs there exists a direct version of FFD [HHK92] where handle displacements are the parameters for design variation. In the RBF settings we simply placed the handles at the center locations on the surface to obtain direct control. But this cannot be done for direct FFD straight forward because the control grid is not restricted to the surface of the object. We would have to specify the location and amount of handles, the location and amount of grid points, and their connection to each other. Hence, we don't evaluate and compare direct FFD.

2.4 Shell Deformations

To show the general character of our methods for optimal deformation setups we analyze a further type of deformation, namely surface deformations. RBF and FFD deform the surrounding space of an object and thereby the object itself. They are independent of the object's discretization. In contrast, surface deformations are purely defined on the surface of the object and depend on its discretization. Exemplarily we describe and analyze a shell-based deformation, which is commonly applied in computer graphics [BS08, BKP$^+$10]. Similar to direct manipulation of RBF the deformation is controlled by displacements of handles, i.e., chosen points of the discrete surface (similar to Figure 2.5). Based on these displacements the remaining vertices are deformed by minimizing physical energies. We choose the minimization of a bending energy, which results in smooth and plausible deformations. For the continuous calculus we refer

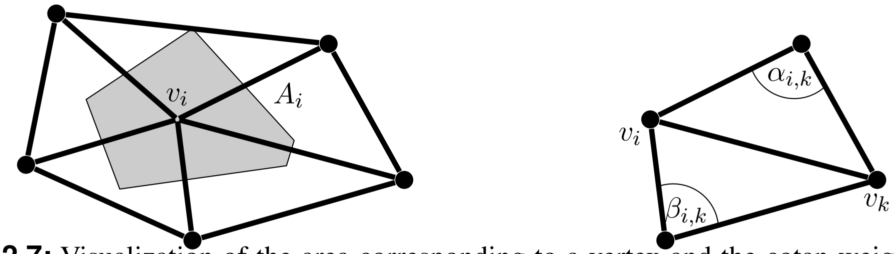


Figure 2.7: Visualization of the area corresponding to a vertex and the cotan weights for the computation of the Laplacian.

to [BS08, BKP+10] and show the discrete version only. Given the displacements $\mathbf{d}_j \in \mathbb{R}^{d_2}$ for n vertices selected as handles we compute the displacements of the remaining vertices $\mathbf{d}_i \in \mathbb{R}^{d_2}$ by solving:

$$\Delta^2 \mathbf{d}_i \overset{!}{=} \mathbf{0} \tag{2.9}$$

with the bi-Laplacian $\Delta^2 \mathbf{d}_i$, which is recursively defined by

$$\Delta^2 \mathbf{d}_i = \Delta(\Delta \mathbf{d}_i) = \Delta \left(\frac{1}{2A_i} \sum_{v_k \in \mathcal{N}(v_i)} (\cot \alpha_{i,k} + \cot \beta_{i,k}) (\mathbf{d}_k - \mathbf{d}_i) \right)$$

Here, A_i is the corresponding area of vertex v_i, $\mathcal{N}(v_i)$ denotes the neighbors of v_i, and $\cot \alpha$ and $\cot \beta$ are the cotangent weights defined as shown in Figure 2.7 [BKP+10]. For given displacements of the handles $\mathbf{D} = (\mathbf{d}_1^{\mathrm{T}}, \ldots, \mathbf{d}_n^{\mathrm{T}})^{\mathrm{T}}$ Equation (2.9) is solved for the displacement of the remaining vertices $\bar{\mathbf{D}} = (\bar{\mathbf{d}}_1^{\mathrm{T}}, \ldots, \bar{\mathbf{d}}_{m-n}^{\mathrm{T}})^{\mathrm{T}}$ which results in a linear system:

$$\begin{pmatrix} \mathbf{I} \mid \mathbf{0} \\ \mathbf{L} \end{pmatrix} \begin{pmatrix} \mathbf{D} \\ \bar{\mathbf{D}} \end{pmatrix} = \begin{pmatrix} \mathbf{D} \\ \mathbf{0} \end{pmatrix} = \begin{pmatrix} \mathbf{I} \\ \mathbf{0} \end{pmatrix} \mathbf{D}\,.$$

The entries of $\mathbf{L}$ are computed according to Equation (2.9). Defining

$$\mathbf{L}^* := \begin{pmatrix} \mathbf{I} \mid \mathbf{0} \\ \mathbf{L} \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{I} \\ \mathbf{0} \end{pmatrix}$$

leads to the unified matrix notation of the shell deformation:

$$\mathbf{X}' = \mathbf{X} + \mathbf{L}^* \, \mathbf{D} \tag{2.10}$$

equivalent to equation (2.6).

Although the shell deformation is as flexible as RBF with respect to the amount and location of the control handles, the numerical robustness of the computation of the deformation matrix $\mathbf{L}^*$ significantly depends on the mesh quality of the surface. Despite this drawback we evaluate the shell deformation to confirm the general character of our approach to measure the quality of linear deformation setups as mentioned before.

2.5 Summary

In this chapter we described our deformation-based design framework with a CMA-ES as the evolutionary algorithm of our choice. Moreover, we described our employed deformation representations which are RBFs, FFD, and shell deformation. These deformations are constituted of linear combinations of simple basis functions to express complex deformations. Their linearity allows to express them in general matrix notation. An optimization algorithm is initialized with the deformation matrix $\mathbf{O}$, which we refer to as deformation setup. The deformation parameters $\mathbf{P}$ (compare Equation (2.6)) are optimized according to a given fitness function, e.g., an aerodynamic drag simulation. The initialization of the deformation matrix/setup consists of choosing

- the amount of parameters,
- locations where these parameters influence the design by specifying for instance the RBF center, FFD control point, or RBF/shell handle distribution,
- and how parameter variation deforms the design by choosing the RBF kernel function, FFD spline basis, or the energy to be minimized for shells.

Each of these three points and additional numerical properties of the deformation matrix/setup influence the performance of an evolutionary optimization process. Thus, an optimal evolutionary design process requires an optimal setup for increased performance. To determine and improve the setup's quality automatically for an optimization run we require quality criteria. In the next chapter we motivate three quality criteria based on the concept of evolvability, which is a recommended meta-attribute in complex system engineering [MBBY06].

3 The General Concept of Evolvability

The capabilities to come up with novel and enhanced solutions for industrial design processes are restricted by "old-fashioned" classical engineering approaches. A modern design optimization requires advanced concepts like complex system engineering for high-performing designs [MBBY06]. This concept motivates the adaptation of representations during the design process with *meta-attributes* where *evolvability* is regarded as a key feature [MBBY06, UD11]. Hence, keeping the representation's evolvability high during the optimization process improves its performance.

We discuss the concept of complex system engineering first in order to motivate and characterize evolvability. Additionally, we describe key features and modelling approaches.

There is no unique definition of evolvability not even in the biological context, where the term stems from. Thus, we give an overview of different interpretations of evolvability in biology, and then bridge the gap from biological simulations to the optimization of technical problems. Note, that in [Mor17] a quite general summary of evolvability is proposed as well but without precise suggestions for applications to representations. In contrast, our interpretation leads to a model for evolvability based on three attributes: *variability*, *regularity*, and *improvement potential*. These attributes characterize the exploratory capabilities, the expected convergence speed, and the exploitation of given knowledge of the representation. Hence, these three attributes provide the basis for our mathematical model to quantify the expected performance of deformation representations in an evolutionary design optimization process.

Our interpretation of evolvability might seem too narrow at the first glance because further prominent aspects like *robustness* and *modularity* are not directly incorporated. But that is not the case, as we point out in an additional discussion of further aspects and topics related to evolvability in this chapter.

Because evolvability comprises the concepts of *exploration* and *exploitation*, which are prominent in evolutionary computation, we analyze various definitions in the end of this chapter.

This chapter is based on our publication [RBM15] and guides us to evolvability-motivated quality criteria ready to be applied to linear shape deformation operators as representations for design optimization.

3.1 Complex System Engineering

A system is a construct or collection of different elements that together produce results not obtainable by the elements alone [HFK$^+$06]. Creating and linking these elements is the task of an engineer in order to fulfill the customer's needs throughout the entire life cycle of the system.

Complexity arises through many different objectives or conflicting interests, (nonlinear) interactions of components, large design spaces, feedback loops, or adaptive processes [FS11, CJS13]. Varying fields of research are involved, such as engineering (dynamical systems and their control), computer science (modeling and simulation), biology (self-organized systems), or physics (physical models) [UD11]. Complex systems are based on simple, maybe different, but separated components. They are combined and linked together in order to achieve multiple, unpredictable, and time-varying goals. The system should be able to change the representation or even the fundamental structure. Examples are communication or transportation networks, financial markets, organisms, or insect colonies [MBBY06].

In *classical* engineering the designer gathers information to specify the problem as precisely as possible. Uncertainty is eliminated as much as possible. As a consequence many pieces of information are needed, e.g., about the conditions of the environment and the task that should be performed. This requires specialized knowledge and competence of the designer, who has to model the functionality and the overall process. The typical approach is to simplify a system as much as possible. Only if the designer tested the specialized system well enough, it is completely fixed and reproduced. This leaves no space for a later adaptation other than intentionally planned by the designer. The *classical* goal is to obtain a single specialized solution that can be reliably reproduced. The required predictability, transparency, and controllability inevitably prevents self-adaptation and thereby, reduces the chance to discover inspiring novel solutions [MBBY06].

Complex systems, in contrast, have to operate in unknown, uncertain, unforeseeable, dynamic environments. The focus is set on the adaptation potential to handle these demands. The required flexibility is gained by simple, local, and linked processes, which in concert solve the global problem. The designer models these simple processes and their connection, and thereby produces a "blueprint" [UD11]. The system is responsible for the setup of the processes and their re-evaluation and adaptation during the life time of the system (self-organization and evolution). Thus, the goal in complex system engineering is to develop a method that enables the system to autonomously interact with its environment; or as it is stated in Mina et al. [MBBY06]: "becoming" is "being".

For this approach the following two main characteristics are important: *self-organization* and *evolution* [MBBY06, FS11]. *Self-organization* is a large-scale and local organization of many simple components. The term large-scale describes the interaction of differently conceptualized components with varying complexity. Like many other concepts, self-organization is inspired by nature, e.g., in molecules many atoms form a large structure through local forces. Although human-designed systems usually (and intentionally) ignore this feature, we unconsciously use self-organization in engineering problems: Small teams are built to solve sub-problems, new links between solutions are created, or problems are adjusted to new conditions. The result is that non-trivial, large-scale optimization can be produced by simple local processes [MBBY06], thereby leading to adaptive behavior without external command.

As stated in [MBBY06] engineering complex systems requires *evolutionary processes* for the integration of self-organization. The system is designed to solve unpredictable problems on its own with little information about the environment. The appropriate connection of components has to be re-adjusted or even the functionality of components has to extend over time to accomplish the varying goals. The evolutionary selection–mutation–recombination scheme handles these demands. The random mutation and recombination may vary the components and their linkage and the fittest configurations are selected with respect to the current problem (environment) by algorithms that implement these biological concepts. We want to state that evolutionary processes are not restricted to complex systems but advantageous in general engineering approaches.

The designer has to accept uncertainty as a system feature. It should be seen as a chance to generate a variety of unexpected new solutions. Depending on additional information designers have the freedom to choose some of them. An unrestricted dimension of the design space is seen as a benefit since it increases the variation of design solutions. Hence any limitation of the design space has to be modeled carefully. A three-stage approach is proposed by Frei and Serugendo [FS11], where desired, allowed, and possible areas are specified. This induces a user-defined expectation, which cannot be set by the system itself. The engineer has to implement concepts that keep the system running within the specific area but this at the same time restricts the development of the system. Thus, a trade-off has to be found, since the system should still be able to adapt its behavior.

Modularity as well as weak linkage are further concepts. Separated (modular) components can be modeled and exchanged easily if their dependencies are limited (weak linkage). The individual components/processes have to be fully functional even under changing conditions and have to be sufficiently flexible to achieve goals varying over time. This concept is referred to as *robust optimization* [BS07]; it can be used as one design methodology in complex system engineering. Another concept is *multi-functionality*, also known as degeneracy [Whi10]: Multiple processes may perform the same task in one environment (redundancy), but work on different tasks under new conditions (flexibility).

Besides self-organization further self-*-properties may be added as characteristic properties of complex systems. Frei and Serugendo [FS11] propose self-(re)configuration (parameter adjustment over time) or self-repair (ability to correct failures). The major aspects of complex systems—modularity, simplicity and linkage of components, self-*-properties, or design space models—are examined in greater detail in [OSL11, UNS12].

Given that particular research area, the important challenges for complex system engineering are that:

- The simple components may be defined in a classical manner but their connections have to be set flexibly.
- The design space has to be modeled carefully for the (partly random) system evolution to find varying solutions.
- The focus has to be on the optimal setup instead of the “optimal” solution.
- Convergence criteria are non-trivial to model, since the system’s progress is unpredictable.

But how is evolvability involved in this process?

As the system has to reflect on itself, it requires meta-attributes to quantify the potential of its current configuration. These criteria are robustness, degeneracy, or adaptability, and they can be subsumed under the concept of evolvability. Evolvability improves or guarantees the progress of the system’s development. It classifies the behavior of the system; it quantifies the design space that can be reached by the current representation; it guarantees the performance improvement of the system. In essence, evolvability covers the survivability, the solution variety, the potential for improvement, and the evolution speed in one single meta-attribute, and therefore is a key quality criterion.

With this comprehensive quality criterion we aim at measuring, optimizing, and adapting representation setups of complex systems based on customer demands, optimization targets, or environmental restrictions. But defining and measuring evolvability is a difficult challenge. In the next section we present a comprehensive analysis of biological and technical approaches in order to collect different aspects and modeling techniques for complex system engineering.

3.2 Evolvability

New offspring, which is produced via mutation and recombination in evolution, should ideally be able to survive in the current environment and it should also be able to adapt flexibly to environmental changes since this improves the evolutionary development. In general, evolvability is meant to characterize the developmental potential or capability of individuals in the evolutionary process.

There is no unique precise definition of evolvability. In biology, where the term evolvability originates from, many different definitions have been proposed. Evolvability has been defined as the ability of a genotype to produce heritable phenotypic variation [WA96, KG98, Wag08], as the potential of a population for producing novel mutations for their use in the evolution of adaptations [Pig08], or as a quantity to explain lineages of populations in the tree of life [Bro13]. These are only some of the many different biological concepts. In general, evolvability describes the quality of biological evolution or the evolutionary capabilities of an individual or a population. It evaluates potential future benefits [KG98, Bro09]. Influencing factors are, e.g., the genotype-phenotype relation, the variety of phenotypes, or the speed of a phenotype to improve its fitness in natural (changing) environments.

Deriving from these approaches, and in agreement with Sterelny [Ste04], we understand evolvability as a combination of the three attributes:

- *regularity*, which describes the quality of an individual or a population independent of the current environment to speed up the evolutionary process,
- *variability*, which aims at a rich design space (phenotype),
- *improvement potential*, which ensures the phenotype's ability to improve in changing conditions.

Note that we replaced the term *heritability* from Sterelny [Ste04] by *regularity* in order to avoid conceptual conflicts, as heritability itself can be considered as a measure for the evolutionary potential (see the heritability vs. evolvability discussions in [Hou92, HPH11, HMK16]). In the following we discuss the concept of evolvability by analyzing the individual attributes step by step.

3.2.1 Regularity

Formulating a suitable fitness function for the development of complex systems is a difficult and cumbersome process, since one cannot incorporate every individual quality aspect into the fitness criterion. This would inflate the fitness function, increase the computational cost of its evaluation, and would thereby slow down the overall optimization tremendously. Therefore, one should try to prevent the generation of infeasible (mortal) offspring *before* the environment (i.e., the optimization) evaluates it. We understand regularity as a stability attribute that reduces this infeasibility and thereby speeds up the evolutionary process. For instance, in automotive design processes a poor distribution of e.g. RBF handles easily results in (infeasible) self-intersections. Therefore such configurations should be avoided. In the context of system engineering the restriction to feasible (regular) offspring is a limitation of the design space on the one hand, but on the other hand regularity can be interpreted as a safety guard.

The regularity of phenotypes oftentimes is not employed as an individual attribute, but rather incorporated into the attribute *robustness* (e.g. [FW06]), which we discuss later in this chapter. We explicitly emphasize and follow the approach of Sterelny [Ste04], where regularity is declared as an extra trait, but also as a part of a more complex evolvability concept. The regularity criterion has to avoid problems that do not depend on the fitness function and which cannot be handled by the variability criterion. Sterelny [Ste04] describes it as an *anti-outlaw condition*. A concrete example is given in [LM12], where control lattices for free-form deformation are constructed such that control points are well separated. This reduces the chance of flipping of control points and thereby avoids infeasible self-intersections of the deformed object.

We cannot give a more precise definition of regularity because it strongly depends on the representation and the actual optimization scenario. The designer has to define this attribute as an environmental-independent criterion. We incorporate it in our definition of evolvability in order to speed up an optimization process.

3.2.2 Variability

As we have shown before, complex systems operate in uncertain, dynamic environments. Therefore a criterion that measures and preserves the flexibility of the representation, independent of the environment, has to be incorporated in the definition of evolvability. This criterion should furthermore characterize the ability and potential to extensively explore the design space (phenotype). In biology, there are many different synonyms for this concept, such as innovation, variation/variability, or even evolvability itself, as for example in [WA96, Wag05, Wag08]. Variability describes the future potential of obtaining varying phenotypes [WA96].

Wagner's approach of analyzing RNA networks [Wag08] reveals interesting properties and limitations of variability measures. This approach can be considered as a representative for a whole class of biological approaches that are based on the concept of *neutrality* [Wag05, PM11, PMW14, WBM14, PW14, GSAL16, CWMC18]. Two genotypes are neutral if they map to the same phenotype, and they are neighboring when they are connected via a single point mutation (a mutation that changes just one parameter). The first variability definition of Wagner [Wag08] is based on the neighborhood of a genotype (local definition), while the second definition characterizes the neighborhood of all neutral genotypes of a given phenotype (global definition). Both approaches compute the diversity of the phenotype and are purely discrete. The second one even requires global information of the phenotype. For complex systems this is a major drawback, since they have to operate with little information, and therefore cannot analyze the whole parameter space or design space. In the automotive scenario both spaces are continuous, which makes the definition of a neighborhood cumbersome and imprecise.

Jin and Trommler [JT10] solve this problem by replacing single point mutations by arbitrary mutations and by measuring the ratio of phenotype diversity to genotype variation. This requires a proper definition of distance metrics for both genotypes (parameter space) and phenotypes (design space). Lehmann and Menzel [Men11, LM12] transfer this idea to a shape matching optimization using free-form deformation. Their variability criterion, defined as the ratio of phenotype variation to genotype variation, characterizes the quality of different representation setups and is used to improve the performance of the optimization. As the computation of this criterion requires global information, it is not really suitable for complex systems. Moreover, we advise against a definition of the ratio of phenotype variation to genotype variation in general because a mapping with high variability would map slight genotype variation to large phenotype variations and this contradicts the goal of numerical stability.

A possible solution can be derived from an approach called *novelty search*, which replaces the fitness selection criterion of evolutionary algorithms by a variability criterion. Example applications for maze navigation and biped robot experiments are described in [LS11a, IJH$^+$13, SN14, VC14, LWS16]. In [MLC16] the novelty search is extended to *evolvability search*. We do not discuss particular algorithmic details here, but we briefly describe the idea for deriving variability: The variability of each individual of an offspring is measured by a novelty/evolvability metric, which evaluates the dissimilarities to each other individual in the population resulting in a highly diverse population. In order to decide whether a concrete goal has been reached, one has to perform an additional objective-oriented evaluation. In a general (non-evolutionary) optimization context, this approach can be seen as the generation and the evaluation of local samples for different representation setups which can be costly [MLC16].

The ability to extensively explore the design space is measured by the variability criterion, which characterizes the genotype-phenotype mapping. We promote a definition following the basic goal to reach each possible phenotype. Since global information is not available in complex systems, one has to fall back to local methods for computing variability.

3.2.3 Improvement Potential

During the development of a complex system some parts may already be (close-to) optimal in the current state, while other parts need further adaptation to specific demands or changing conditions. For instance, in an automotive design process the roof may already be satisfactory, but the fender has to be improved with respect to drag. A representation that can only change both targets simultaneously is counterproductive. Enabling the representation to adapt to sensitive regions for fitness improvements requires to incorporate a fitness-dependent criterion into the evolvability definition.

In the biological context several approaches identify evolvability itself with adaptation potential or adaptation speed of a population to an environment, e.g. [Pig08, DPWP10], and thereby target the improvement of the fitness. But this definition is rather imprecise, since a population is called adapted as soon as a beneficial trait occurs significantly more often. In the engineering context the optimization potential in a varying fitness landscape is investigated in [Suz03, BCCV09, TM14]. In [RSM05, RM06, RM07] the structural bias of an environment is analyzed. When different regions of the fitness landscape are linked, a representation that can learn that linkage will better adapt to changes in the environment than a representation that ignores the fitness landscape. This approach, however, requires knowledge about the different environments and the connections between them. We regard this as a contradiction to unpredictability in complex system engineering. The idea to include a fitness-dependent learning process is promising though. While it may slow down the development of the system in the beginning, it improves the long-term performance. Since fitness evaluations typically are computationally expensive, computationally cheaper surrogates can be used to approximate the original fitness function and replace it in the optimization [JOS00, Jin11, LOM$^+$13]. While the surrogates are easier to evaluate by construction, they require an additional learning step.

Aulig [Aul11] selects the representation that results in the best compromise for a variety of predefined environments and defines this representation to have the highest improvement potential. However, an additional evolutionary optimization for each environment has to be performed in order to evaluate each candidate and find the compromise, which can be rather costly. This approach can be considered as performing local optimization for computing the improvement potential. Heading in similar direction, a probability-based definition is employed for improving the fitness of the offspring compared to the parent's fitness in an evolutionary optimization [SAESF17].

Transferred to complex systems the third criterion characterizes the (fitness) improvement potential of a representation. One has to classify different representation setups either in a learning process, with an additional local optimization, or even with a combination of both. Based on available information about the environment this choice is left to the designer. In the spirit of complex system engineering we can also incorporate his/her experience, to specify important/ sensitive regions for later adaptation.

3.3 Further Targets of Evolvability

In the context of evolvability many more attributes are used and can be investigated. In complex system engineering *robustness* is one important aspect, but its relation to evolvability is discussed contrarily. Another important aspect of complex systems is *modularity*. How it is integrated into our evolvability model is shown after discussing robustness. The setup of evolutionary algorithms influences the performance of the system as well. From that point of view evolvability can be defined even for algorithms, which we discuss here.

3.3.1 Robustness vs. Evolvability

In optimization scenarios a solution is considered robust if noise does not affect its quality. The setup of an algorithm is considered robust if noise on the input data still leads to the same solution. Generally, the concept of robustness is important in complex systems to induce stability. In the biological context robustness reduces the mortality of offspring and therefore promotes the evolutionary process. Wagner and colleagues define it as the persistence of an organismic trait under perturbations [FW06]. The authors analyze robustness on three different levels: The first one is independent of the environment and is called stochastic noise. Robustness preserves the general quality if stochastic fluctuations occur in (biological) systems. The second level characterizes the influence of genetic variation on the phenotype. A trait is regarded as robust if genetic variation preserves it. The third level describes the survivability of a phenotype when changes of the environment occur and thereby addresses the fittest individuals. Our definition of evolvability can be considered to include robustness by interpreting stochastic noise as one part of regularity. Moreover, robustness to varying environments is covered by the attribute improvement potential.

Whether robustness to genetic variation promotes or hinders variability is discussed contrarily. In biology, robustness and evolvability are usually reduced to the variety of phenotypes [Wag05, BLOA06, Wag08, HB10, MT10, PM11, PMW14, WBM14, PW14, Wag17]. Hence one could argue that a highly robust phenotype is not variable but this statement is not always true. Wagner [Wag08] defines robustness through the concept of neutrality in two ways: local and global (similar to the variability discussion before). Robustness quantifies the neutrality in the neighborhood of a phenotype or the size of a whole set of neutral phenotypes, respectively. Interestingly, the local definitions of robustness and variability contradict each other, while the global ones agree: A large neutral set of phenotypes has many, typically diverse neighbors, and therefore also a higher variability. This approach is questioned in [MH17] where the authors explain that Wagner's assumptions are too strict and unrealistic.

The concept of cryptic gene variation [WA11] relies on biological observations and includes both robustness and variability. When individuals are adapted to an environment their phenotype variation decreases, which reduces mortality because few non-adapted phenotypes occur. However, cryptic gene variation preserves the variation hidden in the genotypes, which promotes adaptation after a change in environmental conditions. In [DPWP10] a non-monotonic relation between robustness and variability is described, meaning that the most variable phenotype is medium robust. The authors argue that non-robust phenotypes cannot survive and very robust ones cannot evolve. In technical engineering this potential conflict is accepted, e.g., in [JGS09, JGPS09, Aul11, Men11, LM12]. The goal is to find variable solutions that are as robust as possible. The solutions on this Pareto front can be used according to the behavior of the environment. If it is stable, robust ones are preferred. If it is varying, variable ones are more promising.

The general conflict that robust phenotypes cannot adapt to new conditions is analyzed in [LP07, Whi10, DS10, WRBY12]. *Degeneracy* is proposed as the solution and is defined as multifunctionality of components. For example, two different components (e.g. proteins) may perform the same task in the current environment, but different tasks once the environment changes. The switch between redundancy and diversity, which characterizes degeneracy, increases the robustness as well as the improvement potential. If one component fails the task is performed by the redundant component, which reduces mortality. Like this, variability and improvement are ensured even under environmental changes. In [WRBY10, WRBY12] a multi-agent-system is proposed as a successful example for integrating degeneracy. The concept of degeneracy has two major drawbacks: First, multi-functional components require more resources and are more complicated, which contradicts the goal of simple components in complex systems. Second, degenerate mappings are a contradiction to the mathematically well-behaved mappings (e.g. bijections) that are typically preferred in optimization.

There exist different concepts for including robustness as a stability criterion in complex systems, either as an additional meta-attribute or in the evolvability definition. We intentionally avoid the stated conflict between robustness and variability, and include it in regularity and improvement potential, the other two aspects of the proposed evolvability definition instead.

3.3.2 Modularity

Modularity is one important feature that has to be incorporated into complex systems as it increases their performance. The articles [WA96, Pig08, HB10] link evolutionary biology with evolutionary computer science, and emphasize the importance of the genotype-phenotype mapping. Modularity is one property that can be included in this genotype-phenotype relation. In the biological context it describes the independence of functionally different regions of an individual. Independent components reduce mortality in case one component fails, and therefore induce robustness from another perspective as discussed before [KG98, DS10, DMF10]. The flexibility to replace components improves the adaptation potential. As a consequence, a high modularity promotes the development of a system, such that it can be evolved, optimized, or adapted easier and faster. We incorporate modularity into the choice of the genotype-phenotype mapping instead of defining it as an evolvability-relevant attribute. For instance, in an automotive design scenario different shape morphing methods can be used, such that the optimization may pick the most promising one for different regions of the shape.

3.3.3 Evolvability of Algorithms

The evolutionary algorithm that is used for developing and optimizing engineering systems has a strong influence on the resulting performance. For instance, if it hinders variability the system cannot change or adapt. This is investigated in the field of genetic programming, e.g., in [Alt94, BdS10, BAK13, AOCP14], or for a broader field of algorithms in [OWCK16, WLZY17]. Evolvability, regarded as the adaptation potential of a population to the environment, characterizes the mutation, the recombination, and the selection criterion of evolutionary algorithms. Since these are stochastic algorithms, a probability-based definition of evolvability is commonly used. Obtaining the probability measures (e.g., the probability that a phenotype varies) is one major problem, and methods based on additional optimization steps or local information are frequently used. Of course, algorithmic operators, such as the adaptation of mutation step size or recombination probability, repair and support the evolvability during optimization.

In the more general context—evolvable hardware—different targets of the quality criterion evolvability are investigated in [Kal00, Tor04, SKL05, SdLSA$^+$10, HT11, Rus12, Sri13, MN16, MPN17]. In these works the setup and choice of algorithms is discussed, as well as the representation setup, or different fitness strategies to achieve different optimization goals. The focus is set to these aspects, instead of to a general quality criterion. Many relevant attributes of complex systems are covered, such as the use of simple components, modularity, evolutionary methods, or varying goals. This enriches our understanding of the different attributes that can be evaluated by evolvability.

3.4 The conflict: Exploration vs. Exploitation

Because complex systems are modeled for uncertain environments the representation has to ensure flexibility (variability): It has to promote the exploration of the phenotype to increase the chance of finding good solutions. But, at some point if a promising trend is discovered the representation should exploit this knowledge for a more efficient search. The conflict between exploration and exploitation is a widely discussed issue not only in evolutionary computation [Lap10, FLS10, HTL$^+$15]. E.g. in [HTL$^+$15] exploration/exploitation is even discussed for human decisions. Thus, we address it here for a comprehensive motivation for our model of evolvability.

As described in [CXP$^+$09] identifying exploration vs. exploitation as the conflict between global to local search is too narrow. Instead, definitions are proposed based on how historical information is utilized. E.g. random search does not employ any information, hence is linked to exploration, whereas gradient descent utilizes fitness information, hence is linked to exploitation. The importance to vary the focus during an optimization is one key conclusion of [CXP$^+$09]. Transferred to evolvability, a measure refers to exploration if it is independent of information about the environment and it refers to exploitation if fitness information are utilized. In this spirit, variability as well as regularity characterize exploration whereas improvement potential covers exploitation. But, in the spirit of exploring the phenotype space, the variability criterion is better suited.

Typically, the exploration/exploitation discussion revolves about the selection, mutation, and crossover operator of evolutionary algorithms as pointed out in the survey [ČLM13]. From this survey we obtain three key issues relevant for the definition of quality measures for representations. First, there are approaches to measure exploration/exploitation on the genotype *and* on the phenotype. Considering the basic motivation of particularly exploration, i.e., to explore the phenotype, we emphasize phenotype-based approaches to measure exploration and exploitation. Second, there are many suitable different definitions, e.g., difference-/distance-/probability-based ones. The representations for design optimization are continuous maps such that probabilistic approaches or threshold-based discrete differences are not feasible. Thus, a distance-based measure for exploration/exploitation is most promising for us. Third, the focus between exploration and exploitation should change during an optimization run. But, the optimal trade-off strategies are highly problem dependent. Independent of the employed measures of exploration or exploitation, this fact lowers our expectations to find an optimal and general trade-offstrategy for different representations and design optimization scenarios.

3.5 Summary

The most frequently used approach for solving engineering tasks still is classical engineering: dividing the problem into sub-problems, simplifying and solving them [MBBY06]. After this process the solution is fixed, which prevents the adaptation to unpredictable environments. A specialized solution ready for reproduction is the primary classical goal. But, the developmental phases of modern industrial products become more and more complex due to varying manufacturing constraints, customer demands, or changing objectives. This requires feedback loops and adaptive processes in the development phase of a system, which further increases the unpredictability of system behavior. Therefore the complex system engineering approach does not target specialized solutions, but rather aims to implement a blueprint and development guidelines to let the system unfold and adapt itself while interacting with the environment.

The representation setup significantly determines the developmental capability of a system. An inflexible setup prevents the adaptation of the system to changing conditions/objectives/environments. Hence, the representation setup has to be flexible enough during the system's lifetime. This level of self-organization, led by an evolutionary process, can result in beneficial unexpected solutions. The designer is responsible for the layout of the components or the representation. In this chapter we considered *evolvability* as a quality criterion for measuring the potential of a representation for complex systems. Based on basic analyses in biology, evolvability characterizes the potential success of a population in evolution. We analyzed concepts in biology and their transfer to technical systems, and pointed out disadvantages and promising approaches. Summarizing from literature a three-stage classification is motivated.

The first aspect that evolvability has to cover is a fitness-independent quality, which we interpret as *regularity*. It is included to speed up an evolutionary process. The second aspect preserves the potential variety, and is fitness-independent, too. If we achieved a beneficial design during the design optimization we need this *variability* to react on changing environments or targets. Some design regions may be crucial for the design process. It is important that the representation promotes the improvement of these regions. Quantifying this *improvement potential* improves the performance of the design process and is a third aspect of evolvability.

We discussed robustness in this context, since it is an important feature in engineering. The basic idea is to induce stability, reduce mortality, and thereby improve the evolutionary progress. It can be analyzed on the three levels like evolvability. With respect to variability contradicting positions exist in literature. We have shown arguments supporting the assumption that robustness promotes variability and arguments against this statement. We implicitly included robustness into our evolvability concept, since we regard a regular or adaptable individual as robust.

As the interplay of exploration and exploitation is another important issue in evolutionary computation we categorized regularity, variability, and improvement potential. Clearly variability covers exploratory capabilities and improvement potential refers to the exploitation of design regions. However, following a general definition of exploration, a designer might link regularity to this attribute, too.

We are aware that the optimization algorithms for engineering complex systems influence their performance, too. In our research, however, we focus on the representation setup. We gathered the articles that we included in this chapter and categorize them in the attached Table 3.1. Short notices to the different approaches towards evolvability, their basic results, and the context of the evolvability analysis are summarized in this table. Based on the analysis in this chapter we derive mathematical models for quality criteria for linear deformation representations in evolutionary design optimization in the next chapter.

Table 3.1: Summary of various approaches for evolvability

	Definitions	Result	Context	Articles
Theoretical approaches	evolvability = variability and modularity	genotype-phenotype mapping is a key object for the representation problem; robustness, modularity, variation are important for evolvability; fitness independent definition	biological analysis, biological principles in internet evolution	[WA96, Pig08, HB10, KG98, DS10]
	evolvability = variability and adaptation	changing fitness promotes variation thus evolvability; fitness dependent definition of evolvability	evolutionary computation	[GS00, WHK15]
	evolvability = variability and heritability	heritability alone is not a good measure; evolvability definition is highly problem dependent; evolvability definition is fitness and selection criterion dependent	biological analysis of animal population studies	[Hou92, HPH11, Bro09, HMK16]
	evolvability as probabilistic model	evolvability is the probability of a future trait given the current environmental and features of the population; promoting features are (low) mutation rate and variation	biological analysis	[Bro13]
		mathematical framework for algorithm analysis; structures (e.g. specific functions) are evolvable with respect to different probability distributions	conceptual mathematical analysis	[AK14, Val07, Val12, Fel09, Tou03]
	robustness as probabilistic model	mathematical (probability) framework of robustness for networks defined	mathematical biological networks analysis	[CL11, CL13a, CL13c, CL13b]
	robustness = regularity and variation and adaptation	robustness is persistence of a trait under perturbations; perturbation are: noise, genetic variation, environmental variation	biological analysis	[FW06]
	cryptic gene variation as biological concept of robustness and adaptation	genotype variety of a phenotype promotes robustness in a stable environment and adaptation in a varying environment; cryptic gene variation depends on neutrality	biological analysis	[WA11]
	evolvability as a criterion for long term development	many attributes have to be incorporated, e.g., modularity, adaptation, testability, variability, consistency, etc.	software engineering	[BBR09]
Biological simulation	evolvability, robustness = variation on the genotype and phenotype	genotype robustness hinders genotype evolvability, phenotype robustness promotes phenotype evolvability; definitions based on neutrality; fitness independent	RNA, Transcription factor binding sides analysis, neutral networks	[Wag05, Wag08, PW14, GJLA14, CWMC18, Wag17, MH17, GSAL16]
	variation additionally defined through probability	robustness promotes this definition	gene regulatory circuits, protein simulations	[PM11, PMW14, WBM14, BLOA06]
		recombination/mutation rate analysis; setup of the algorithm highly influences evolvability	algorithm analysis	[MT10, BP03, LWK15, WLZY17]
	evolvability as product of phenotype and genotype distance	robustness integrated in definition automatically; fitness independent	transcription factors in cell growth	[JT10]
	evolvability = adaptation	robustness-evolvability relation is non-monotonic	RNA simulations	[DPWP10]
	degeneracy = multi-functionality	switch between redundancy and diversity (dependent on the environment) improves robustness and evolvability (adaptation); concept based on neutrality neutrality	simulations of a multi-agent system	[Whi10, WRBY10, WRBY12]

	Definitions	Result	Context	Articles
Algorithm setup in engineering	novelty (variation) search as selection criterion in algorithms	positive correlation between novelty search and optimization performance; sampling needed to compute variation (evolvability)	maze navigation, robot walk	[LS11b, LS11a, VC14, LS13, LWS16]
		changing environments promote evolvability	digital evolution	[OWCK16]
		combination of novelty and fitness based search is most promising	neural networks in maze navigation, pattern guessing, robot walk	[IJH+13]
	evolvability search as an extension of novelty search	improves novelty search	maze navigation, robot walk	[MLC16]
	evolvability as number of successful solved problems	novelty search worse than fitness search when target changes	maze navigation	[SN14]
	evolvability = performance of algorithm	modularity of genotype-phenotype mapping increases evolvability	maze navigation, neural networks	[DMF10]
		gradient information used to improve individuals after recombination; fitness dependent	genetic programming, gene expression programming	[BAK13, BdS10, Alt94]
		genotype size, choice and setup of algorithms investigated with respect to fast fitness improvement	evolvable hardware	[Sri13, SKL05, SdLSA+10, Kal00, Tor04, Rus12, HT11]
		design of evolution strategies investigated and improved for fast fitness improvement	evolving digital circuits	[MN16, MPN17]
Evolvability for representations	evolvability = robustness and variation (and heritability)	increasing evolvability promotes optimization; variation and robustness negatively correlated; evolvability as trade-off; heritability gained through control volume setup; fitness independent	continuous free-from deformation, discrete Boolean functions	[Men11, LM12, JGS09, JGPS09]
	evolvability = adaptation potential	evolvable setups superior to robust ones regarding the adaptation to new fitness environments; compromise between fitness dependent and independent definition most successful	spline matching, pattern guessing, neural network learning, sting writing grammar, hexapod simulation	[Aul11, RSM05, RM06, RM07, BCCV09, Suz03, KVK+14, TM14]
		redundancy promotes evolvability	grammar evolution	[LP07]
		predicting the fitness development improves evolvability and optimization performance; model based on probability measures used for a search operator	car optimization, simulation on test functions	[LOM+13, AOCP14, WLZY17]

4 Definition of Evolvability Criteria for Linear Deformations

As we described in Chapter 2 our employed representations for evolutionary design optimization are linear, hence can be expressed in matrix notation. In this chapter, we transfer the biological motivations derived from the concepts of evolvability in Chapter 3 to models for the quantification of the performance of these deformation matrices. To this end, we propose mathematical formulations for the three evolvability criteria *variability*, *regularity*, and *improvement potential*. These criteria are designed to depend on the deformation matrix $\mathbf{O}$ only, and should therefore generalize to any linear deformation. To simplify the notation and derivation, we assume that the displacement function $\mathbf{o}(\mathbf{x})$ is *scalar-valued*, such that the coefficients $p_j \in \mathbb{R}$, hence $\mathbf{p} = (p_1, \ldots, p_n) \in \mathbb{R}^n$, and consequently $\mathbf{o} = \mathbf{O}\mathbf{p} \in \mathbb{R}^m$. Since the deformation matrix $\mathbf{O}$ is identical for the 1D and 3D deformation, this simplification does not change the resulting formulation of variability and regularity. However, we propose two versions for the criterion improvement potential because it depends on the dimension.

As a result of this chapter we propose three simple formulas for variability, regularity, and improvement potential, which are also published in [RAMB16]. The mathematical motivations and formulations were developed by the author of the thesis. The second author, J. Achenbach, provided an efficient algorithm for test scenarios analyzed in Chapter 5.

4.1 Variability

Variability is meant to quantify the potential for exploring the phenotype space—independent of the possibly complex objective function—by varying the genotype parameters $\mathbf{p}$ and mapping them to phenotype variations $\mathbf{o} = \mathbf{O}\mathbf{p}$. The biological concepts mainly evaluate the ratio between phenotype to genotype variation [Wag08, PMW14], which could be interpreted as

$\|\mathbf{Op}\| \,/\, \|\mathbf{p}\|$ in our setting. Other approaches, e.g., [LS11a, SN14], analyze the genotype neighborhood (specified by "ϵ") and consider the distance between the corresponding phenotypes as a measure for variability, which we would interpret as $\|\mathbf{Op}_1 - \mathbf{Op}_2\|$ for $\|\mathbf{p}_1 - \mathbf{p}_2\| < \epsilon$. Both approaches would regard a genotype–phenotype mapping with a large gradient as a good choice, which is known to cause numerical problems. In the context of free-form deformation in [LM12] a variability definition is based on global information, which is derived from the analysis of the whole design space. This was possible in the configuration, since the design space was bounded by a box around the initial design, but the concept is not suitable for problems with unbounded parameter spaces—which we aim at. These limitations motivate the need for an improved variability measure.

A representation has maximal variability if it can control the "1D-displacement" $o(\mathbf{x}_i)$ for each vertex $\mathbf{x}_i \in \mathbf{X}$ independently. This, however, would require an intractable number $m = n$ of optimization parameters, and therefore a much smaller number $n \ll m$ is typically chosen in practice. Consequently, not every desired shape variation $\bar{\mathbf{o}} = (\bar{o}_1, \ldots, \bar{o}_m)$ can be represented as $\mathbf{Op}$. Hence, the variability criterion will estimate how well a given arbitrary displacement $\bar{\mathbf{o}}$ can be approximated as $\mathbf{Op}$, by averaging the approximation error $\|\bar{\mathbf{o}} - \mathbf{Op}\|$ over all possible deformations $\bar{\mathbf{o}}$.

For a given deformation $\bar{\mathbf{o}}$, the optimal parameters $\mathbf{p}$, corresponding to the least squares approximation, can be computed through the normal equations [GVL12]:

$$\min_{\mathbf{p}} \|\mathbf{Op} - \bar{\mathbf{o}}\|^2 \quad \Leftrightarrow \quad \mathbf{O}^{\mathrm{T}}\mathbf{Op} = \mathbf{O}^{\mathrm{T}}\bar{\mathbf{o}} \quad \Leftrightarrow \quad \mathbf{p} = \mathbf{O}^{+}\bar{\mathbf{o}} \tag{4.1}$$

with $\mathbf{O}^{+} = \left(\mathbf{O}^{\mathrm{T}}\mathbf{O}\right)^{-1}\mathbf{O}^{\mathrm{T}}$ being the pseudo-inverse of $\mathbf{O}$. The best-approximating deformation therefore is $\mathbf{OO}^{+}\bar{\mathbf{o}}$, and its least squares approximation error is

$$\left\|\bar{\mathbf{o}} - \mathbf{OO}^{+}\bar{\mathbf{o}}\right\|^2 \;=\; \left\|\left(\mathbf{I} - \mathbf{OO}^{+}\right)\bar{\mathbf{o}}\right\|^2 . \tag{4.2}$$

With the thin singular value decomposition (thin SVD) $\mathbf{O} = \mathbf{U\Sigma V}^{\mathrm{T}}$ (with $\mathbf{\Sigma} \in \mathbb{R}^{n\times n}$ and the orthogonal matrices $\mathbf{U} \in \mathbb{R}^{m\times n}, \mathbf{V} \in \mathbb{R}^{n\times n}$) and the pseudo-inverse $\mathbf{O}^{+} = \mathbf{V\Sigma}^{-1}\mathbf{U}^{\mathrm{T}}$ we write equation (4.2) as:

$$\left\|\left(\mathbf{I} - \mathbf{OO}^{+}\right)\bar{\mathbf{o}}\right\|^2 \;=\; \left\|\left(\mathbf{I} - \mathbf{UU}^{\mathrm{T}}\right)\bar{\mathbf{o}}\right\|^2 . \tag{4.3}$$

To analyze how well the design space can be explored through variations of the initial design $\mathbf{X}$, we choose a uniformly distributed variation $\bar{\mathbf{o}}$ of unit length and compute the expected approximation error $\mathbb{E}$ of (4.3). We employing the $\mathrm{trace}\,(\cdot)$ of a matrix and the covaricance matrix $\mathrm{Cov}(\cdot)$ in the following calculus.

$$
\begin{aligned}
&\mathbb{E}\Big(\|(\mathbf{I}-\mathbf{U}\mathbf{U}^{\mathrm{T}})\,\bar{\mathbf{o}}\|^{2}\Big)\\
&= \mathbb{E}\big(\bar{\mathbf{o}}^{\mathrm{T}}\,(\mathbf{I}-\mathbf{U}\mathbf{U}^{\mathrm{T}})^{\mathrm{T}}\,(\mathbf{I}-\mathbf{U}\mathbf{U}^{\mathrm{T}})\,\bar{\mathbf{o}}\big) \;=\; \mathbb{E}(\bar{\mathbf{o}}^{\mathrm{T}}\,(\mathbf{I}-\mathbf{U}\mathbf{U}^{\mathrm{T}})\,\bar{\mathbf{o}})\\
&= \mathrm{trace}\,(\mathbb{E}(\bar{\mathbf{o}}^{\mathrm{T}}\,(\mathbf{I}-\mathbf{U}\mathbf{U}^{\mathrm{T}})\,\bar{\mathbf{o}})) \;=\; \mathbb{E}(\mathrm{trace}\,(\bar{\mathbf{o}}^{\mathrm{T}}\,(\mathbf{I}-\mathbf{U}\mathbf{U}^{\mathrm{T}})\,\bar{\mathbf{o}}))\\
&= \mathbb{E}(\mathrm{trace}\,((\mathbf{I}-\mathbf{U}\mathbf{U}^{\mathrm{T}})\,\bar{\mathbf{o}}\bar{\mathbf{o}}^{\mathrm{T}}))\\
&= \mathrm{trace}\,((\mathbf{I}-\mathbf{U}\mathbf{U}^{\mathrm{T}})\,\mathbb{E}(\bar{\mathbf{o}}\bar{\mathbf{o}}^{\mathrm{T}}))\\
&= \mathrm{trace}\,((\mathbf{I}-\mathbf{U}\mathbf{U}^{\mathrm{T}})\,(\mathrm{Cov}(\bar{\mathbf{o}})+\mathbb{E}(\bar{\mathbf{o}})\,\mathbb{E}(\bar{\mathbf{o}}^{\mathrm{T}})))\\
&= \mathrm{trace}\left((\mathbf{I}-\mathbf{U}\mathbf{U}^{\mathrm{T}})\,\mathbf{I}\frac{1}{m}\right) \;=\; \frac{1}{m}\mathrm{trace}\,(\mathbf{I}-\mathbf{U}\mathbf{U}^{\mathrm{T}})\\
&= \frac{1}{m}\,(m-\mathrm{trace}\,(\mathbf{U}\mathbf{U}^{\mathrm{T}})) \;=\; \frac{1}{m}\,(m-\mathrm{trace}\,(\mathbf{U}^{\mathrm{T}}\mathbf{U}))\\
&= \frac{1}{m}\,(m-\mathrm{trace}\,(\mathbf{I}_n))\\
&= \frac{m-n}{m}
\end{aligned}
$$

Because n is the rank of the deformation operator $\mathbf{O}$ we define the variability V based on the rank of $\mathbf{O}$, but normalize it by phenotype dimension in order to scale the values to $[0,1]$:

$$
V(\mathbf{O}) \;:=\; \frac{\mathrm{rank}\,(\mathbf{O})}{m}\,. \tag{4.4}
$$

According to this definition, a representation with good variability should have maximum rank, i.e., all optimization parameters $\mathbf{p}$ should be truly independent. The maximum theoretical variability $V(\mathbf{O})=1$ is achieved for $m=n$, but typically $V(\mathbf{O})\ll 1$ due to $m\gg n$.

4.2 Regularity

Regularity is understood as a fitness-independent quality criterion and is interpreted in different ways. It can be regarded as an anti-outlaw condition [Ste04] to prevent infeasible offspring. In design optimization self-intersections are such infeasibilities, which are reduced in [LM12]. Although regularity is not quantified in this article, it is implicitly optimized for a single point of interest in a FFD control volume. Preventing or reducing infeasible designs (before costly fitness evaluation) speeds up the optimization process in general. Inspired by this property, we directly interpret regularity as a criterion to measure the optimization/convergence speed. Typically, optimization algorithms try to identify a trend in the fitness landscape, a direction in which the fitness improves the most. This is significantly influenced by the representation. A local (strong causal) representation maps small changes in the genotype to small phenotype variations, thereby preserving the local neighborhood structure, resulting in a numerically stable mapping. This makes it easier for the optimization to discover a promising direction in the fitness landscape, in contrast to non-local (weak causal) representations. Hence, causality [WCT12] and locality [Rot06] are two concepts that are related to the convergence speed of evolutionary algorithms. But they are typically associated to the mutation operator (e.g., [SKVS97, GLMOB11, SMC12, TR14]), and are defined for discrete genotypes and phenotypes only. We propose a definition of regularity that generalizes these concepts to linear deformations in continuous design optimization problems.

In essence, regularity should characterize the numerical stability of the deformation process to address the expected speed of convergence. A function is considered numerically stable if a small (relative) change in the input leads to a small (relative) change in the output [TDB97]. We want to characterize the stability of the mapping $\mathbf{p} \mapsto \mathbf{O}\mathbf{p}$, i.e., we analyze the variation of the displacement $\mathbf{o} = \mathbf{O}\mathbf{p}$ due to a change in the parameters $\mathbf{p}$. In a numerically stable deformation representation each deformation parameter p_j should have about the same influence on the resulting deformation $\mathbf{O}\mathbf{p}$. More formally, a change in genotype $\mathbf{p} \mapsto \mathbf{p} + \boldsymbol{\delta}$ leads to a corresponding change in phenotype $\mathbf{O}(\mathbf{p} + \boldsymbol{\delta}) - \mathbf{O}\mathbf{p} = \mathbf{O}\boldsymbol{\delta}$. For a stable representation the *amount* of phenotype change $\|\mathbf{O}\boldsymbol{\delta}\|$ should depend on the *amount* of genotype change $\|\boldsymbol{\delta}\|$ only—and not on the direction of $\boldsymbol{\delta}$ in genotype space.

In our case of a linear function, the numerical stability is measured by the condition number of the involved matrix:

$$\kappa(\mathbf{O}) = \frac{\sigma_{\max}}{\sigma_{\min}}\,,$$

where $\sigma_{\max}$ and $\sigma_{\min}$ denote the smallest and largest singular value of the matrix $\mathbf{O}$. In fact, the condition number measures the ratio of maximum to minimum phenotype variation for unit-norm genotype changes: $\|\mathbf{O}\boldsymbol{\delta}_{\max}\| \,/\, \|\mathbf{O}\boldsymbol{\delta}_{\min}\|$, where $\boldsymbol{\delta}_{\max}$ and $\boldsymbol{\delta}_{\min}$ correspond to the maximum/minimum right singular vectors of the matrix $\mathbf{O}$ [GVL12].

It might be tempting to incorporate the fitness function f into the regularity criterion, for instance by analyzing the numerical stability of the mapping $\mathbf{p} \mapsto f(\mathbf{x} + \mathbf{O}\mathbf{p})$, but we cannot assume analytic knowledge of the fitness function. It is known, however, that basic evolutionary optimization perform better if all genotype parameters have a similar influence on the phenotype variation. A few dominant parameters might slow down the optimization or otherwise require sophisticated adaptation techniques.

As mentioned before, *locality* characterizes the convergence speed in evolutionary computation [WCT12, TR14]. Rothlauf proposes a definition of locality for discrete genotypes/phenotypes in [Rot06]:

$$\sum_{d^g_{x,y}=d^g_{\min}} \left| d^p_{x,y} - d^p_{\min} \right|\,,$$

where $d^g_{x,y}$ defines the distance between the discrete genotypes x and y, $d^p_{x,y}$ the distance between their discrete phenotypes, and $d^g_{\min}$ and $d^p_{\min}$ the minimal values of these distances. Assuming genotype variations of equal amount ($d^g_{x,y} = d^g_{\min}$) the optimal/minimal locality is achieved if the corresponding phenotype variations also have the same amount ($d^p_{x,y} = d^p_{\min}$).

Using the condition number as a regularity criterion can therefore also be understood as a generalization of the concept of locality from *discrete* genotypes/phenotypes to *continuous* optimization.

Thus, we define regularity based on the condition number, but use the inverse in order to bound the values to $[0, 1]$, with 0 being the worst value and 1 the optimal value:

$$R(\mathbf{O}) \;:=\; \frac{1}{\kappa(\mathbf{O})} = \frac{\sigma_{min}}{\sigma_{max}}\,. \tag{4.5}$$

4.3 Improvement Potential

While a high variability allows the optimization to eventually find beneficial shape variations, and a high regularity suggests that it does so rather efficiently, both criteria cannot guarantee that the optimization performs well for the specific problem at hand—since both variability and regularity are agnostic of the fitness function. In the biological context several approaches identify evolvability itself with adaptation potential or adaptation speed of a population to an environment, e.g. [Pig08]. But this definition is rather imprecise, since a population is called adapted as soon as beneficial traits occur more often. Referring to technical optimization problems, where the goal is to improve an initial solution, the third aspect of evolvability is *improvement potential*. This criterion is investigated for optimization in varying [BCCV09, TM14] as well as in static [RM07] fitness landscapes. Estimating the improvement potential of a representation is difficult due to, first, a lack of knowledge of the technical application, where information about the fitness landscape may not be available at the beginning of an optimization. Second, it can be computationally expensive, especially for industrial applications, if the local improvement potential is tested during the optimization by additional data sampling [Aul11] and (surrogate) modeling steps [MT10]. These drawbacks motivate the need for an alternative approach to determine the improvement potential in design optimization.

In an optimization process some regions of the phenotype might already be close to optimal, while other parts still have to be improved further. A successful representation should then allow for and promote these particular required shape variations. Analyzing whether the representation allows the optimizer to push the design towards beneficial configurations (larger fitness value) requires knowledge of ascent directions in genotype space.

The direction that locally improves the fitness function the most is its gradient with respect to phenotype parameters, i.e., $\nabla_{\mathbf{x}} f(\mathbf{x}) := \partial f(\mathbf{x})/\partial \mathbf{x}$. Since the analytic fitness gradient is not known in most cases, we assume that at least an approximate gradient direction $\mathbf{g}$ is available, for instance through learning from previous optimization runs or adjoint optimization approaches.

Given the improvement direction $\mathbf{g}$, we measure how well the representation can approximate it as $\mathbf{Op}$. We proceed similar to Chapter 4.1, assume $\|\mathbf{g}\| = 1$, and find the least squares approximation error to be $\|(\mathbf{I} - \mathbf{OO}^{+})\,\mathbf{g}\|^2$. Since $\mathbf{g}$ is normalized, this error is bounded from above by 1. To have the criterion in the intuitive range $[0, 1]$, with 1 being the optimal value, we define the improvement potential $P(\mathbf{O})$ (for a given unit-norm improvement direction $\mathbf{g}$) as

$$P(\mathbf{O}) \;:=\; 1 - \left\| \left(\mathbf{I} - \mathbf{O}\mathbf{O}^{+} \right) \mathbf{g} \right\|^{2} . \tag{4.6}$$

The straightforward generalization to vector-valued deformation functions with an approximate gradient matrix $\mathbf{G}$, the Frobenius norm $\|\cdot\|_F$, and$\|\mathbf{G}\|_F = 1$ is

$$P(\mathbf{O}) \;:=\; 1 - \left\| \left(\mathbf{I} - \mathbf{O}\mathbf{O}^{+} \right) \mathbf{G} \right\|_F^{2} . \tag{4.7}$$

4.4 Summary

Based on the concept of evolvability we proposed three mathematical models for variability, regularity, and improvement potential for linear deformations or deformation matrices, respectively, as our employed representations.

Variability is basically defined as the rank of the employed deformation matrix and regularity as its condition number. Both measures are independent of the fitness function and can be linked to the exploratory capabilities of the deformation setup. In contrast, improvement potential employs fitness knowledge by measuring the approximation quality of a given estimated fitness gradient. All three criteria are easy to compute and don't require any further fitness evaluations, which might be costly in real world application like automotive product design.

After having defined quantitative formulations for the three evolvability criteria variability, regularity, and improvement potential, we evaluate how well these criteria predict the true quality and performance of different representation setups in two test scenarios in the following chapter.

5 Evaluation of the Evolvability Criteria

In the previous chapter we proposed formulas for the criteria variability, regularity, and improvement potential, for the evaluation of the performance of linear deformation setups in automotive design optimization. But empirical evaluation of these criteria within an automotive design optimization scenario is impractical, since (1) the complicated fitness function inevitably requires a computationally expensive and thus slow optimization process, and (2) no ground truth in the form of an analytically known global optimum of the fitness function exists. Thus, we evaluate the definition of variability, regularity, and improvement potential in two simpler design optimization test scenarios, which are 1D function approximation and 3D template fitting. We perform our experiments on a customer computer Intel Xeon, 8x3.60GHz, with 8Gb memory.

The general evaluation process is as follows: First, random deformation setups are generated. Second, the quality of the setups is computed. Third, the design is optimized for the different setups. Finally, the correlation between quality of the setups (variability, regularity, and improvement potential) and the performance of the optimization (convergence speed and fitting error) is computed. Ideally, our analysis reveals strong correlations for the quality criteria. En passant we compare the different deformation methods (RBF, direct and indirect manipulation, FFD, and shells) to each other to get deeper inside into their advantages and drawbacks.

We start the discussion of the simpler 1D scenario after which we analyze the more complex 3D scenario like in our publication [RAMB16]. The author of this thesis designed the experiments, implemented the CMA-ES, and interpreted the results. For a more efficient analysis a deterministic optimization approach by J. Achenbach, the second author of [RAMB16], was employed to generate ground truth solutions for the evaluation of the evolutionary ones.

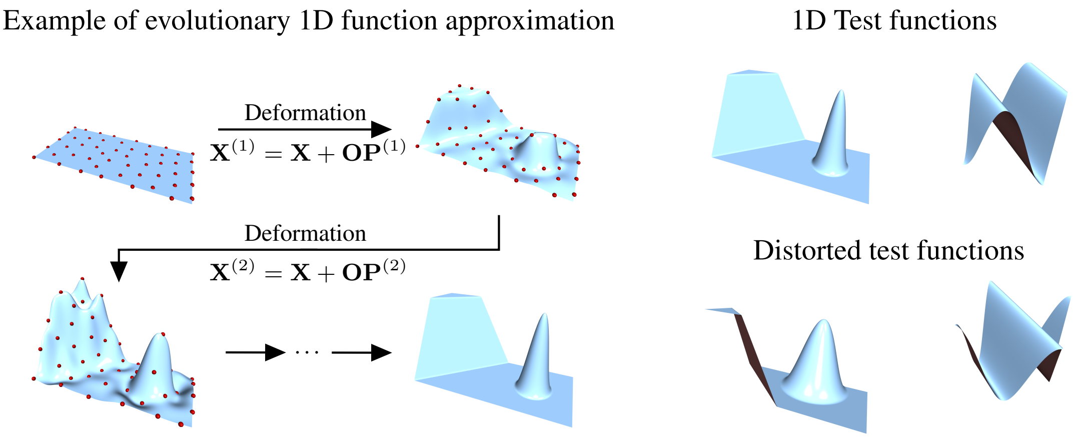

Example of evolutionary 1D function approximation
Deformation
$\mathbf{X}^{(1)} = \mathbf{X} + \mathbf{OP}^{(1)}$
Deformation
$\mathbf{X}^{(2)} = \mathbf{X} + \mathbf{OP}^{(2)}$
1D Test functions
Distorted test functions

Figure 5.1: The 1D function approximation scenario. Left: The initial plane $\mathbf{X}$ is deformed via a linear deformation $\mathbf{OP}$. E.g., the handle displacements $\mathbf{P}^{(1)}, \mathbf{P}^{(2)}, \ldots$ (red dots) of direct RBF manipulation or shell deformation result in the deformed meshes $\mathbf{X}^{(1)}, \mathbf{X}^{(2)}, \ldots$ aiming to improve the fitting accuracy to a target height field. Right: Our two test functions, the Giannelli function from [GJS12] (left one) and a sine wave (right one) as well as distorted variants (below).

5.1 Test Scenario: 1D Function Approximation

As a first evaluation scenario we have chosen a least-squares function approximation problem. Starting from a plane discretized by a regular grid of 150×150 vertices $\mathbf{x}_i = (x_i, y_i)^{\mathrm{T}}$, we use a deformation function $o\colon \mathbb{R}^2 \to \mathbb{R}$ to approximate a given scalar height field, see Figure 5.1, left. The two test functions to be approximated are a simple sine wave

$$s(x,y) \;=\; \sin(\pi \cdot (x+y))^2$$

with $(x,y) \in [0,1]^2$ and a more complex function used by Giannelli et al. [GJS12]:

$$s(x,y) \;=\; \begin{cases} 1 & y - x \geq 0.5\,, \\ 2(y-x) & 0 < y - x < 0.5\,, \\ 0.5\cos(4\pi \cdot q^{0.5}) + 0.5 & q < \frac{1}{16}\,, \\ 0 & \text{otherwise}\,, \end{cases}$$

with $(x,y) \in [0,2] \times [0,1]$ and $q = (x-1.5)^2 + (y-0.5)^2$, see Figure 5.1, right. We discretize the target functions using the same 150×150 points $\mathbf{x}_i$ and define scalar height values $s_i := s(\mathbf{x}_i)$.

The fitness function $f(\mathbf{p})$ measures the least squares approximation error between the target values s_i and the current "design" $o_{\mathbf{p}}(\mathbf{x}_i)$, and has to be minimized with respect to the deformation parameters $\mathbf{p} = (p_1, \ldots, p_n)$:

$$f(\mathbf{p}) \;=\; \sum_{i=1}^{m} \left(o_{\mathbf{p}}(\mathbf{x}_i) - s_i\right)^2 \;=\; \|\mathbf{Op} - \mathbf{s}\|^2 \;\to \min\,, \tag{5.1}$$

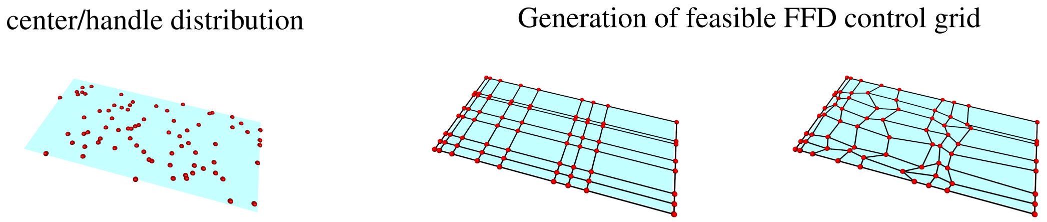

center/handle distribution
Generation of feasible FFD control grid

Figure 5.2: Example of random distributions of control points. Centers for RBF or handles for shell deformations (left) are much simpler to distribute than random FFD control points. For FFD we first distribute random points on the boundary of the bounding box of a design (middle) and than randomize the inner points (right).

with $\mathbf{s} = (s_1, \ldots, s_m)^{\mathrm{T}}$. The deformation matrix $\mathbf{O}$ is defined according to either the indirect or direct RBF method, FFD, or shell deformation, as shown in Equation (2.2), (2.5), (2.8), and (2.10), respectively.

In this particularly simple optimization scenario the global optimum can directly be computed equivalent to Equation (4.1), leading to $\mathbf{p}_{opt} = \mathbf{O}^{+}\mathbf{s}$ and $\mathbf{o}_{opt} = \mathbf{O}\mathbf{O}^{+}\mathbf{s}$. We will use this solution as a reference when evaluating the evolvability criteria, since this allows us to compare the evolutionary solution to the analytic one.

In order to experimentally evaluate the evolvability criteria defined in Chapter 4, we generate a large variation of RBF setups using different kernel types, different numbers of kernels, different support radii, and direct or indirect manipulation. Additionally, we analyze FFD and shell deformation to underline the generalization of our quality criteria for linear deformations.

To test the criteria regularity and the improvement potential we randomly generate 100 different center distributions for each Wendland and triharmonic kernel function, where each distribution consists of 25 centers. For the variability analysis we expand the tests to 50, 75, and 100 centers per setup. In the case of the compact Wendland kernels we set the support radii s (identical for all centers), such that each point $\mathbf{x}_i$ of the design mesh is overlapped (and hence can be varied) by at least l RBF kernels. We chose l to be 2, 5, and 15, such that we can distinguish between more local ($l = 2$) and more global ($l = 15$) setups. We denote the triharmonic kernel with *Tri*, the different Wendland kernels with W_l, and we use the abbreviations *im*/*dm* for indirect or direct manipulation in the plots and tables later on. For shell deformation the random distribution of handles is as simple as distributing RBF centers (Figure 5.2 left).

But for FFD the construction of randomized control grids without self-intersections is not straightforward. First, we construct randomized rectangular grid cells by placing the grid points partly random on the bounding box of the initial plane (as depicted in Figure 5.2 center). Second, the inner points are moved in a random direction at most 40% of the distance to the nearest neighbor (Figure 5.2 right). This guarantees randomized grids without self-intersections. For the test of regularity and improvement potential we utilize a 5×5 grid and for the variability analysis we expand the tests to 7×7, 9×8, and 10×10 grids.

Given a random set of centers/handles/grid points we then analyze how well our evolvability criteria predict the actual performance of the design optimization. We compute the Spearman correlation [Dan90] (using R [R C15]) between the variability/regularity/improvement potential scores of the deformation setups and the quality/convergence speed of the design optimizations. This stochastic test analyzes the monotonic correlation between two quantities and calculates its significance (p-value). The intervals $[0, 0.2[$, $[0.2, 0.4[$, $[0.4, 0.6[$, $[0.6, 0.8[$, and $[0.8, 1]$ are classified as very weak, weak, moderate, strong, and very strong correlation [Wei15]. We interpret p-values smaller than 0.01 as significant and report the p-values of the non-significant test. Note that our evolvability criteria should be maximized (with 1 being the optimal value), while the properties they characterize (e.g., fitting error, numbers of iterations) should be minimized. Hence, large negative correlations are better, but in order to simplify interpreting the numbers using the above intervals, we negate all reported correlation coefficients, such that 1 is the optimal value.

5.1.1 Results: Variability

Variability characterizes the potential for design space exploration, and is measured as the (normalized) rank of the deformation matrix $\mathbf{O}$, see Equation (4.4). Deformation setups with a high variability are expected to result in more accurate fits.

In all our experiments the deformation matrices generated from random control point setups have full rank n, such that the variability depends on the numbers of kernels, handles, or grid points, but not on the type of basis functions and their placement (as long as the control points do not coincide). We therefore increase the variability by adding more control points. Since we characterize the *potential* for accurate fits, instead of the actual result of an evolutionary process, we compute the optimal fitting errors using the analytic solution and thereby rule out any negative effects of a randomized search.

The table in Figure 5.3 shows the correlation between fitting accuracy and variability, which is significant for all tested functions. The triharmonic kernel and the widest Wendland kernel (W_{15}) show very strong correlations for both test functions. In contrast, the local Wendland kernel with indirect manipulation (W_2 im) only shows a weak correlation. The box-plot visualizes this trend for the Giannelli test (Figure 5.3). A more spread fitting error for the local kernel W_2 can be observed compared to the kernel functions with larger support (compare W_2, W_5, W_{15}). A reason for this result is that local basis functions with a small support are very sensitive to the random center distribution, whereas global kernels are more robust. For instance, the chance to randomly place control points in already optimal regions, such they do not influence the fitting result, is much higher for local basis functions. Hence, we can state that global kernels are typically more accurate.

With our settings now we can support the assumption that a RBF deformation with linear precision (direct manipulation, Equation (2.5)) is superior to a RBF deformation without the polynomial (indirect manipulation, Equation (2.2)) for local kernels, e.g., W_2. In contrast, for a global deformation setup (W_{15}, Tri) the fitness scores are equivalent (compare W_{15}/Tri im to

	W_2 im	W_2 dm	W_5 im	W_5 dm	W_{15} im	W_{15} dm	Tri im	Tri dm	Shell	FFD	All
Giannelli	0.42	0.63	0.72	.85	0.88	0.91	0.88	0.87	0.63	0.70	0.68
Sine	0.36	0.66	0.45	0.74	0.83	0.83	0.94	0.93	0.66	0.60	0.53

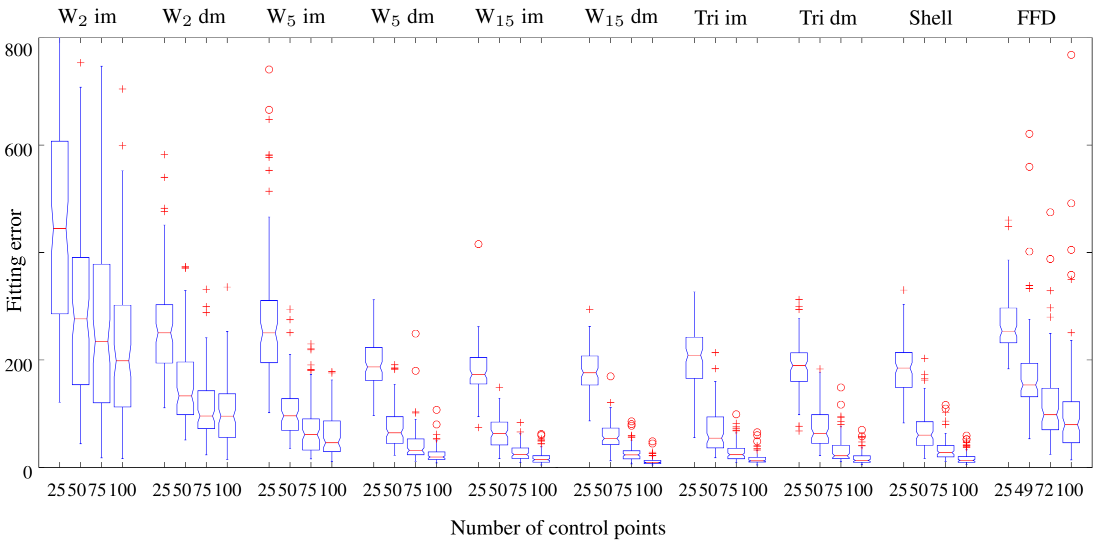


Figure 5.3: Variability results for 1D function approximation. The table shows the Spearman's correlation between variability and the fitting error. For the widest Wendland kernel W_{15} and for the triharmonic kernel the correlation is very strong. The box-plot visualizes the key trend for the Giannelli test function, showing that a higher variability results in a better fit for a chosen deformation method.

W_{15}/Tri dm in Figure 5.3). We also note that the fitness scores of shell deformations is similar to global triharmonic RBFs because the employed shell deformation minimizes a similar fairness energy [BKP$^+$10].

FFD is slightly worse than global RBF but comparable to local direct RBF and superior to local indirect RBF. But, for FFD we only tested one spline basis function in contrast to the 8 different settings for RBFs such that we can not state that one particular deformation method is generally superior with respect to the fitting quality.

Analyzing all tests together reveals a strong correlation between variability and fitting error for the Giannelli function. The correlations of the results for the sine function are very similar, such that we omit their visualization and can generally conclude that higher variability results in better fits.

5.1.2 Results: Regularity

Regularity is meant to estimate the expected convergence speed of an evolutionary algorithm and is computed as the inverse condition number $\kappa^{-1}(\mathbf{O})$ of the deformation matrix, see Equation (4.5). As the optimization algorithm of the fitting problem we use a (1,10)-CMA-ES of the shark3.0 library [AH12] as we motivated before. We measure convergence speed by counting the number of iterations until the optimization converges and we consider the algorithm converged as soon as the optimizer reaches a fitness value that is within a 5% tolerance of the true analytic solution.

Figure 5.4 shows the Spearman's correlations for the sine and Giannelli test function and it shows the plots of the regularity criterion against the number of iterations until convergence for the different basis functions exemplary for the Giannelli test function. Only 3 out of the 20 tests are not significant, according to our barrier of 0.01, and they are marked with the p-value in brackets. Again, the resulting correlations for the Giannelli and the sine test function are very similar, such that we omit the plot of the latter one.

For direct RBF manipulation of all Wendland kernels we observe a strong correlation between regularity and convergence speed. For Triharmonic kernels and direct manipulation the correlation still is moderate. This correlation becomes weaker for the indirect manipulation results. When comparing indirect and direct manipulation according to their scores, Figure 5.4 (top left) reveals that in the experiments direct manipulation setups have a better regularity and converge faster than the indirect ones, which is in line with the results of [MOS06].

The relation of our regularity criterion with Rothlauf's locality measure [Rot06] motivates the analysis of Wendland kernels with different support radii. It can be observed in Figure 5.4 (top right and bottom center) that more local kernels converge faster than more global ones, which is also hinted at by their better regularity values.

However, the very low regularity scores for "global" Wendland and triharmonic kernels requires further investigation. We assume that the overlapping of kernel functions has a large impact on the regularity score. If no kernels overlap each other then an orthogonal deformation matrix, which has optimal regularity, can easily be constructed with a uniform center distribution. But, the more kernels overlap, the more they disturb each other, which reduce regularity drastically, especially for truly global kernels. This assumption is supported by the link between the condition number and the minimum angle among the columns of the deformation matrix. A small minimum angle caused by a large overlap of kernels results in badly conditioned matrices [CC08]. The direct manipulation approach repairs this effect to some extent. For example, the variation of one parameter can affect centers, which are not linked to this parameter, for indirect manipulation but not for direct manipulation.

The medium correlations and the plot of the shell tests (Figure 5.4 top left) are similar to the direct Triharmonic RBF tests. Whereas, the weak FFD correlations are slightly above our significance barrier. Although FFD has low regularity scores, the optimization converges rather quick (Figure 5.4 top left) similar to the smaller indirect Wendland kernel (Figure 5.4 top right). But these two basis functions result in rather low fitness scores as we have discovered during the

	W_2 im	W_2 dm	W_5 im	W_5 dm	W_{15} im	W_{15} dm	Tri im	Tri dm	Shell	FFD	All
Giannelli	0.58	0.71	0.55	0.74	0.33	0.78	0.35	0.57	0.45	0.26 (0.01)	0.91
Sine	0.53	0.71	0.52	0.71	0.35	0.81	0.08 (0.42)	0.52	0.59	0.23 (0.02)	0.91

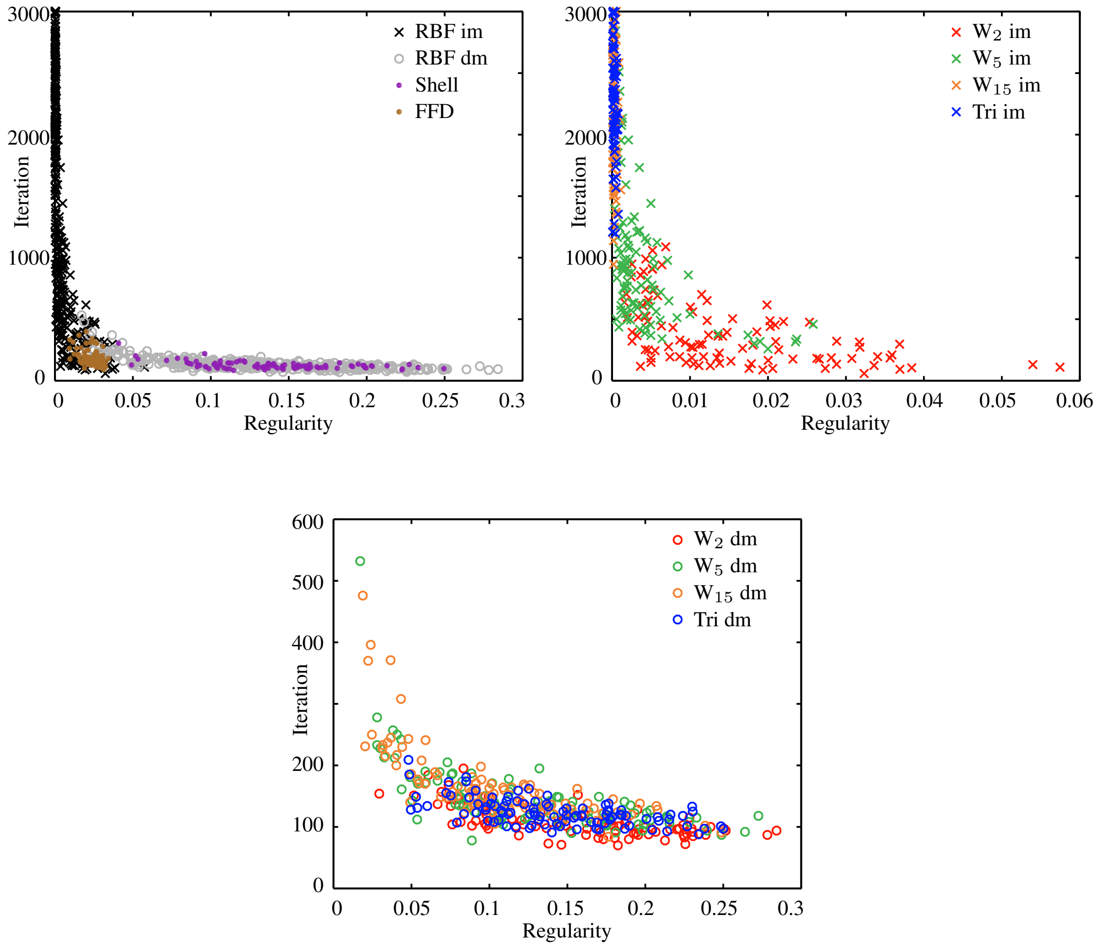


Figure 5.4: Regularity results for 1D function approximation. The table shows the correlation between regularity and convergence speed (#iterations until convergence) with p-values in brackets if the tested deformation function is not significant (threshold 0.01). The plots visualize the trend curve for the Giannelli test function. Top left plot: Comparison of indirect/direct RBF manipulation, FFD, and shells. Direct manipulation and shells have better regularity scores and converge faster than the other methods. The other plots show a detailed comparison of indirect (top right) to direct RBF manipulation (bottom): Local kernels are preferable in terms of convergence speed. In general, it can be observed that a higher regularity results in a faster convergence.

variability discussion (Figure 5.3) and one general reason for fast convergence is minor quality of the results. Another source, that causes noise in the FFD experiments, is the smaller range of regularity scores.

But still, FFD fits the trend of all other plotted basis functions in Figure 5.4 (top left), showing a strong correlation between regularity and convergence speed for a range of different methods, which is also confirmed by the Spearman's coefficients in the table (last column).

The RBF, FFD, and shell tests show that the regularity criterion is a good indicator for the convergence speed of a deformation representation. Our direct-vs-indirect RBF and local-vs-global results are in agreement with results known from the literature [SKVS97, MOS06, Rot06, WCT12, SMB12, TR14], stating that w.r.t. convergence speed direct RBF manipulation of local kernels is best.

5.1.3 Results: Improvement Potential

The improvement potential estimates how much a given deformation setups can potentially improve the fitness value, based on how accurate a given *approximate* gradient $\mathbf{g}$ can be reproduced, as described in (4.6). We expect deformation setups that can approximate the direction $\mathbf{g}$ well to result in solutions with a better fitness value.

To emulate this approximate knowledge of a beneficial direction, we define $\mathbf{g}$ to be a distorted version of the true analytic gradient $\nabla_{\mathbf{x}} f$. For the computation of the gradient we blend the true fitness function with a varied version, which—in our scenario—can be interpreted as a distortion of the target to be fitted. Thus, we define a distorted sine wave as:

$$s(x,y) \;=\; \sin(\pi \cdot (x+y+0.5))^2$$

and distort the Giannelli function through:

$$\hat{s}(x,y) \;=\; \begin{cases} 1 & y+x<0.5\,, \\ 2(1-y-x) & 0.5<y+x<1\,, \\ 0.5\cos(2\pi \cdot q^{0.5})+0.5 & q<\frac{1}{4}\,, \\ 0 & \text{otherwise}\,, \end{cases}$$

with $q = (x-1.25)^2 + (y-0.75)^2$ as depicted in Figure 5.1 (bottom right). The distorted fitness is then defined as $\hat{f}(\mathbf{p}) = \sum_{i=1}^{m} (o_{\mathbf{p}}(x_i,y_i) - \hat{s}_i)^2$. Given this variant we compute the approximated gradient $\mathbf{g}$ with respect to the phenotype as:

$$\tilde{\mathbf{g}} \;=\; \beta \frac{\nabla_{\mathbf{x}} f}{\|\nabla_{\mathbf{x}} f\|} + (1-\beta)\frac{\nabla_{\mathbf{x}} \hat{f}}{\left\|\nabla_{\mathbf{x}} \hat{f}\right\|}\,, \qquad \mathbf{g} \;=\; \frac{\tilde{\mathbf{g}}}{\|\tilde{\mathbf{g}}\|}\,.$$

The blending parameter β models the "reliability" of the approximate gradient $\mathbf{g}$, which matches the exact gradient for $\beta = 1$. Here, we conduct our experiments with $\beta = 0.75$, which induces distortion on the gradient according to a misleading target. In [RAMB16] we applied a constant

	W$_2$ im	W$_2$ dm	W$_5$ im	W$_5$ dm	W$_{15}$ im	W$_{15}$ dm	Tri im	Tri dm	Shell	FFD	All
Giannelli	0.97	0.90	0.93	0.90	0.93	0.95	0.95	0.96	0.94	0.91	0.95
Sine	0.97	0.95	0.91	0.92	0.86	0.93	0.98	0.95	0.93	0.95	0.94

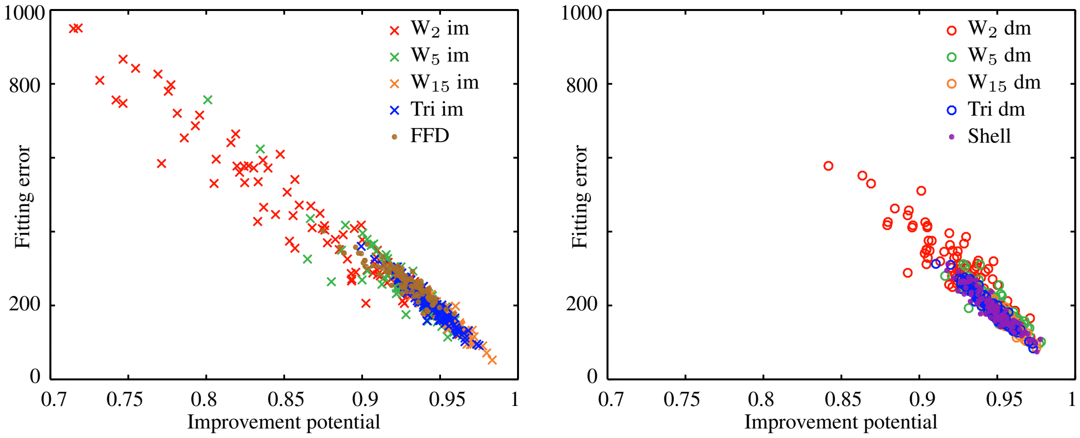


Figure 5.5: Results of the analysis of improvement potential for 1D function approximation. The table shows the correlation between improvement potential and the fitness (fitting error) where all tests are significant (thus the p-values are omitted). The plots visualize the trend curve for the Giannelli test function. Left plot: indirect RBF manipulation and FFD. Right plot: direct RBFs and shells. It can be observed that global RBF kernels and shells lead to the best potential scores and the best fitting results.

distortion in normal direction of the height field. But because such a simple distortion can be exactly reproduced with the linear precision of the deformation methods, we applied the more complex one here.

Before performing any tests we can state for RBFs without the polynomial term: For identical center placement and kernel function the improvement potential for direct and indirect manipulation is identical because:

$$P\left(\mathbf{\Phi\Psi}^{-1}\right) = \left\| \left(\mathbf{I} - \mathbf{\Phi\Psi}^{-1}\left(\mathbf{\Phi\Psi}^{-1}\right)^{+}\right)\mathbf{g} \right\|^{2} = \left\| \left(\mathbf{I} - \mathbf{\Phi\Phi}^{+}\right)\mathbf{g} \right\|^{2} = P(\mathbf{\Phi}) \ .$$

The table in Figure 5.5 shows the Spearman's correlation between our improvement potential and the final fitting accuracy. Since we want to measure the *potential* for improvement, we measure the fitting accuracy by the distance to the (known) analytic solution. Since the (quadratic) function approximation problem can be solved by setting the fitness gradient to zero, the correlation is 1 if a precise gradient is known ($\beta = 1$), which is not shown. For a distortion ($\beta = .75$) the correlation between our criterion and the resulting fitting error still is very strong but naturally would decrease with β.

All results are significant for both test functions. The plots in Figure 5.5 depict the improvement potential against the resulting fitting error, left for indirect RBFs and FFD, right for direct manipulation and shells. It can be observed that the triharmonic kernel, the Wendland kernel

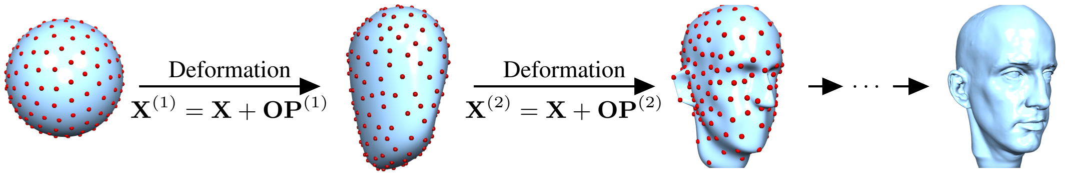

Deformation
$\mathbf{X}^{(1)} = \mathbf{X} + \mathbf{OP}^{(1)}$
Deformation
$\mathbf{X}^{(2)} = \mathbf{X} + \mathbf{OP}^{(2)}$
...

Figure 5.6: The 3D template fitting scenario: An initial sphere $\mathbf{X}$ is deformed via a linear deformation function $\mathbf{OP}$. The handle displacement $\mathbf{P}^{(1)}, \mathbf{P}^{(2)}, \ldots$ (red dots) results in the deformed meshes $\mathbf{X}^{(1)}, \mathbf{X}^{(2)}, \ldots$ aiming to improve the fitting accuracy to a target scan.

W_2 im	W_2 dm	W_5 im	W_5 dm	W_{15} im	W_{15} dm	Tri im	Tri dm	Shell	FFD	All
0.52	0.89	0.84	0.90	0.94	0.94	0.94	0.94	0.89	0.88	0.69

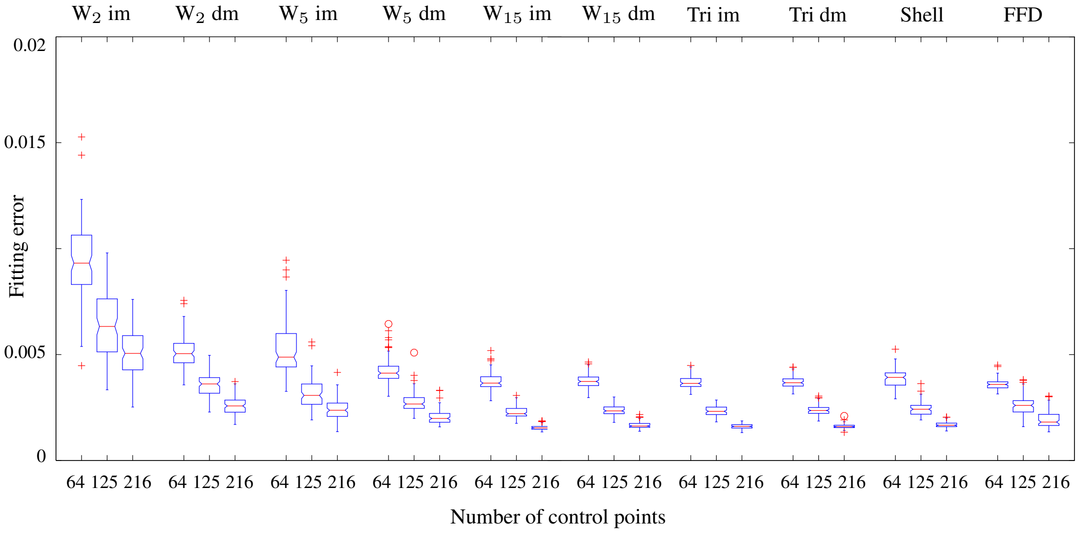


Figure 5.7: Variability results for 3D template fitting. The table shows the Spearman's correlation between variability and the fitting error. For the widest Wendland kernel W_{15} and for the triharmonic kernel the correlation is very strong. The box-plot visualizes a similar trend as for the 1D Giannelli test function, showing that a higher variability results in a better fit for a chosen deformation method.

with largest support, and the shell deformation approximate the estimated gradient direction best and also result in a better solution. Like at the variability discussion FFD is superior to local Wendland kernels but not as good as the global ones or shells.

Overall, these results demonstrate that the idea of estimating the improvement potential by approximating an (approximate) fitness gradient works well even for imprecise gradient information, which might very well be available in practical real-world applications.

5.2 Test Scenario: 3D Template Fitting

In the previous 1D function approximation scenario we were able to analytically compute the global optimum, which we exploited for the analysis of the evolvability criteria. The 3D template fitting scenario described in this section is considerably more complex, since the fitness function has many local minima and the global solution is not known. The goal is to fit a given triangle mesh $\mathbf{X}$ (a sphere in our experiments) to a target face scan $\mathbf{T}$ using a deformation $\mathbf{o}\colon \mathbb{R}^3 \to \mathbb{R}^3$, as depicted in Figure 5.6. The sphere model is discretized using $m \approx 10k$ vertices. The scan consists of $l \approx 12k$ points. In contrast to the 1D function approximation problem, where each point $(x, y, 0)^{\mathrm{T}}$ on the plane corresponds to a point $(x, y, s(x, y))^{\mathrm{T}}$ on the height field, there is no such one-to-one correspondence between the vertices of the sphere and points of the scan. Hence, the fitness function, which measures the approximation error between the two models, computes distances between any point on the (deformed) sphere and its closest point of the scan, and vice versa. These closest points are denoted by

$$\mathbf{c}_{\mathbf{T}}(\mathbf{x}_i) = \arg\min_{\mathbf{t}_j \in \mathbf{T}} \|\mathbf{x}_i - \mathbf{t}_j\| \ , \quad \mathbf{c}_{\mathbf{X}}(\mathbf{t}_j) = \arg\min_{\mathbf{x}_i \in \mathbf{X}} \|\mathbf{t}_j - \mathbf{x}_i\| \ .$$

These closest-point-pairs vary during the iterative optimization depending on the deformed sphere $\mathbf{X}^{(k)}$ at iteration k. Adding a Laplace regularization term, which prevents over-fitting and results in a higher surface quality, yields the (non-static) fitness function

$$\begin{aligned} f(\mathbf{P}) \ &= \frac{1}{m} \sum_{i=1}^{m} \left\| \mathbf{x}_i + \mathbf{o}_{\mathbf{P}}(\mathbf{x}_i) - \mathbf{c}_{\mathbf{T}}\left(\mathbf{x}_i^{(k)}\right) \right\|^2 \ + \ \frac{1}{l} \sum_{j=1}^{l} \|\mathbf{t}_j - \mathbf{c}_{\mathbf{X}^{(k)}}(\mathbf{t}_j)\|^2 \\ &+ \frac{w_r}{A} \sum_{i=1}^{m} A_i \, \|\Delta \mathbf{o}_{\mathbf{P}}(\mathbf{x}_i)\|^2 \ , \end{aligned} \tag{5.2}$$

where $\mathbf{x}_i^{(k)}$ denotes the i-th vertex of the mesh $\mathbf{X}^{(k)}$. For the regularization term $A = \sum_{i=1}^{m} A_i$, A_i denotes the corresponding area of vertex $\mathbf{x}_i$ (see Figure 2.7), and w_r denotes the regularization weight. For our purpose the simple regularization is sufficient (see [AZB15] for details and extended discussions).

In the evolutionary setting, we again employ the CMA-ES of the shark library [AH12, IHMG08] to minimize the fitness function (5.2). Since we cannot compute the analytic solution as a ground truth reference for analyzing the evolvability criteria, we fall back to the gradient-based Gauss-Newton approach of [AZB15] for computing the reference solution.

The analysis of the *variability* criterion with 64 (4×4×4), 125 (5×5×5), and 216 (6×6×6) kernels/handles (grid points) confirms the findings of the previous section: Since the deformation matrices have full rank, adding more control points directly increases the variability, which correlates (very) strongly with the fitting accuracy, see Figure 5.7.

Analyzing the relation of our *regularity* criterion and convergence speed reveals mainly significant correlations. For indirect RBF manipulation (triharmonic and widest Wendland kernel) and FFD the results are clearly not significant because these tests were terminated between 20000

W_2 im	W_2 dm	W_5 im	W_5 dm	W_{15} im	W_{15} dm	Tri im	Tri dm	Shell	FFD	All
0.57	0.52	0.51	0.56	0.08 (0.42)	0.73	0.1 (0.31)	0.76	0.24 (0.01)	0.08 (0.46)	0.94

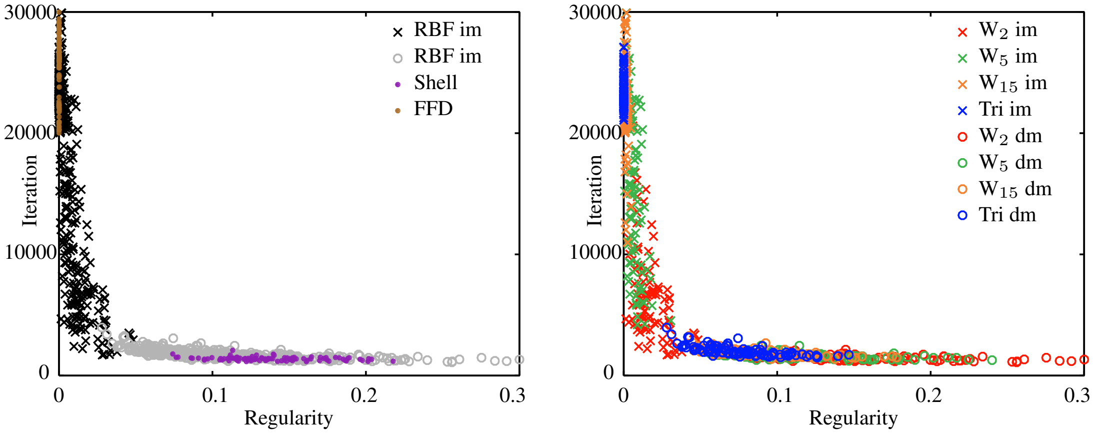


Figure 5.8: The results of the regularity tests for 3D template fitting are similar to the 1D test case. Although four tested functions are not significant (as the p-values depict in brackets) the overall correlation between regularity and convergence speed (number of iterations) is significant and strong. Moreover, direct RBF manipulation converges faster than indirect RBFs and local kernels converge faster than global ones, FFD, and shells.

and 30000 iterations before they converged. The p-value is at the border for the shell deformation. But the overall correlation is very strong and plotting regularity against numbers of required iterations (Figure 5.8) shows the same trend as in the 1D function approximation scenario, which confirms that our regularity formulation characterizes convergence speed even in more complex scenarios. Moreover, the comparisons between indirect and direct RBF manipulation, FFD, and shells strengthen the 1D results: Direct manipulation converges faster than indirect manipulation (Figure 5.8, right), local kernels converge faster than global ones, FFD converges slow, and shells converge similar to direct triharmonic kernels (Figure 5.8, left).

Because the 3D deformation leads to an $(m \times 3)$-dimensional fitness gradient we utilize equation (4.7) for the computation of improvement potential. During the optimization procedure the closest-point-correspondences change, which leads to an inaccurate gradient right after the first iteration. Moreover, we compute an approximation of the initial gradient by omitting the regularization term ($w_r = 0$ in equation (5.2)) to evaluate the improvement potential. The correlation between the improvement potential and the final fitting accuracy is significant for all deformation functions. All correlations are roughly moderate to strong for each function with an almost very strong correlation for all tests together (table in Figure 5.9). The plot in Figure 5.9 again is similar to the 1D scenario with equivalent conclusions: global RBF and shell fit better than local RBF kernels. But contrarily to the 1D scenario, FFD shows very good fitting results. The results demonstrate again that the approximate gradient information can be rather inaccurate and will still lead to a valid prediction of the eventual fitness.

W_2 im	W_2 dm	W_5 im	W_5 dm	W_{15} im	W_{15} dm	Tri im	Tri dm	Shell	FFD	All
0.47	0.53	0.37	0.51	0.50	0.62	0.65	0.67	0.63	0.77	0.8

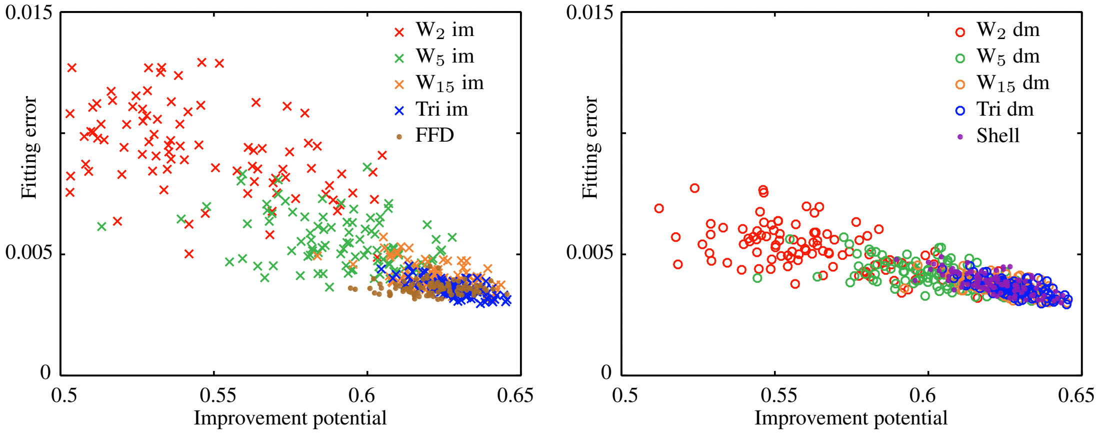


Figure 5.9: The correlation between improvement potential and fitting quality is strong even for a rough estimation of the fitness gradient in the 3D scenario. Like in the 1D tests local kernel have a lower potential as well as a higher final fitting error than global RBFs and shells. But FFD performs similarly good as these two deformations in contrast to the results in the 1D scenario.

Overall, the analysis of the 3D template fitting scenario confirms the results of the simpler 1D function approximation setting: Our formulations of variability, regularity, and improvement potential indeed represent reliable criteria for evaluating the quality of deformation setups.

Given the quality criteria the next step is their efficient optimization, which we discuss in the next chapter. However, we highlight the conflict between regularity and improvement potential. The variability score is constant for a given number of control points. But, regularity and improvement potential depend on their distribution and on the employed basis function. The randomized tests show that highly regular setups (e.g. local RBF kernels) have low improvement potential scores and vice versa. Thus, we analyze this conflict and Pareto-optimal solutions next.

5.3 Summary

A smart representation design tremendously supports the efficiency of industrial product optimization. Inspired by the biological concept of evolvability we analyzed our mathematical model for quantifying the quality of linear deformation representations. Our formulation is based on the three characteristics variability, regularity, and improvement potential.

Our results confirm that increasing the *variability* of the deformation reduces the fitting error, hence it improves the quality of the solution. Furthermore, our experiments show a significant and strong correlation between *regularity* and convergence speed. And finally, the experiments support that approximated gradient information yields significant correlations between *improvement potential* and fitting accuracy.

Apart from the analysis of these three quality criteria a direct comparison between RBF, FFD, and shell deformation has been targeted. In general all three methods result in similar fitting quality. However, especially local RBFs had lower fitting scores than global ones. But, with respect to convergence speed these local RBF and direct manipulation clearly out-performed the other deformation functions.

Hence, the conflict between regularity and improvement potential is subject for further investigation in the next chapter. There we target the efficient construction of optimal compromise setups for RBF deformations.

6 Pareto-optimal RBF Centers

Our transformation of the concept of evolvability results in promising quality criteria for deformation setups. Thereby, we are able to distinguish fast converging setups from slow converging ones and we can rank setups according to their expected fitness improvement. Moreover, a direct optimization of the setups is feasible to increase the algorithm's convergence speed and fitting quality. But, our analysis of the quality criteria revealed that variability, regularity, and improvement potential are partly contradicting. For example, local RBF kernels result in faster convergence but inferior fitting quality than global ones and vice versa. It never happened that a setup could enable both: fast convergence *and* good fitting results. Thus, our next step towards optimal setups is an analysis of optimal compromises computed with a multi-objective optimization, where each objective consists of maximizing variability, regularity, and improvement potential. Again, we call the optimization process of a setup as "setup construction" to clearly distinguish from the design optimization process.

In real world applications designer-driven approaches still are applied to construct initial deformation setups. The designer defines target regions where the design has to be varied/optimized and places control points adapted to these regions. For example, in [SMB12] a FFD grid is manually constructed and RBF kernels are manually set up. For basic automated deformation setups commercial tools provide a uniform distribution of control points, e.g., a glider optimization [CBG$^+$14]. But these approaches lack the quality of a computational optimization approach.

Originally, deformation representations are employed in scattered data approximation, e.g., for approximating a target shape. In [BSJ11] the control points of non-uniform rational B-splines are optimized by a gradient-based method to improve the approximation quality of a wing. In [ZWS05, VGJS11] a uniform setup of a control grid is refined in sensitive regions, i.e., parameters are added, resulting in an improved approximation. Amoignon [AHN14] tackles the problem that uniform control grids for FFD might have empty grid cells. Instead of adjusting the grid to the design he deforms the design (e.g., wings) to completely fill out the grid. To obtain RBF setups that are adapted to a target, different basis functions are iteratively evaluated and selected at fixed locations [WS98] or their location is being optimized [GY00, OBS04]. All these approaches are specialized to set up control points for approximating one fixed target. Thereby, they neglect numerical properties of the deformation setup which are important for, e.g., the convergence speed of an evolutionary optimization.

The representation setup of adaptive B-splines for an evolutionary design process is targeted in [OJS01, YSTY16]. The optimization alternates between approximation of a shape and adaptation of the representation. To test whether an adjusted representation is beneficial for the optimization this process is performed for a few iterations. Thereby the performance of a representation is measured by the objective function of the actual optimization task. In [SIHE14] the representation is optimized implicitly by adding its parameters to the approximation problem. The criterion for a high-quality representation purely depends on the target of the optimization omitting further aspects of this process like convergence speed.

This again motivates the application of our evolvability criteria definitions. These criteria address the convergence speed *and* include target information or human knowledge. Based on our model we analyze how to obtain high-quality center distributions efficiently in this chapter. Although, we would have to perform a three-objective optimization it turns out that variability is constant because we fix the number of kernels/centers. During our optimization tests of the center distributions the deformation matrix always had full rank and thereby optimal variability. Thus, we omit this criterion here.

As described in Chapter 2 the setup's quality depends on the amount of parameters, where they vary the design, and which kernel is employed. For a basic analysis of compromise setups, we utilize RBF deformations with a constant number of identical kernels and omit the polynomial term for direct manipulation for the following reasons.

(1) Although the convergence speed (Chapter 8) and fitting quality (Chapter 5) of FFD and shell deformations is similar to RBFs, their computation is much more complex. The evaluation of a vertex with an RBF kernel only requires the computation of one polynomial for indirect manipulation. Additional, a low dimensional matrix has to be inverted for direct manipulation. In contrast, shell deformation requires the inversion of a large (Laplace) matrix and FFD even requires a Gauss-Newton algorithm for the computation of the local coordinates. Thus, the computation of the deformation matrix is magnitudes faster for RBF deformation, which enables the application of evolutionary optimization strategies for its optimization.

(2) The flexibility of RBFs enables the placement of different kernel types at the RBF centers in theory. But, mixing different kernel types increases the difficulty to find general optimal configurations and is in conflict with theoretical results which we utilize, e.g., [Wen04]. Thus, we employ the same kernel at each center. Moreover, we omit the support radius as a free parameter for Wendland kernels to prevent degenerated results, e.g., a radius of 0 or ∞. In contrast to the previous section, where we computed the support radius such that each point of the design is covered by a certain amount of kernels, we simply set it to a certain fix value.

(3) Convergence speed and fitting quality of course depend on the number of employed parameters. But, infeasible settings would again lead to optimal results. For instance if each vertex of a design would be a RBF kernel maximal improvement and maximal variability would be achieved. Employing only one kernel, as the other limit, results in optimal regularity because the maximal and minimal singular values are identical and thereby the condition of the deformation matrix is 1. Thus we fix the number of parameters.

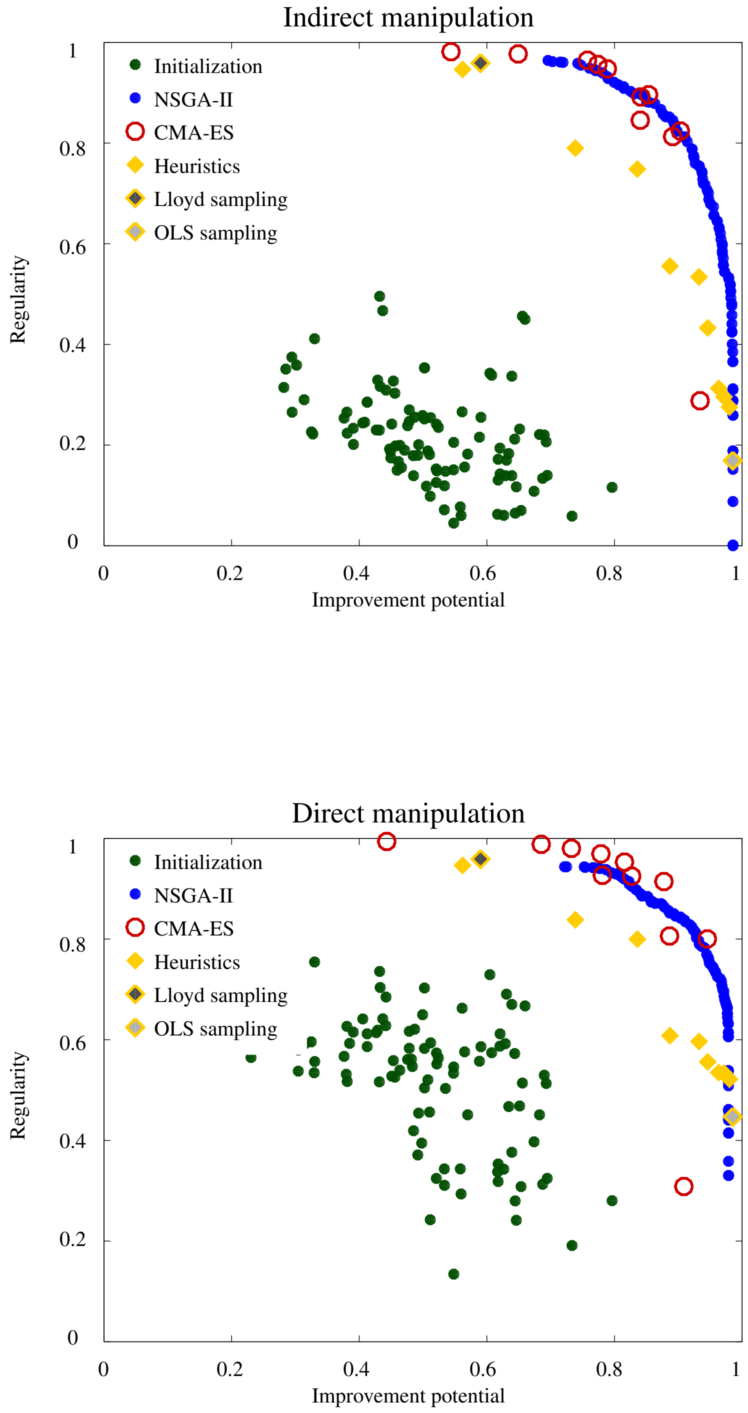
(4) For a proper comparison between kernel placements of indirect and direct manipulation we omit the polynomial term. Otherwise, indirect manipulation would employ less kernels than direct manipulation. The polynomial term is recommended only for triharmonic kernels for numerical reasons [Wen04]. In our experiments leaving out the polynomial never caused numerical problems and ensures that direct and indirect manipulation span the same design space.

In essence, the target subject for optimization with respect to regularity and improvement potential are the locations of the RBF kernels (or the RBF Center distributions). We start with the analysis of Pareto-optimal solutions computed with the multi-objective algorithm NSGA-II [DPAM02]. This leads to an estimation of the Pareto front as a ground truth. Although such a multi-objective optimization results in many optimal center distributions, its high computation time motivates alternative optimization approaches. In application scenarios we are rather interested in one particular compromise determined by the preference of a designer. Therefore, we evaluate the CMA-ES as a weighted single-objective evolutionary optimization, which reduces the computation time. For an even more efficient optimization process we evaluate the quality of heuristic strategies (Lloyd and OLS sampling), which “optimize” (set) the center distribution almost instantly. The proposed heuristics lack the quality of a computational optimization algorithm. However, they provide good initial deformation setups to increase the algorithm’s performance. The evaluation of the multi- and single-objective optimization as well as the heuristics are published in [RAMB17, RMB17]. In the last part of this chapter we evaluate a gradient-based deterministic optimization, which erases the random effects of the evolutionary algorithms. We employ the heuristic OLS sampling with an implementation by J. Achenbach, the second author of [RAMB17]. Its analysis and combination with the Lloyd sampling in the context of evolvability, as well as the evaluation of evolutionary and deterministic optimization approaches were conducted by the author of this thesis.

6.1 Multi-objective Evolutionary Optimization

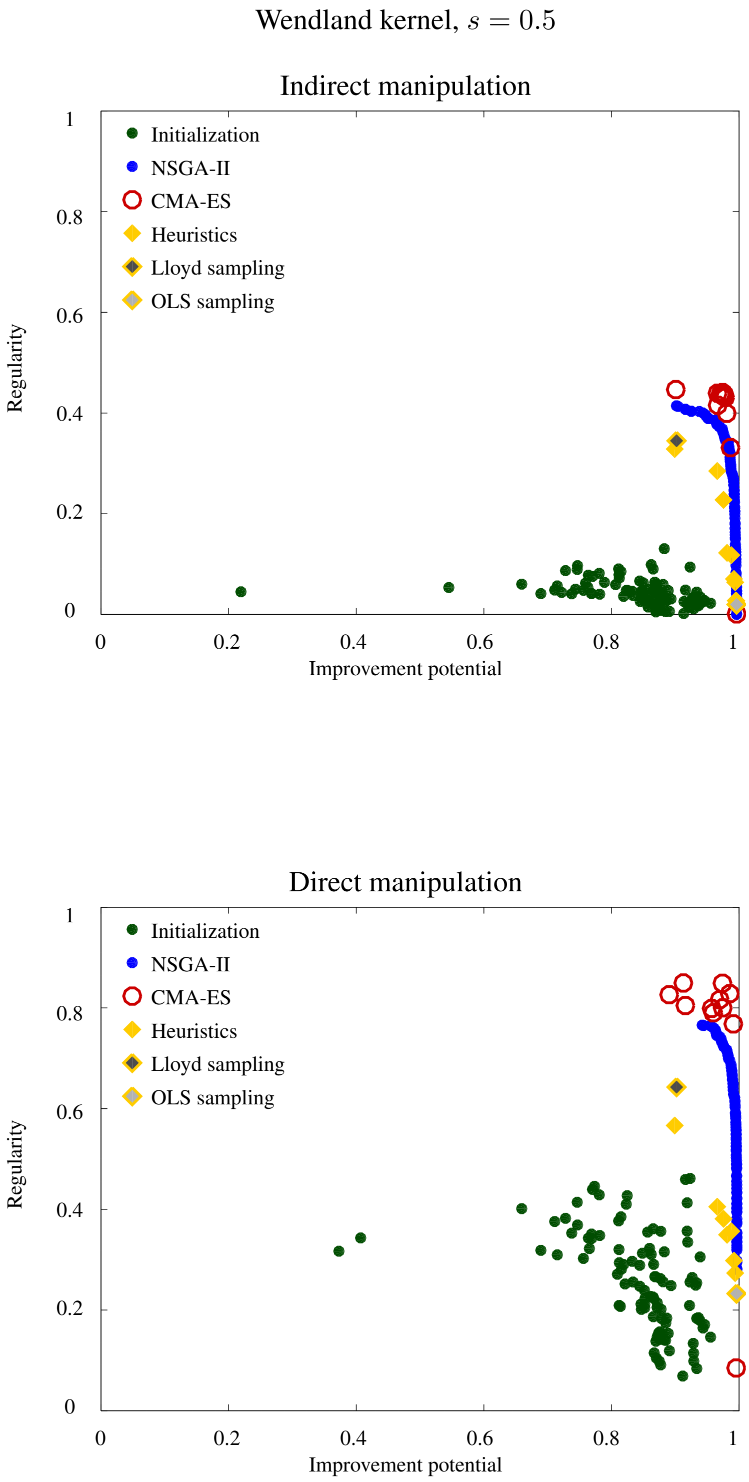
In the first test scenario the target of the design optimization is the fit of an initial plane to a height field, as shown in Figure 5.1 before. Instead of performing the actual fit, our goal now is to construct a well-performing *deformation setup*, which is characterized by the distribution of RBF centers. Note, that we optimize the numerical properties of the deformation matrix, which are independent of the RBF centers, in Chapter 8. We compute the deformation matrix $\mathbf{O}$ according to either indirect or direct manipulation, Equation (2.2) or (2.5), and optimize for the conflicting targets regularity and improvement potential (Equation (4.5), (4.6)). To cover a

Wendland kernel, $s = 0.25$



Indirect manipulation

Direct manipulation


Wendland kernel, $s = 0.5$
Indirect manipulation
Initialization
NSGA-II
CMA-ES
Heuristics
Lloyd sampling
OLS sampling
Regularity
0
0.2
0.4
0.6
0.8
1
Improvement potential
0
0.2
0.4
0.6
0.8
1
Direct manipulation
Initialization
NSGA-II
CMA-ES
Heuristics
Lloyd sampling
OLS sampling
Regularity
0
0.2
0.4
0.6
0.8
1
Improvement potential
0
0.2
0.4
0.6
0.8
1

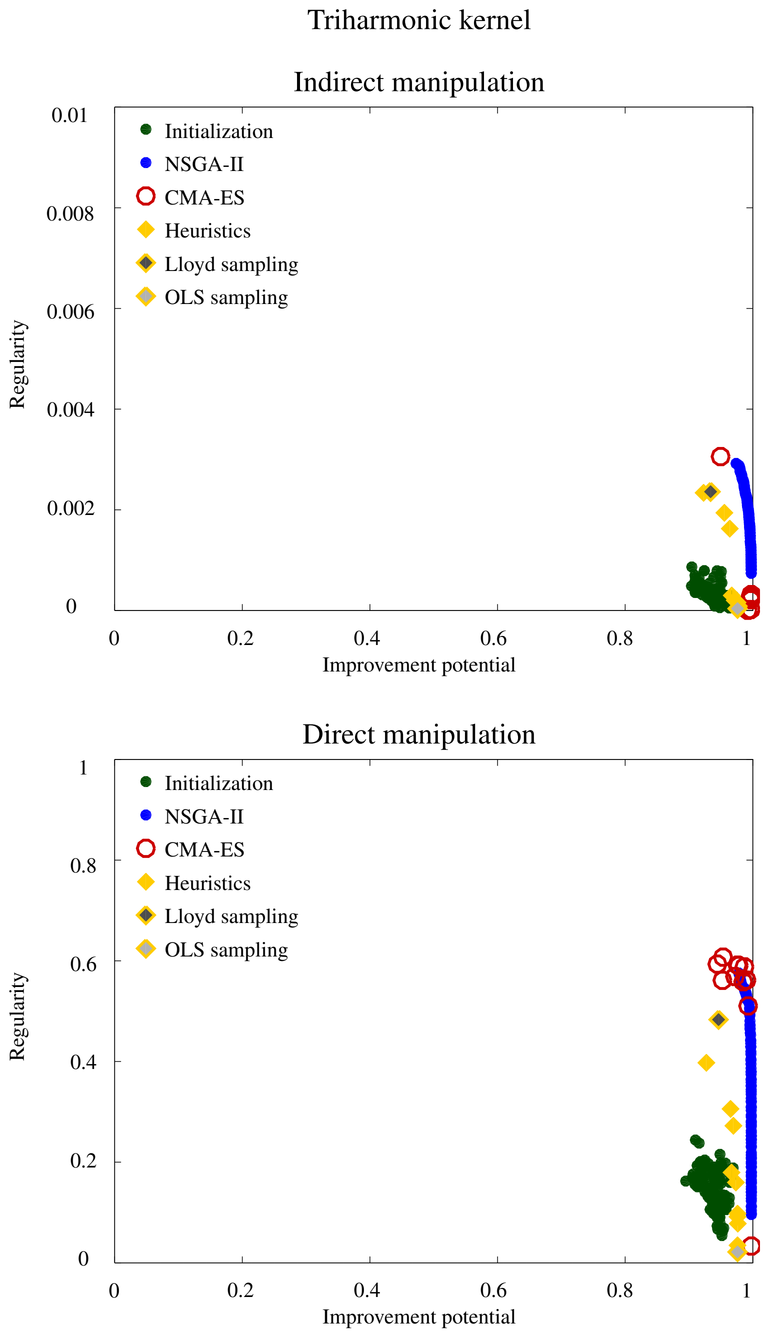


Figure 6.1: The Pareto front of NSGA-II (blue) and the initial random population (green) for the 1D scenario. The red circles are the results of the weighted single-objective CMA-ES which can out-perform the bi-objective NSGA-II. The orange diamonds refer to our combination of the efficient Lloyd and OLS sampling, which alone result in very regular setups (dark grey) or setups with a very good improvement potential (light grey), but their combination lacks quality.

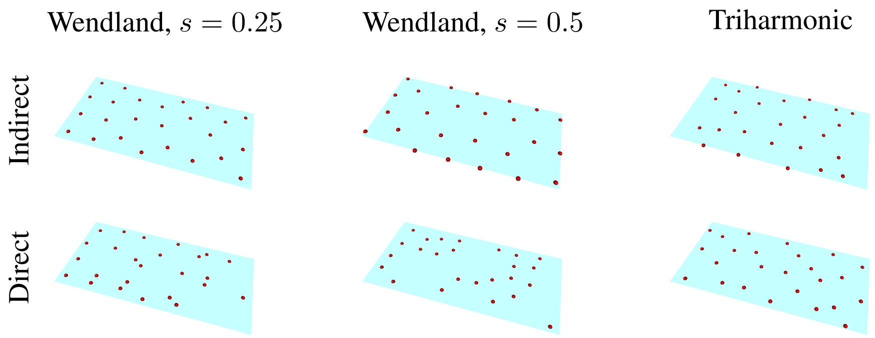


Figure 6.2: Optimized center distributions towards regularity are uniform for indirect manipulation (top) but tends to be unintuitive for direct manipulation (bottom).

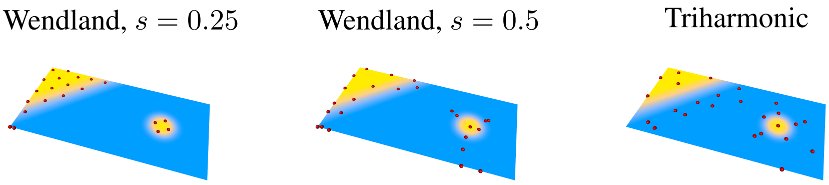
variety of kernel types, from rather local to global, we employ compact Wendland kernels with a fix support radii s of 0.25 and 0.5, and global triharmonic kernels. We chose $n = 25$ centers, each with two coordinates, which results in a 50-dimensional optimization problem.

We realize the multi-objective optimization with the NSGA-II algorithm of the shark 2.3 library [IHMG08] with the following settings: 100 individuals, tournament selection, polynomial mutation rate with a probability of 1/50, crossover with a probability of 0.9, and 25000 iterations. We initialize the algorithm with randomized center distributions and restrict the centers to the initial plane ($[0, 1] \times [0, 2]$) during the optimization. With these settings one optimization run took approximately 2 days on the used computer: Intel Xeon, 8x3.60GHz, 8Gb memory.

In Figure 6.1 we plot the resulting Pareto front as blue dots for the three tested kernels with indirect and direct manipulation, respectively. The green dots are the values of the initial population. The tests indicate a smooth well-shaped Pareto front. For the local Wendland kernel the front almost reaches the optimal value of 1 for regularity and improvement potential, respectively. Note that the very low regularity values of the triharmonic kernel for indirect manipulation goes along with our results of Chapter 5 and theoretical results in [Wen04]. However, these results demonstrate that achieving the optimal regularity score of one is limited if we only optimize the center distribution.

The results of Chapter 5 motivate global triharmonic or Wendland kernels (with a large support radius) for high improvement potential and thereby for very good fitting quality. This, is again depicted in Figure 6.1 with the green random kernel distributions. But, optimizing these distributions with NSGA-II leads to optimal improvement potential scores with only marginal differences between the local Wendland kernels (support 0.25) and the global triharmonic ones (Figure 6.1, blue dots). Hence, our general recommendation of global kernels for good fitting quality becomes weaker now because an optimized local kernel can become better than a unmodified global version and it is almost as good as a tuned global one.

Especially the center distributions maximizing either regularity or improvement potential, respectively, are interesting because they can be computed through a single-objective optimization. For *indirect* manipulation we obtain uniformly distributed centers (Figure 6.2, top) resulting in



Wendland, $s = 0.25$ Wendland, $s = 0.5$ Triharmonic

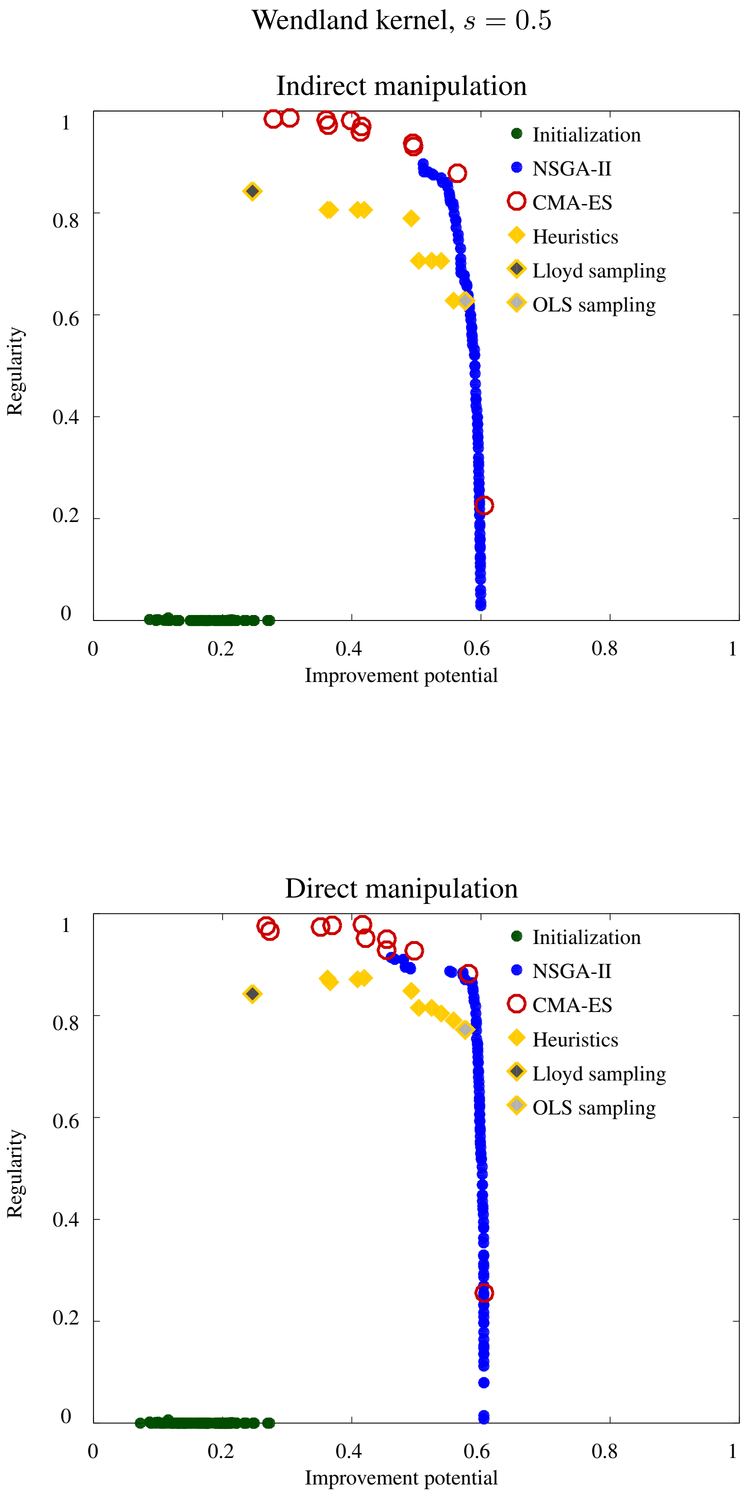
Figure 6.3: Target-adapted center distributions with optimal improvement potential. The compact kernels are mainly placed in regions with locally high fitting error (yellow), whereas triharmonic kernels are placed less intuitive (blue).

maximal regularity, in agreement with [Wen04]. In contrast, the center distributions leading to optimal regularity for *direct* manipulation are unpredictable (Figure 6.2, bottom), which shows the advantages of an automatic procedure for distributing centers in contrast to a purely designer-driven approach.

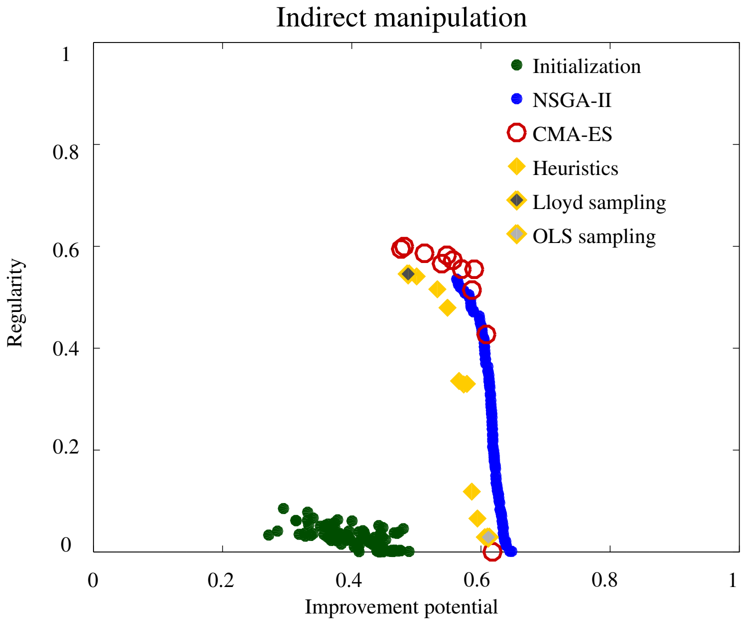
Center distributions with maximal improvement potential are adapted to the target height field for the compact Wendland kernels (Figure 6.3). The distribution is denser in regions which have to be deformed more. In contrast, centers for the global triharmonic kernel are not primarily placed in these regions (Figure 6.3, right), which is unintuitive for a designer. This again emphasizes the demand for an automatic construction of setups instead of a purely designer driven approach.

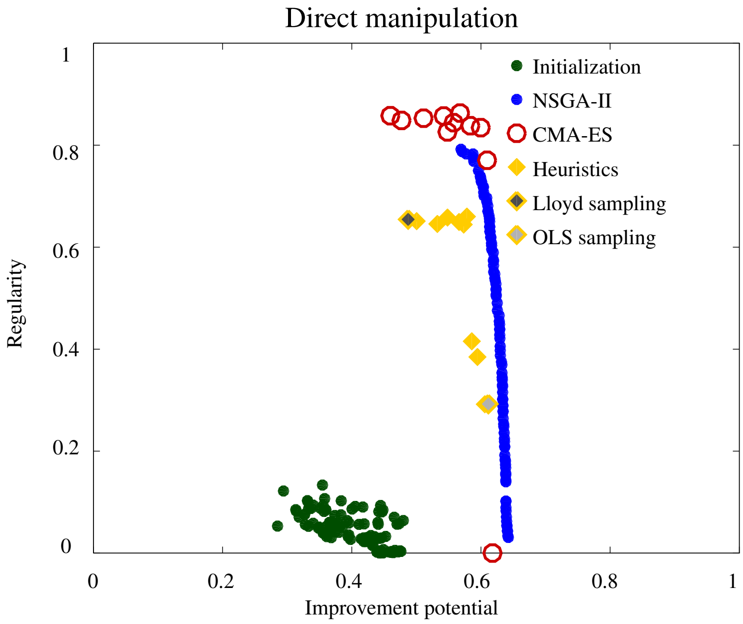
In the second test scenario we deform an initial sphere to closely fit the point cloud of a given face scan (see Figure 5.6). Like in the height field approximation scenario we intend to *set up* an optimal center distribution rather than performing the fitting. However, distributing centers for template fitting is more complex because the sphere and the scan are embedded in 3D such that each of the 25 centers consists of 3 coordinates, resulting in 75 parameters to be optimized. We choose the initial distributions randomly on the initial sphere, restrict the search domain to its bounding box $[-1, 1]^3$, and choose support radii of 0.5 and 1 for the Wendland kernels. The support radii are increased because the initial domain is larger than the domain of the function approximation scenario. Apart from the mutation rate, which we set to 1/75 according to the 75 parameters, we perform the multi-objective optimization with identical settings as in the function approximation scenario.

In Figure 6.4 we plot the Pareto front for the three kernel types with direct and indirect manipulation, respectively. These plots are qualitatively equivalent to the plots of the function approximation scenario, compare to Figure 6.1. In the 1D scenario we restricted the centers to the initial plane, which might be to restrictive for general applications. Hence, we skipped this restriction for the 3D tests. However, the optimized distributions converge towards the sphere as depicted in Figure 6.5. A reason for optimal RBF centers on the initial sphere is the stronger causal relation between the deformation parameters and shape deformation. Although the visualization is cumbersome, a uniform arrangement of centers is derived for optimal regularity and a target adapted one for the other end of the Pareto front, optimal improvement potential (Figure 6.5).


Wendland kernel, $s = 0.5$
Indirect manipulation
Regularity
Improvement potential
0
0.2
0.4
0.6
0.8
1
Initialization
NSGA-II
CMA-ES
Heuristics
Lloyd sampling
OLS sampling
Direct manipulation
Regularity
Improvement potential
0
0.2
0.4
0.6
0.8
1
Initialization
NSGA-II
CMA-ES
Heuristics
Lloyd sampling
OLS sampling

Wendland kernel, $s = 1$





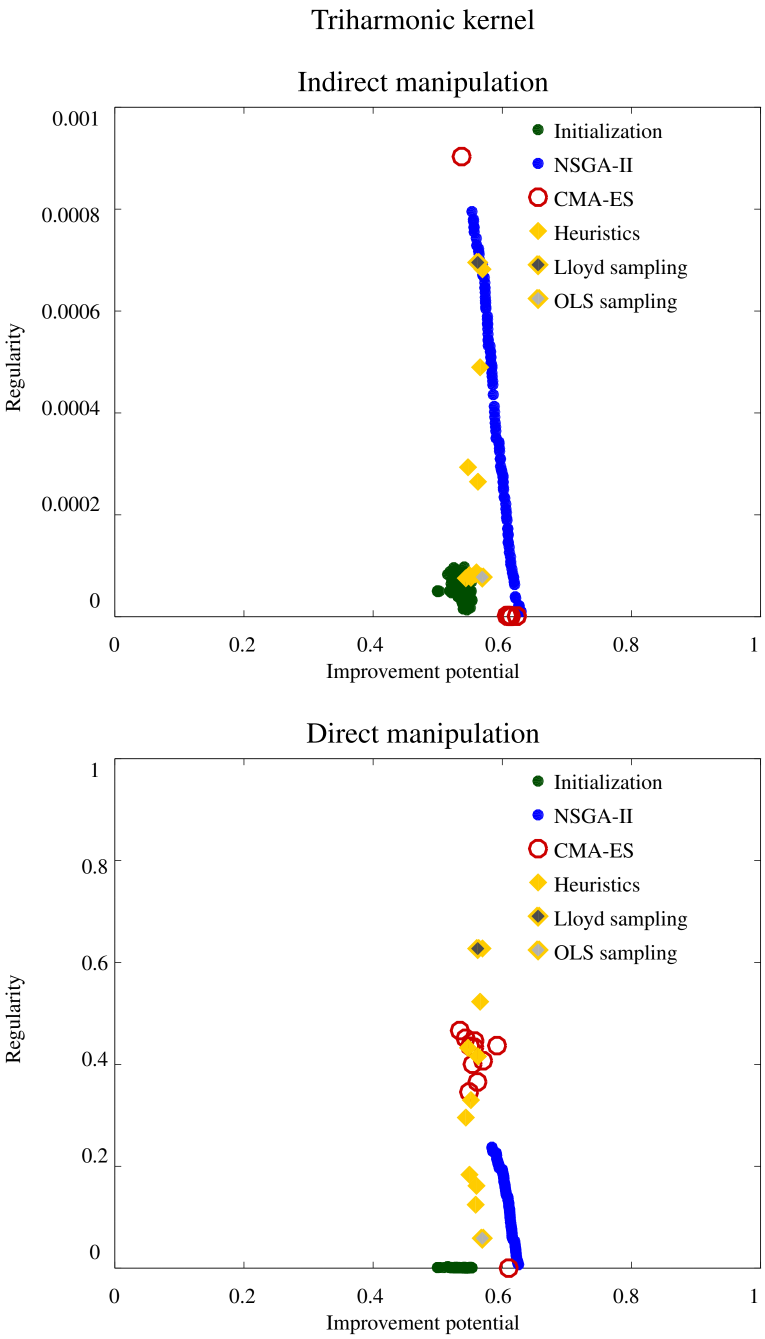


Figure 6.4: The Pareto front of NSGA-II (blue) and the initial random population (green) for the 3D scenario. The single-objective optimization CMA-ES (red) out-performs the bi-objective NSGA-II. The efficient Lloyd and OLS sampling alone result in very regular setups (dark grey) or setups with a very good improvement potential (light grey), respectively. But their combination (orange diamonds) lacks the quality of a true optimization procedure like in the 1D scenario.

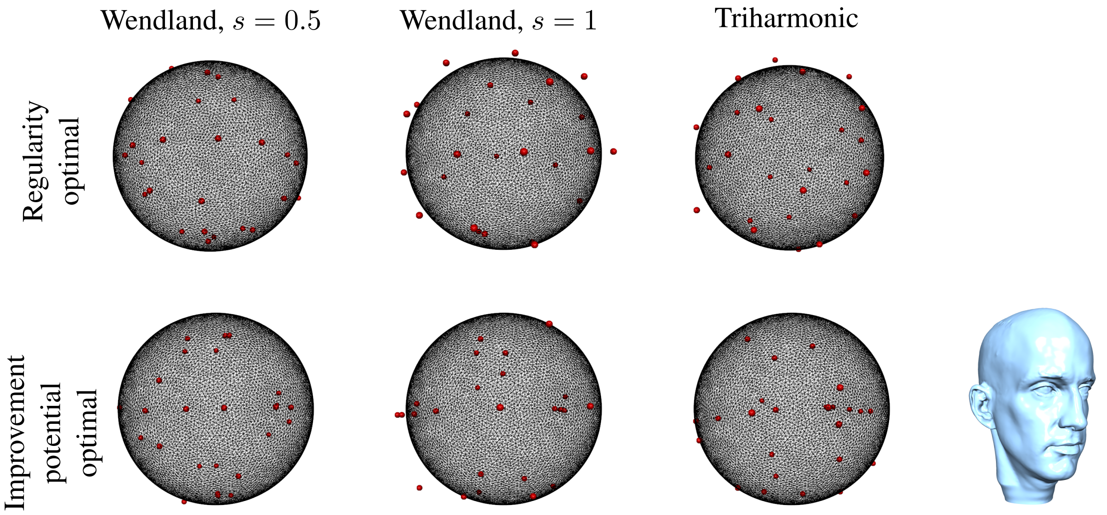


Figure 6.5: The optimized center distributions in the 3D scenario converge towards the initial sphere. Regularity-optimal centers are uniformly placed on/around the sphere whereas center distribution with optimal improvement potential are partly clustered.

The multi-objective optimization in both test scenarios, height field approximation and template fitting, runs up to 2 days, hence we target more efficient approaches for real-world applications. Instead of computing the whole Pareto front we are rather interested in one particular setup trading off regularity and improvement potential according to our preference. Therefore, we employ a weighted single-objective optimization next and utilize the Pareto front as a ground truth to test if this optimization is able to converge towards the front.

6.2 Single-objective evolutionary Optimization

The runtime of 2 days of a multi-objective optimization in our tests motivates alternative optimization approaches. Instead of computing the whole Pareto front the designer guides the construction of trade-off setups between regularity and improvement potential by setting a preference $\lambda \in [0, 1]$ based on their expertise.

By weighting Equation (4.5) and (4.6) we define an objective function f_λ for a preference-based single-objective optimization:

$$f_\lambda(\mathbf{O}) = \lambda R(\mathbf{O}) + (1 - \lambda) P(\mathbf{O}) \ . \tag{6.1}$$

Because such a single-objective optimization might not converge to the Pareto front, we analyze this in the following. As an optimization algorithm we choose a (25,100)-CMA-ES of the shark 2.3 library [IHMG08], we choose the preferences λ to be 0, 0.1, 0.2, …, 1 for Equation (6.1) and run the optimization for 1000 generations. The optimization of a setup for one preference took approximately 2 hours.

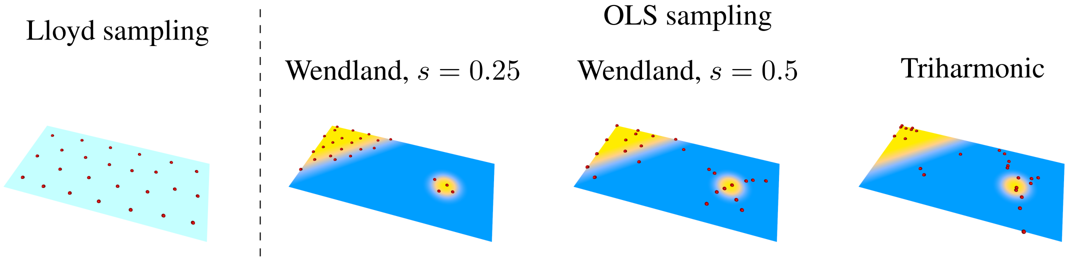


Figure 6.6: The Lloyd sampling results in a uniform center distribution independent of the employed kernel. Heuristic OLS setups with high improvement potential are adapted to the target. Wendland kernels are placed in regions with locally high fitting error (yellow) rather than in already optimal ones (blue).

The results of the single-objective optimization in Figure 6.1 and Figure 6.4 are depicted with the red circles. The clustering of solutions, e.g., Figure 6.1 and Figure 6.4 middle, shows that uniformly distributed preferences λ do not result in uniformly distributed solutions along the Pareto front. Therefore, a designer has to set the preference carefully. The single-objective optimization converges towards the solutions of the multi-objective NSGA-II and even performs slightly better because of its focus in one preferred direction. But, when focusing improvement potential the CMA-ES seems to get stuck in local optima (Figure 6.4).

The existence of these local optima especially for improvement potential can be explained as follows. Assume that all centers of a local Wendland kernel are placed beneath the plateau in the upper left yellow region of the plane in Figure 6.3. Moving even some of them through the already optimal blue region to fit the yellow peak in the center would increase the fitting error. The condition number as the regularity criterion behaves much smarter. As shown in [MY09] optimizing the condition number is a quasi-convex problem, i.e., a problem that can be approximated by a sequence of convex problems. Because for each convex problem finding a global optimum is relatively easy [BV04], we regard the optimization of regularity not as problematic as the optimization of improvement potential where local optima exist.

All in all we have shown the feasibility of such a single-objective optimization for scenarios where a designer is interested in a setup for one particular preference. The runtime of 2 hours and local optima motivate efficient heuristics to distribute centers.

6.3 Heuristic Approaches

Heuristic methods aim to generate good center distributions in a robust and efficient manner. They are analytically and geometrically motivated but lack the guarantee to be Pareto-optimal. In our test scenarios a single-objective optimization still runs for hours and might get stuck in local optima. Because we expect these drawbacks to become worse for more complex scenarios, e.g., with a more complex initial design or a larger amount of parameters, we propose and analyze

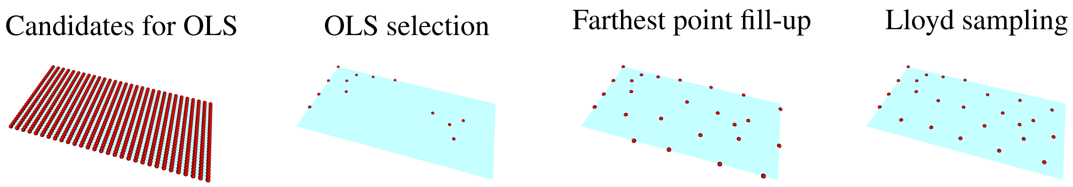


Figure 6.7: The combination of OLS and Lloyd sampling. Among a set of centers as candidates (left most) the best (13) centers are selected with OLS (center left). Than the remaining centers are placed with a farthest point sampling (center right) and adjusted with the Lloyd sampling (right).

a geometry-motivated approach for very regular setups and an analytically motivated approach for setups with high improvement potential.

Optimal center distributions targeting regularity are uniform distributions in our tests with indirect manipulation (Figure 6.2, top). Hence, we apply Lloyd sampling [Llo82] which is also known as k-means clustering [Mac67], which results in uniform center distributions similar to the regularity-optimal solutions (compare Figure 6.6, left most, and Figure 6.2, top). But the Lloyd sampling is not bound to the surface of, e.g., the sphere in the 3D scenario. Thus, we project the heuristically distributed centers to the sphere, which is inspired by the solutions of the 3D scenario.

Comparing the regularity score of the resulting setup to the Pareto front (see Figure 6.1, Figure 6.4, dark grey diamonds) reveals that the Lloyd sampling is close to the front for local Wendland kernels ($s = 0.25$ for the plane or $s = 0.5$ for the sphere). Uniform Lloyd sampling results in good regularity even for direct manipulation. For the triharmonic kernel in the template fitting scenario the heuristic even out-performs the multi- and single-objective optimization (Figure 6.4, bottom right). According to Equation (4.5) regularity is the ratio of the smallest to the largest singular value of the deformation matrix. For indirect RBF manipulation this singular value is bounded by the *separation distance*, which measures the minimal distance between any pairs of centers [Wen04]. The uniform Lloyd sampling by construction has a good separation distance and thus results in good regularity. This sampling performs better than any tested random distributions. It performs better than the evolutionary optimization in one test and it is fast to set up (less than one minute).

Previously, we motivated improvement potential (Equation (4.6)) by solving the approximation problem $\mathbf{g} = \mathbf{Op}$ for an estimated fitness gradient $\mathbf{g}$, the deformation matrix $\mathbf{O}$, and the deformation parameters $\mathbf{p} = (p_1, \ldots, p_m)$. Each parameter p_j is the coefficient for a kernel function $\varphi_j(\mathbf{x})$, which corresponds to a column $\mathbf{O}_j$ of the deformation matrix and to a center $\mathbf{c}_j$ for indirect manipulation. The orthogonal least squares method (OLS, [CBL89, GY00]) determines the influence of each parameter to minimize the approximation error to the estimated gradient in a greedy manner. OLS ranks the parameters according to their influence and thereby ranks the centers. Hence, given a set of candidate centers we can select the most important ones. We initialize OLS with a large set of candidates, 30×30 on a uniform grid on the initial plane in

2D (e.g. Figure 6.7 left), or 500 uniformly distributed on the initial sphere in 3D, and greedily select the best k ones for approximation of the gradient information.

However, we cannot apply this procedure for direct manipulation because the interpolation matrix $\mathbf{\Psi}^{-1}$ in Equation (2.5) disbands the correspondence between parameters and centers. Because direct manipulation and indirect manipulation result in equal improvement potential for identical center distributions, we simply apply this algorithm for indirect manipulation and switch to direct manipulation afterwards. For the function approximation scenario we show that target-adapted setups for the Wendland kernels in Figure 6.6 (center left, right) are similar to the Pareto-optimal solutions in Figure 6.3. But for the global triharmonic kernels OLS results in an unintuitive center placement (Figure 6.6 right). Nonetheless, the OLS setups are close to the Pareto front or even hit it in both test scenarios (see Figure 6.1, Figure 6.4, light grey diamonds) for the compact Wendland kernels. In conjunction with the small computation time of 1 minute, OLS is very efficient.

We now combine both methods to construct compromise solutions of Equation (6.1) almost immediately. The given preference $\lambda \in [0, 1]$ weights regularity and $(1 - \lambda)$ weights improvement potential. Thus, given n centers to be distributed we place $rd((1 - \lambda) \cdot n)$ centers in a gradient-adapted manner (using OLS) and $rd(\lambda \cdot n)$ centers in a regular manner (using Lloyd sampling). Note that we round up in favour of improvement potential with the rounding operator rd. For example, we distribute 25 centers with OLS for $\lambda = 0$, or we place 25 centers with the Lloyd sampling for $\lambda = 1$, or we distribute 13 centers with OLS and 12 centers with Lloyd for $\lambda = 0.5$. In general, we first distribute $(rd(1 - \lambda) \cdot n)$ centers with OLS sampling and then distribute the remaining $rd(\lambda \cdot n)$ centers with the Lloyd algorithm as depicted in Figure 6.7. We initialize the Lloyd algorithm with the chosen OLS centers and fill up the remaining $rd(\lambda \cdot n)$ centers by farthest point sampling [ELPZ97]: We add centers one by one such that they are as far away as possible from all other centers. We run the Lloyd algorithm on these centers and keep the OLS centers fixed. Because the Lloyd algorithm moves centers slightly off a curved domain, we simply project these centers back.

Our approach of combining Lloyd sampling and OLS lacks the quality to construct Pareto-optimal center distributions (Figure 6.1, Figure 6.4 orange diamonds). Furthermore, only $n + 1$ compromises can be constructed this way, which especially for a small number of centers leads to a very low resolution of the compromises. Nevertheless, the constructed compromise setups are better than random ones, which makes them good candidates for the initialization of a single-objective optimization.

Thus we analyze the evolutionary CMA-ES with heuristic initialization for the preferences $\lambda = 0.0, 0.5$, and 1 for Equation (6.1) in the 1D scenario. To increase the robustness with respect to local optima we apply the standard settings of the shark library which leads to a (7,15) CMA-ES for 25 centers/50 parameters. We terminate the optimization if a new result is only up to 0.1% better than the average of the previous 50 iterations. This strategy gives a certain robustness against an early termination of the randomized search.

In Figure 6.8 we show convergence plots of the fitness of the deformation setup $\mathbf{O}$ (for direct manipulation), either being solely improvement potential $f_0(\mathbf{O}) = P(\mathbf{O})$, a compromise between

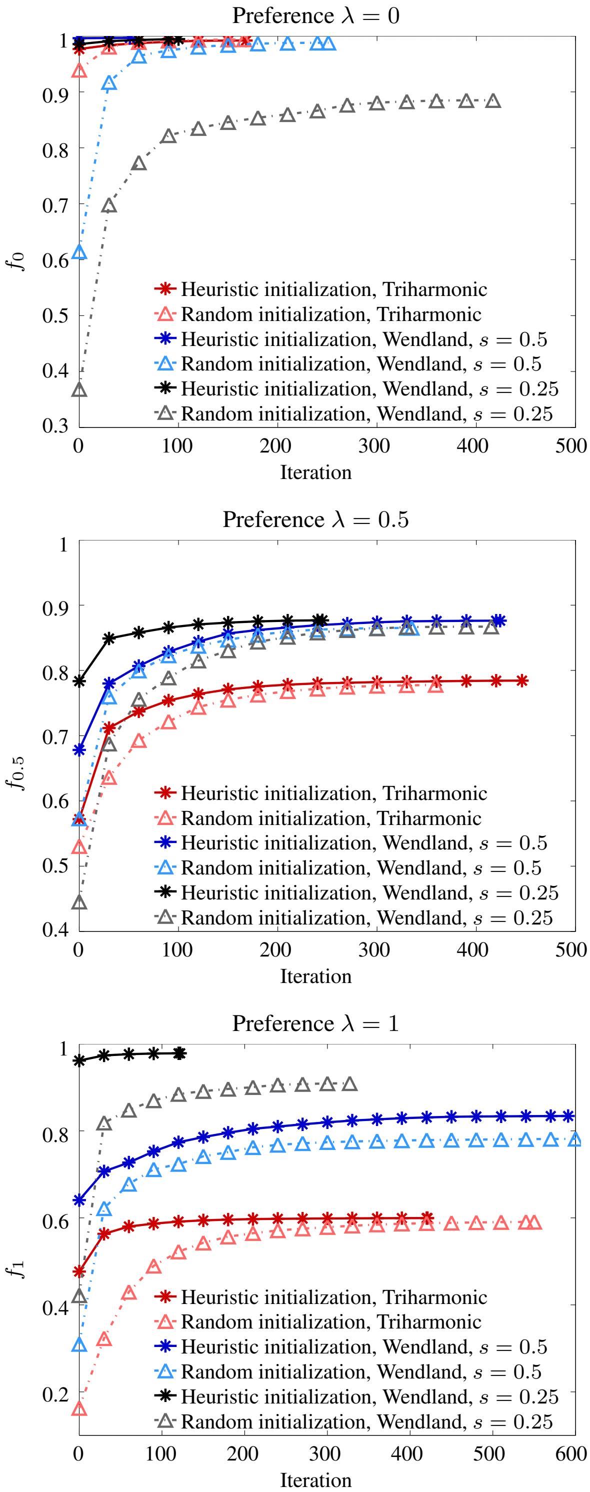


Figure 6.8: A setup optimization with our heuristic initialization (solid lines) converges faster and yields better values than the optimization with a randomized initialization (dashed lines).

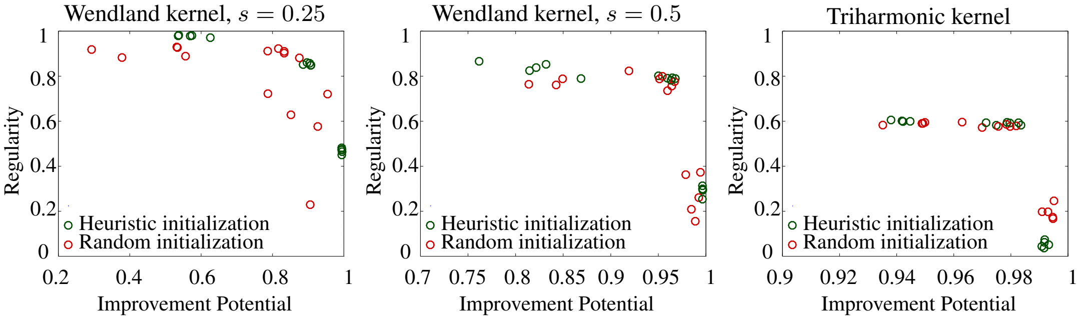


Figure 6.9: Comparison of heuristic (green) and random (red) initialization for the single-objective optimization of the kernel distribution for three preferences. Heuristic initialization leads to improved setups on average and results in less distortions.

improvement potential and regularity $f_{0.5}(\mathbf{O}) = 0.5 \cdot P(\mathbf{O}) + 0.5 \cdot R(\mathbf{O})$, or solely regularity $f_1(\mathbf{O}) = R(\mathbf{O})$. The fitness values are averaged over five trials. For each kernel the heuristic initialization (solid lines in Figure 6.8) out-performs random initializations (dashed lines in Figure 6.8). The CMA-ES with heuristic initialization converges to better values, converges faster to same values, or even converges faster *and* reaches better values than with random initializations.

Comparing the generated setups to the solutions computed with random initializations (Figure 6.9, red) reveals that a weighted single-objective optimization with heuristic initialization (green) shows a very high performance, except for the values of improvement potential for the triharmonic kernel, which are slightly worse (Figure 6.9 right). These results and the largely superior performance of the single-objective optimization confirm the benefit of our proposed OLS/Lloyd initialization for setup construction.

But, we also realize the effects of a randomized evolutionary search. Although, we already utilize a heuristic initialization the optimal setups are still spread along the Pareto front. For instance, if the focus is on regularity only, then the 5 tests result in setups with a large variance in improvement potential (Figure 6.9, center/right).

Whether this variance becomes worse for a fine-grained preference analysis we analyze next. We conduct the weighted single-objective optimization for 51 preference weights for each of the three kernels (indirect manipulation only) in the 1D and in the 3D scenario. Like before, 25 centers are optimized on the 2D plane. But we optimize 75 centers on the 3D sphere to improve the quality of the final fits later on. The color-coded weights in Figure 6.10 (light to dark) show mostly well ordered results for the 1D tests. In contrast, the solutions for 3D are partly distorted for the Wendland kernels and the optimized distributions are almost randomly ordered for the triharmonic kernel. The low regularity score for these kernels might be one reason and again shows the difficulty to obtain well spread solutions for equally distributed preference weights.

This noise might not effect a coarse analysis of three different preference weights, but for the analysis of all 51 preferences we target an improved distribution of the compromise solutions. We discuss a deterministic gradient-based optimization in the next section in order to exclude the random effect of the employed evolutionary search.

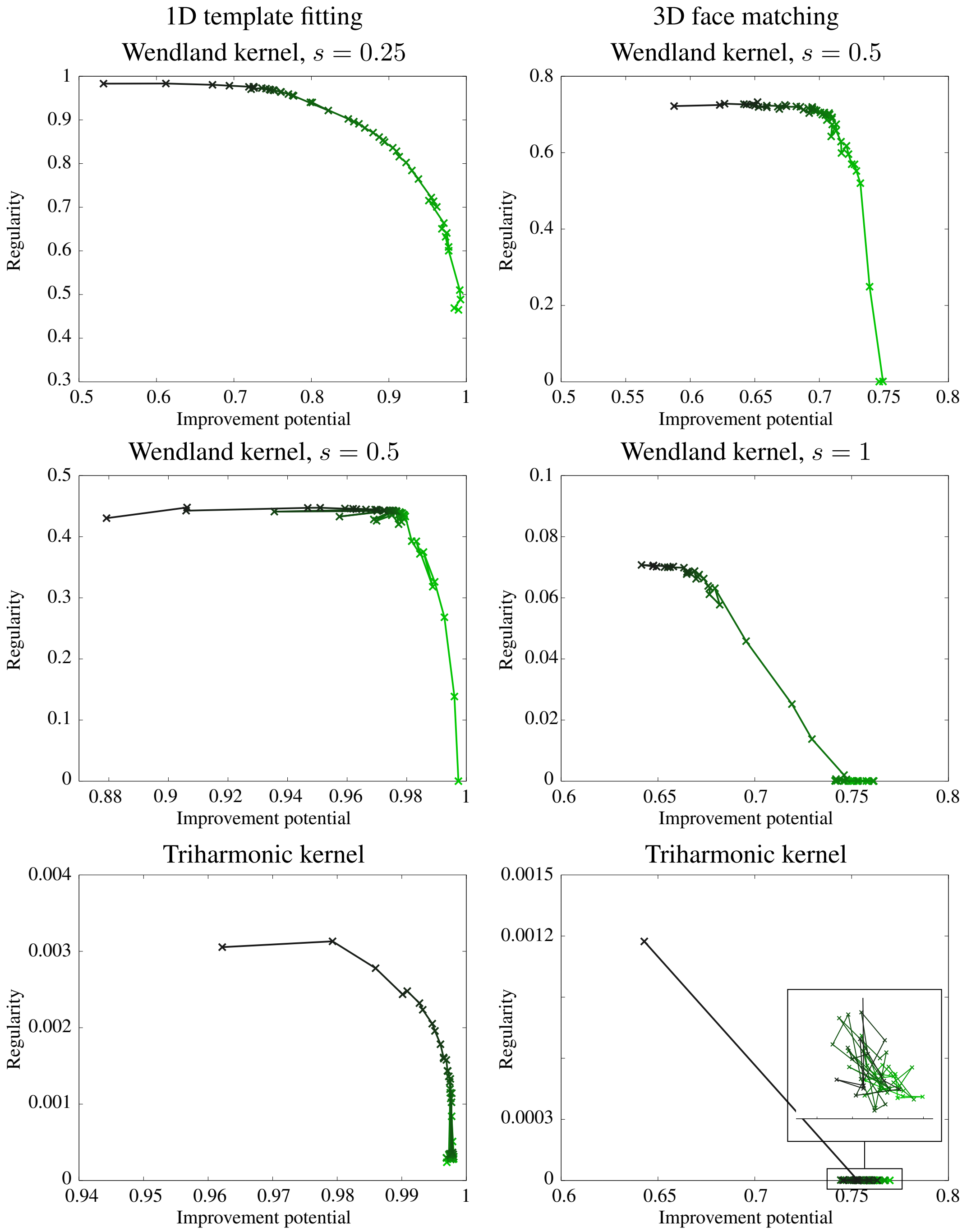


Figure 6.10: The solutions of an evolutionary single-objective optimization (CMA-ES) for 51 weights trading of between regularity and improvement potential. For the 1D scenario (left) 25 RBF centers and for the 3D scenario (right) 75 centers are optimized. The preference weight is color-coded from a focus on regularity (light green) to improvement potential (black).

6.4 Gradient-based Optimization

A non-deterministic evolutionary optimization for RBF center distributions leads to randomized results, which disturbs a fine-grained preference analysis. This motivates a deterministic gradient-based optimization, which is able to reduce this noise. Such an optimizer requires the gradient (see, e.g., [BV04]), which consists of the partial derivatives of the objective function (Equation (6.1)) with respect to the parameters, which are the x, y, z-coordinates of the RBF centers $\mathbf{c}_j$. Thus, we have to compute the partial derivatives of regularity and improvement potential (Equations (4.5), (4.6)). We add an additional distance function to constrain the centers to the desired design areas. Here, we just calculate the partial derivatives $\partial f_\lambda / \partial c_{j,x}$ for the x coordinates of the centers because the calculus of the other partial derivatives is analogue. We employ indirect manipulation for an easier computation of the partial derivatives such that the deformation matrix equals:

$$\mathbf{O}_{i,j} = \varphi(\|\mathbf{c}_j - \mathbf{x}_i\|_2) \ .$$

Note that the computation of partial derivatives for direct manipulation is possible by solving:

$$\mathbf{0} = \frac{\mathbf{\Psi}^{-1}\mathbf{\Psi}}{\partial c_{j,x}} = \frac{\mathbf{\Psi}^{-1}}{\partial c_{j,x}}\mathbf{\Psi} + \frac{\mathbf{\Psi}}{\partial c_{j,x}}\mathbf{\Psi}^{-1} \text{ for } \frac{\mathbf{\Psi}^{-1}}{\partial c_{j,x}} x.$$

First, we start with the calculus for regularity:

$$\frac{\partial R(\mathbf{O})}{\partial c_{j,x}} = \frac{\partial}{\partial c_{j,x}} \kappa^{-1}(\mathbf{O}) = \frac{\partial}{\partial c_{j,x}} \frac{\sigma_1}{\sigma_n}$$

where we compute the derivative of the condition number of a matrix $\mathbf{O}$ for which we require its singular value decomposition (SVD):

$$\mathbf{O} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$$

with $\mathbf{\Sigma}_{ii} = \sigma_i$ being the singular values ($\sigma_1 \leq \cdots \leq \sigma_n$), $\mathbf{U} = (\mathbf{u}_1 \ldots \mathbf{u}_n)$ being the matrix of left-singular vectors, and $\mathbf{V} = (\mathbf{v}_1 \ldots \mathbf{v}_n)$ being the matrix of right-singular vectors ($\mathbf{\Sigma} \in \mathbb{R}^{n \times n}, \mathbf{u}_i \in \mathbb{R}^m, \mathbf{v}_j \in \mathbb{R}^n$). Assuming $m \geq n$, which means the number of center is not larger than the number of vertices of the design, we compute the derivative of a singular value σ_k with respect to an entry of the matrix $\mathbf{O}_{i,j}$ following [PL00] as:

$$\frac{\partial \sigma_k}{\partial \mathbf{O}_{i,j}} = u_{ik} v_{jk} \ .$$

With the chain and quotient rule we obtain the partial derivative of regularity as:

$$\begin{aligned}
\frac{\partial R}{\partial c_{j,x}} =& \frac{\partial}{\partial c_{j,x}} \frac{\sigma_1}{\sigma_n} \\
=& \frac{1}{\sigma_n} \sum_{i=1}^{m} \frac{\partial \sigma_1}{\partial \mathbf{O}_{i,j}} \frac{\partial \mathbf{O}_{i,j}}{\partial c_{j,x}} - \frac{\sigma_1}{\sigma_n^2} \sum_{i=1}^{m} \frac{\partial \sigma_n}{\partial \mathbf{O}_{i,j}} \frac{\partial \mathbf{O}_{i,j}}{\partial c_{j,x}} \\
=& \frac{1}{\sigma_n} \sum_{i=1}^{m} u_{i1} v_{j1} \frac{\partial \mathbf{O}_{i,j}}{\partial c_{j,x}} - \frac{\sigma_1}{\sigma_n^2} \sum_{i=1}^{m} u_{in} v_{jn} \frac{\partial \mathbf{O}_{i,j}}{\partial c_{j,x}} \qquad (6.2)
\end{aligned}$$

with $\frac{\partial \mathbf{O}_{i,j}}{\partial c_{j,x}}$ being the derivative of the kernel function φ, which is computed as follows for the triharmonic and Wendland kernels (φ_{tri}, φ_W):

$$\begin{aligned}
\frac{\partial \mathbf{O}_{i,j}}{\partial c_{j,x}} &= \frac{\partial \varphi_{tri}(\|\mathbf{c}_j - \mathbf{x}_i\|_2)}{\partial c_{j,x}} \\
&= \begin{cases} 2\log((\|\mathbf{c}_j - \mathbf{x}_i\|_2) - 1)(\mathbf{c}_{j,x} - \mathbf{x}_{i,x}) & \text{for 2D domains} \\ 3\left\|\mathbf{c}_j - \mathbf{x}_i\right\|(\mathbf{c}_{j,x} - \mathbf{x}_{i,x}) & \text{for 3D domains} \end{cases} \\
\frac{\partial \mathbf{O}_{i,j}}{\partial c_{j,x}} &= \frac{\partial \varphi_{W}(\|\mathbf{c}_j - \mathbf{x}_i\|_2)}{\partial c_{j,x}} \\
&= \begin{cases} \frac{20}{s^2}(1 - \frac{1}{s}\left\|\mathbf{c}_j - \mathbf{x}_i\right\|_2)^3(\mathbf{c}_{j,x} - \mathbf{x}_{i,x}) & \text{for } \|\mathbf{c}_j - \mathbf{x}_i\|_2 < s, \\ 0 & \text{otherwise.} \end{cases}
\end{aligned}$$

Second, we compute the derivative of improvement potential. According to Equation (4.6) we would have to derive the pseudo-inverse $\mathbf{O}^+$ of a matrix. But we note that Equation (4.6) was computed from:

$$P(\mathbf{O}) = 1 - \min_{\mathbf{p}} \|\mathbf{O}\mathbf{p} - \mathbf{g}\|_2^2 =: 1 - \min_{\mathbf{p}} P^*(\mathbf{O}, \mathbf{p}) \ .$$

Thus, instead of computing the derivative of $P(\mathbf{O})$ with respect to the center positions we compute the derivative of P^* with respect to the centers and the parameters $\mathbf{p} = (p_1, \ldots, p_k)^T$. This leads to the following calculus:

$$\begin{aligned}
\frac{\partial P^*}{\partial \mathbf{p}} &= 2(\mathbf{O}\mathbf{p} - \mathbf{g})^T\mathbf{O} \\
\frac{\partial P^*}{\partial c_{j,x}} &= 2\sum_{i=1}^{m}\sum_{k=1}^{n}\left[\mathbf{O}_{i,k}p_k - g_i\right]\frac{\partial \mathbf{O}_{i,j}}{\partial c_{j,x}}
\end{aligned} \tag{6.3}$$

Third, in many design scenarios the center distribution is restricted to certain design regions leading to a constrained optimization problem for which we model soft constraints. Rectangular areas are sufficient for our test scenarios such that each coordinate is bounded from below and above, e.g., $\underline{\mathbf{b}}_j < \mathbf{c}_j < \overline{\mathbf{b}}_j$. For our tests the following exponential error function C for a constraints works well:

$$C(\mathbf{c}_j) = e^{\|\delta\mathbf{c}_j\|_2^2} - 1 \text{ with} \tag{6.4}$$

$$\delta\mathbf{c}_{j,x|y|z} = \begin{cases} \mathbf{c}_{j,x|y|z} - \overline{\mathbf{b}}_{j,x|y|z} & \text{if } \mathbf{c}_{j,x|y|z} > \overline{\mathbf{b}}_{j,x|y|z} \\ \mathbf{c}_{j,x|y|z} - \underline{\mathbf{b}}_{j,x|y|z} & \text{if } \mathbf{c}_{j,x|y|z} < \underline{\mathbf{b}}_{j,x|y|z} \\ 0 & \text{otherwise.} \end{cases}$$

The derivative of this function with respect to a coordinate of the center $\mathbf{c}_{j,x}$ equals:

$$\frac{\partial C(\mathbf{c}_j)}{\partial c_{j,x}} = 2 \cdot e^{\|\delta\mathbf{c}_j\|_2^2}\delta\mathbf{c}_{j,x} \ . \tag{6.5}$$

In the 3D scenario we restrict to the center distributions to the initial unit sphere. A 3D point $\mathbf{x}$ is parameterized with spherical coordinates, the angles α and β, as follows:

$$\mathbf{x} = \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} \cos(\alpha)\sin(\beta) \\ \sin(\alpha)\sin(\beta) \\ \cos(\beta) \end{pmatrix} .$$

Thus, the partial derivatives depend on the angles α, β resulting in an extended calculus. The derivative of the spherical coordinates is computed as follows:

$$\frac{\partial \mathbf{x}}{\partial \alpha} = \frac{\partial}{\partial \alpha} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} -\sin(\alpha)\sin(\beta) \\ \cos(\alpha)\sin(\beta) \\ 0 \end{pmatrix}$$
$$\frac{\partial \mathbf{x}}{\partial \beta} = \frac{\partial}{\partial \beta} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} \cos(\alpha)\cos(\beta) \\ \sin(\alpha)\cos(\beta) \\ -\sin\beta \end{pmatrix} .$$

And according to the chain rule $\partial \mathbf{O}_{i,j} / \partial c_{i,x}$ in Equation (6.2), (6.3) has to be multiplied with:

$$\frac{\partial c_{i,x}}{\partial \alpha_i} + \frac{\partial c_{i,x}}{\partial \beta_i} = -\sin(\alpha_i)\sin(\beta_i) + \cos(\alpha_i)\cos(\beta_i) \tag{6.6}$$

Note that $\partial \mathbf{O}_{i,j} / \partial c_{i,y|z}$ are computed analogue and that the domain of the angles is not bounded. This calculus is only applied for the 3D scenario.

For the 1D scenario we add the constraints leading to the following objective function:

$$f^*_\lambda(\mathbf{O}, \mathbf{p}) = \lambda R(\mathbf{O}) + (1-\lambda)(1 - P^*(\mathbf{O}, \mathbf{p})) + w_c \sum_{i=1}^{m} C(\mathbf{c}_j) \tag{6.7}$$

and its derivative (according to Equation (6.2), (6.3), (6.5)) for a gradient method to optimize the RBF center distribution for a given preference λ and a weight for the soft constraints w_c ($w_c = 100$ in our examples). Note that if centers violate constraints after an optimization procedure we project them to the valid area and compute the final fitness of this configuration. For the 3D tests we omit the constraints, but employ spherical coordinates through Equation (6.6) for the partial derivatives.

Like before, we evaluate the optimization of the RBF center distribution in the 1D test scenario first where we optimize centers in the plane for a preference weight. As we have shown in the previous section a smart initial center distribution constructed with the Lloyd and OLS sampling already improves the quality of an evolutionary optimization process thus is utilized as initialization here. We optimize centers for the preferences $\lambda = 0, 0.25, 0.5, 0.75, 1$ for a first basic analysis, and compare the gradient-based BFGS [Fle13] method with CMA-ES [AH12] for the

Table 6.1: Table: Comparison of the fitness and computation time (in seconds) of center distributions optimized with BFGS and CMA-ES. In 14 out of 15 tests BFGS results in higher fitness than CMA-ES on average. The BFGS results marked with $^+$ are better than the the best CMA-ES trials, whereas a $^-$ indicates a worse BFGS result than CMA-ES on average. The other case is marged with ^{o}.

	Wendland kernel (0.25)					
	BFGS		CMA-ES average		CMA-ES best	
Preference	Fitness	Time	Fitness	Time	Fitness	Time
$\lambda = 0$	0.99711^+	82	0.99519	82	0.99541	85
$\lambda = 0.25$	0.89456^+	132	0.89237	244	0.89329	257
$\lambda = 0.5$	0.87519^+	133	0.87212	226	0.87390	228
$\lambda = 0.75$	0.90791^+	89	0.90576	155	0.90895	170
$\lambda = 1$	0.98054^o	124	0.97610	111	0.98117	138
	Wendland kernel (0.5)					
$\lambda = 0$	0.99740^+	86	0.99643	51	0.99652	54
$\lambda = 0.25$	0.83669^o	233	0.83165	357	0.83981	388
$\lambda = 0.5$	0.70837^+	294	0.70332	329	0.70652	429
$\lambda = 0.75$	0.57481^o	307	0.57016	270	0.57486	240
$\lambda = 1$	0.44792^o	143	0.44716	262	0.45212	402
	Triharmonic kernel					
$\lambda = 0$	0.99229^-	251	0.99286	139	0.99392	153
$\lambda = 0.25$	0.74686^+	247	0.74466	95	0.74487	103
$\lambda = 0.5$	0.49741^o	187	0.49736	81	0.49750	89
$\lambda = 0.75$	0.24983^+	178	0.24969	93	0.24976	92
$\lambda = 1$	0.00326^o	570	0.00316	809	0.00334	1027

two compact Wendland kernels (support $s = 0.25$ and $s = 0.5$) and the global triharmonic kernel. For both optimization methods we use the shark library [IHMG08] with basic settings. For the CMA-ES we use the basic Equation (6.1) and add the constraints (6.4) as the fitness function and for BFGS we use the equivalent Equation (6.7) as fitness. 25 centers have to be optimized in the tests, which results in a 50 dimensional problem. Both optimization methods terminate if the average fitness of the setups of the last 50 iterations is only improved by less than 0.1%. Because the solutions of an evolutionary algorithm are partly random, we perform 5 runs for each preference weight and kernel for them.

The resulting fitness values and the computation time is shown in Table 6.1. Fitness values with a $^+$ point to a BFGS solution that has a better quality than any CMA-ES solution, values with a ^{o} mean that the solution of BFGS is better than the average of CMA-ES, and fitness

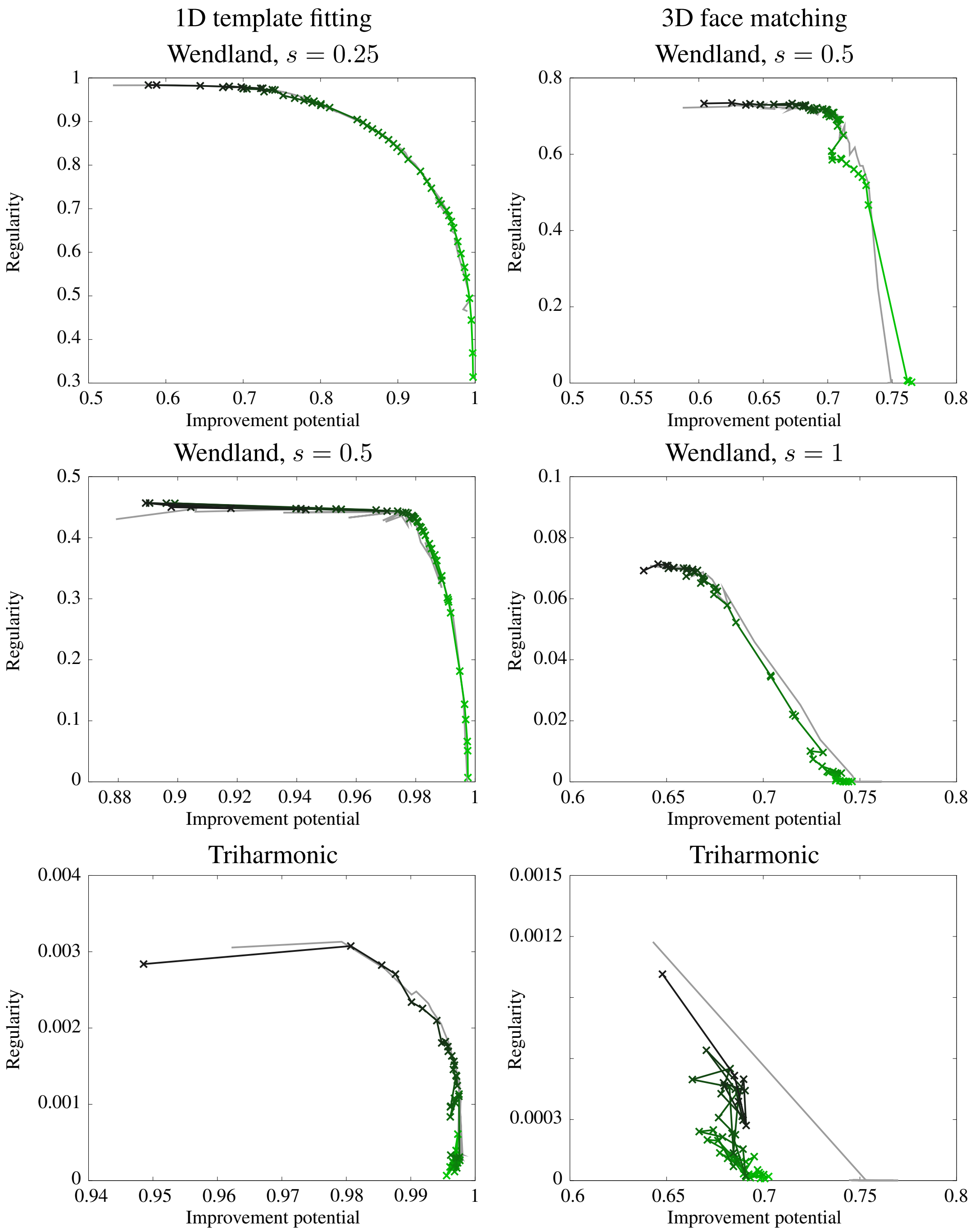


Figure 6.11: The solutions of a gradient-based single-objective optimization (BFGS) for 51 preferences weights trading of between regularity and improvement potential. The preference weight is color-coded from a focus on regularity (light green) to improvement potential (black) and the previously computed solutions of a evolutionary algorithms are depicted in grey.

values with $^-$ mark a BFGS solution that is worse than the average CMA-ES result. The table shows that the BFGS solver results in better setups than the CMA-ES on average in all but one test. Comparing the computation time reveals that the BFGS converges faster for the Wendland kernel with support 0.25. Figure 6.11 visualizes the quality of the resulting trade-offs between regularity and improvement potential. The main drawback of the CMA-ES is the clustering of solutions and the distortions of the order for preferences. The deterministic approach overcomes these issues only partly. For the 1D scenario (Figure 6.11 left) the curve of the BFGS solutions has higher quality (i.e., better solutions and less outlier) as the curve of the CMA-ES (shaded in grey, compare to Figure 6.10) for the Wendland kernels. For triharmonic kernels the resulting quality of both algorithms is similar in the 1D scenario. For the 3D scenario (Figure 6.11 right) the smaller cluster of solutions with high improvement potential motivates the application of BFGS. The solutions for triharmonic kernels are very noisy in 3D for both algorithms. The plots clearly demonstrate the existence of local optima, e.g., the CMA-ES with a focus on improvement potential results in higher scores than BFGS. Thus, we are well aware that the following preference analysis for the 3D test case, especially for the triharmonic kernel, might be very noisy. Nonetheless, we apply the deterministically generated compromises for the preference analysis in the next chapter because they are improved compared to the CMA-ES solutions although only by a small margin.

6.5 Summary

The initial representation setup is crucial for the performance of an evolutionary optimization process. We analyzed the generation of RBF deformation setups for evolutionary design optimization for two test scenarios. The concept of evolvability reveals powerful criteria for setups, namely variability, regularity, and improvement potential, to measure the expected performance of a setup. Regularity and improvement potential are conflicting targets, which we therefore analyze with a multi-objective optimization. As downside this optimization process has a runtime of 2 days for our comparatively simple test scenarios.

In real-world applications we rather aim for one optimal deformation setup with respect to a user-specified preference between regularity and improvement potential. Thus, we avoid the application of a multi-objective algorithm and employ a weighted single-objective version to construct compromise setups. We demonstrated the feasibility of such a weighted single-objective optimization. For some tests the quality of the deformation setup is even better than the quality of setups optimized with a multi-objective algorithm. This process is much faster, but it still runs for 2 hours for our simple problems. Furthermore, the single-objective optimization gets stuck in local optima in some of our tests.

In order to further improve computational performance and robustness we proposed and analyzed heuristics to generate setups. The regular setups constructed by Lloyd sampling are close to the Pareto front for local Wendland kernels. Even for direct manipulation where we lack the geometrical motivation, the regularity of the setup is significantly better than a random initialization. The Lloyd sampling even out-performs the evolutionary solutions in one example. Center distributions constructed with orthogonal least squares have high improvement potential and are on or very close to the Pareto front in all tests. Both methods reduce the computational effort from 2 hours to 1 minute.

Although our proposed combination of Lloyd and OLS sampling lacks the quality of truly optimized distribution of centers, they are efficiently constructed with these heuristics and thereby they are suitable for initializations. With these initializations we increased the resulting quality of an evolutionary optimization procedure.

But still, the random distortions resulting from a single-objective evolutionary algorithm distorts the order of solutions for different preferences. E.g., a solution optimized for high weight for regularity does not have a higher regularity score than a solution optimized for a minor weight for regularity. To overcome this issue we tested a deterministic gradient-based approach. This was successful in the 1D scenario. But for the more complex 3D template fitting the solutions are still distorted. This might negatively effect a detailed preference analysis for dynamic optimization scenarios in the next chapter.

7 Optimal Preferences for Design Optimization

The initial deformation setup has a strong impact on the performance of the optimization process as shown before. The setup's potential performance can be well estimated through our interpretation of the concept of *evolvability*, which we quantified by the three sub-criteria *variability*, *regularity*, and *improvement potential*. The concept of evolvability can not only be used for analyzing existing setups, but also for their initial *construction*. In this case, we (evolutionary) optimize the *deformation setup* itself. Since our shape deformation framework is based on RBFs, setup optimization means to determine where on the shape to place RBF kernels. The inherent conflict between regularity and improvement potential can be resolved by letting the designer choose a compromise weight and performing a weighted single-objective optimization with respect to these two objectives.

Ultimately, for a high-performing design optimization process, the deformation representation has to be able to *adapt* to dynamic environments, such as, e.g., varying fitness environments induced by varying angles of attack in a CFD simulation. In dynamic environments the interplay between *exploration* and *exploitation*, which is covered by evolvability, is key for high-performance as motivated in Chapter 3. Improvement potential defined as the potential of the deformation setup to exploit extracted (gradient) information clearly refers to exploitation. But the quite general, information-independent definition of exploration [CXP$^+$09] leaves room for interpretation: variability as well as regularity can be associated with this. But, our definition of variability in Chapter 4 is independent of the RBF center distribution, such that any deformation setup has equal variability for a fixed amount of even different kernel functions. In contrast, regularity-optimal RBF setups are uniform for indirect RBF manipulation. This fits a genotype-based definition of exploration where a wide-spread genotype (large distances between the centers in our example) characterizes exploratory nature. Thus, we link regularity with exploratory capabilities of the deformation setup.

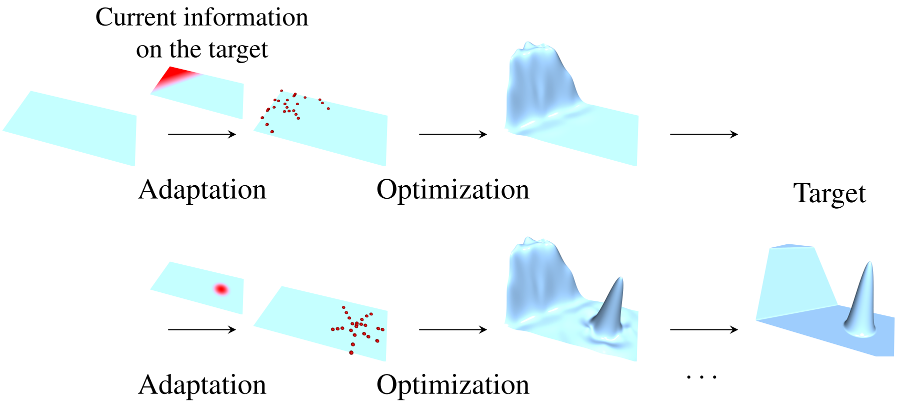


Figure 7.1: Example of a stepwise design optimization process where the representation (RBF kernel distribution, red dots) is adapted in its domain (plane) according to information on the target (color coded estimation of the fitness gradient).

We first perform the next step towards a truly dynamic evolutionary design optimization: In a static design optimization (non-varying fitness function or design target) we alternate between setup optimization/construction (*where to place RBF kernels?*) and design optimization (*which RBF coefficients to use for shape deformation?*). As illustrated in Figure 7.1 we split the design optimization process into the two parts, an adaptation phase, i.e., the construction of the setup, and an optimization phase of the design.

Tuning a deformation setup towards improvement potential requires a rough estimate of the fitness function's gradient. While in later adaptation stages this information can be extracted from previous design optimization phases (see Figure 7.1), the gradient estimate for setting up the initial representation typically comes from expert knowledge. Both gradient information might be out-dated or inaccurate. Hence, the deformation setup must be able to exploit accurate information while at the same time be robust against inaccurate information. We analyze different preference weights for setup optimization and show that an intermediate preference between regularity and improvement potential meets these two requirements. These results are published in [RMB17].

In the second part of this chapter we extend the coarse preference analysis to a fine-grained one for dynamic scenarios. Based on the strength of a change in the fitness function a designer has to gauge exploration (regularity) and exploitation (improvement potential) accordingly. E.g., in an almost static environment a focus on exploitation should be chosen whereas for a strongly changing environment exploration should be the choice. We first measure the strength of the change in the fitness function and then analyze optimal preferences to construct the deformation setups for several classes of shifts ranging from almost static environments to strongly changing ones.

But, the strength of a dynamic shift can be interpreted differently as reliability/accuracy of information. If the information about a fitness environment is reliable or accurate, such that a designer trusts this information, then it should be exploited for the construction of the deformation setup. But, if the fitness information is not reliably or inaccurate an exploratory approach is more appropriate.

7.1 Static Design Loop

Not only a dynamic problem, but also a static one is enhanced by a deformation representation that can react and adapt to gathered information. We alternate between the optimization of the deformation setup (adaptation phase) and the optimization of the design (optimization phase) as a first step towards a dynamic optimization, similar to [Aul11, OJS01, YSTY16]. To evaluate the exploitation potential we require information about the targeted design (i.e. the fitness function). Initially, we employ a rough manual estimation of the target design. Otherwise, we extract such information during the design optimization process without an additional data mining process. Because exploration and exploitation are competing targets we analyze different preferences to weight them, like in [ANMD16], where three different preferences (0, 0.5, and 1) are analyzed for competing targets in topology optimization.

During a non-converged design optimization process we expect that a previously successful variation, i.e., the difference between the design before ($\mathbf{X}$) and after ($\mathbf{X}'$) some optimization steps, will be successful in the next steps, too. Thus, we evaluate the results of a design optimization after a fixed number of iterations k (shown in Figure 7.2) and use the difference between the vertex positions $\mathbf{x}_i$ of the initial design and the result after k iterations $\mathbf{x}'_i$ to compute a new estimation of the gradient (following the approach in [GMH$^+$08]). Without loss of generality we assume a one dimensional deformation of the n points. Thereby, one coordinate x_i of each vertex $\mathbf{x}_i$ is deformed and we define the estimation of the gradient $\mathbf{g} = (g_1, \ldots, g_n)$ as:

$$\hat{\mathbf{g}}_i = (x'_i - x_i)^3, \qquad \mathbf{g} = \frac{\hat{\mathbf{g}}}{\|\hat{\mathbf{g}}\|_2}$$

Note that for deformations of higher dimension the gradient becomes a Jacobian matrix. In this case we apply the above formula for each dimension/coordinate and use the Frobenius norm for normalization. We compute cubic differences for three reasons: First, they preserve the sign of the displacements. Second, linear differences as displacements can be exactly reproduced with the previous center distribution. Hence, a focus purely on improvement potential with these displacements as the estimated gradient would choose the previous centers for the next phase, which would keep the centers fixed. The cubic leads to gradient estimations that cannot be exactly approximated by the old center distribution, thus leading to a varied distribution. Third, during a design optimization process the fitness gain of well-shaped design regions may dominate a small loss of fitness resulting from poorly shaped regions. The resulting deformations in poor regions are smaller than in well-shaped regions, but they still would be incorporated into the new gradient. Thus, the cubic power scales down the small displacements of bad regions and emphasizes larger displacements of well-shaped regions.

The computation of the gradient based on the last k iterations allows us to intentionally forget information. If the old information still was good then the new gradient will be similar to the

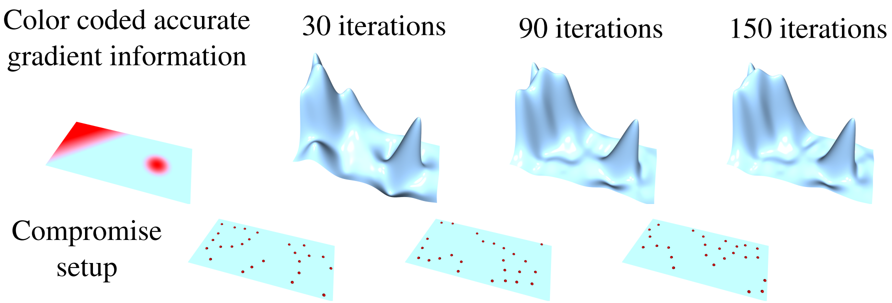


Figure 7.2: Stepwise design optimization initialized with accurate gradient information in the beginning. A compromise preference between improvement potential and regularity is chosen for setup adaptation, leading to good fitting quality and convergence speed.

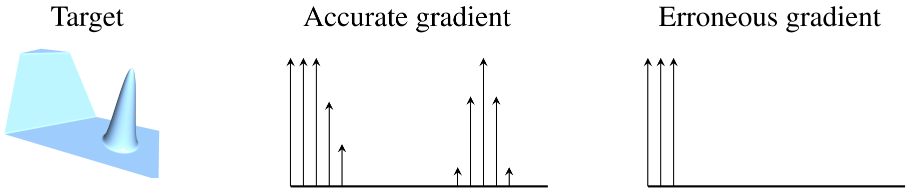


Figure 7.3: Sketch of an accurate gradient direction derived from the target and an erroneous gradient which completely ignores the peak in the target.

old one, such that there will not be a negative effect. But old information may dominate new one, e.g., the changes in the design are rather large in the beginning of an optimization process compared to later stages. This effect is avoided when analyzing just the last k iterations. In our tests we set k to 30 (for 1D function approximation) and 50 (for 3D template fitting) to reduce random effects in the beginning of a design optimization process.

First, we start with the stepwise design optimization for the 1D height field approximation scenario, where we evaluate the three different preference weights ($\lambda = 0$, 0.5, and 1). The design optimization process alternates between setup construction and optimization of the fit. Like before, we distribute 25 RBF kernels with two coordinates with a (7,15)-CMA-ES for the construction of the setup. For the following fitting we optimize the 25 RBF parameters with a (6,13)-CMA-ES. The number of parents and offspring results from the default settings of the shark library. We perform the fitting for $k = 30$ iterations and alternate with the setup construction five times, which we denote as one test (see Figure 7.2). Each test took approximately 40 minutes on an Intel Xeon, $8 \times 3.60\,\text{GHz}$, with 8 GB of memory.

To reduce the random effects of the optimization of the setup and the optimization for the fit we perform five tests for each of the three kernel functions (Wendland kernel with support radius 0.25 and 0.5, and the global triharmonic kernel) and each of the three preference weights ($\lambda = 0, 0.5, 1$). First, we analyze the optimization process with exact gradient information for the initial setup generation (Figure 7.2). But in a real-world optimization scenario such accurate information typically is not available. To simulate such a scenario we intentionally feed the initial setup generation with an erroneous gradient estimation. In Figure 7.3 we sketch such an estimation compared to the accurate gradient direction.

Table 7.1: The regularity for the setups during a design optimization process. The very low values (red) of a setup purely optimized for improvement potential with a triharmonic kernel characterize a slow converging optimization process.

	Preference	Iteration				
		0	30	60	90	120
Wendland kernel, $s = 0.25$	$\lambda = 0$	0.486	0.244	0.151	0.054	0.152
	$\lambda = 0.5$	0.862	0.787	0.735	0.690	0.767
	$\lambda = 1$	0.974	0.977	0.978	0.979	0.977
Wendland kernel, $s = 0.5$	$\lambda = 0$	0.268	0.105	0.098	0.075	0.089
	$\lambda = 0.5$	0.618	0.757	0.640	0.625	0.687
	$\lambda = 1$	0.824	0.848	0.813	0.813	0.809
Triharmonic kernel	$\lambda = 0$	0.052	0.041	0.031	0.028	0.029
	$\lambda = 0.5$	0.578	0.580	0.374	0.536	0.444
	$\lambda = 1$	0.596	0.596	0.596	0.597	0.596

In the case of accurate gradient information we expect that a preference $\lambda = 0$, i.e., a focus on exploitation, results in the best fitting values because the resulting center distribution is adapted to best fit the features (plateau and peak) of the target. A distribution tuned for regularity ($\lambda = 1$) might explore the design space more, but has centers in already optimal regions. An intermediate preference ($\lambda = 0.5$) combines both, the potential to exploit information and to explore.

For the Wendland kernel with small support, the intermediate preference results in fitting values almost as good as for $\lambda = 0$ (Figure 7.4, top left). For the larger Wendland kernel, the intermediate preference is on par with a preference on improvement potential (Figure 7.4, left), and for triharmonic kernels it even out-performs this preference (Figure 7.4, bottom left). As shown in Table 7.1, a preference of 0.5 results in setups with good regularity values, significantly better than a preference of 0 and on a similar level with a preference of 1. Because regularity characterizes the convergence speed, a triharmonic kernel in combination with a preference of 1 converges slowly. An intermediate preference guarantees both: a good fitting quality (exploitation) and good regularity with the potential to explore a design.

In complex design optimization scenarios we have to base the estimation of the gradient information either on designer's input according to an expected target or on data from previous tests. For example, in Figure 7.5 we set the estimated gradient according to a variation of the plateau that a designer manually specified, intentionally ignoring the peak in the middle. In such scenarios the optimization process should be able to explore further design regions. But a focus purely on improvement potential/exploitation would relentlessly construct center distributions according to this misleading estimation of the gradient.

For example, in Figure 7.5 (top) the centers are only placed near the plateau during the whole design optimization process. Choosing a compromise between improvement potential and regu-

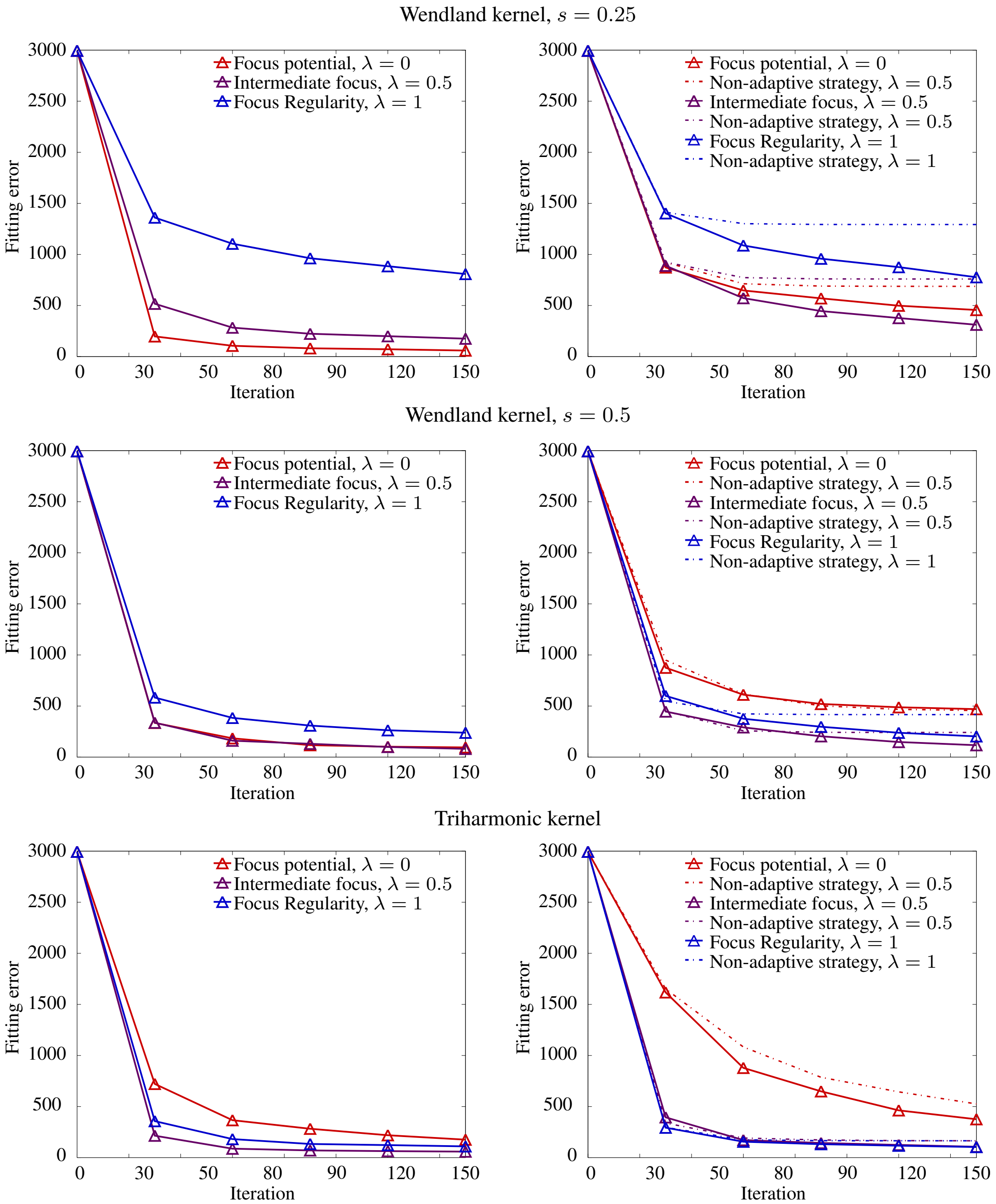


Figure 7.4: Convergence plot for the stepwise optimization procedure for the 1D scenario initialized with accurate gradient information (left) and an estimated version (right). Accurate information and an intermediate preference (solid purple lines) results in good fitting quality for Wendland kernels (top and center, left) and out-performs a pure focus on exploitation for triharmonic kernels (bottom left). An adaptive process (solid lines, right) performs better than a non-adaptive process (dashed lines, right). Choosing an intermediate preference (solid purple lines, right) handles the coarse estimation of the gradient and out-performs a pure preference focus on improvement potential (solid red lines, right) or regularity (solid blue line, right).

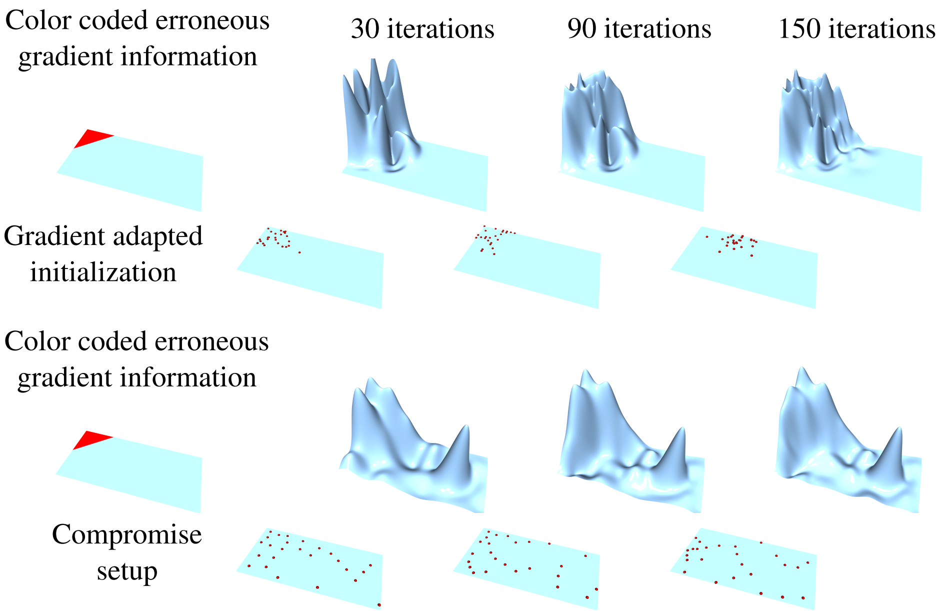

Color coded erroneous gradient information
30 iterations
90 iterations
150 iterations
Gradient adapted initialization
Color coded erroneous gradient information
Compromise setup

Figure 7.5: Design optimization with an erroneous estimation of the initial gradient given by a designer (left). Top: Adapting purely to improvement potential "exploits" the misleading erroneous gradient information, resulting in fits of low quality. Bottom: A compromise between improvement potential and regularity manages to repair the initially misleading gradients.

larity results in center distributions that are more spread on the domain and thereby can explore the design space better. Thus, the estimation of the gradient extracted during the next design optimization phase is more accurate for the following adaptation phase. As a consequence, the center distribution has a higher quality and results in improved designs, as can be seen in Figure 7.5, bottom.

In Figure 7.4 (right) we plot the average fitting values of the five tests for the three kernels and three preferences. A compromise preference (the purple solid line) out-performs a focus on either regularity or improvement potential for the Wendland kernels and is as good as a focus on regularity for the triharmonic kernel. This shows that a compromise between exploration and exploitation can repair initially misleading gradient estimations.

Moreover, we compare the stepwise optimization process, which alternates setup adaptation with shape optimization, to a non-adaptive optimization procedure. For the latter we construct the deformation setup just once in the beginning, according to the estimated gradient, and omit the following adaptation phases. As expected, the stepwise optimization performs better than the non-adaptive one (see Figure 7.4, dashed lines, right), which demonstrates its benefit for design optimization.

Because the 1D fitting of height fields is a rather simple test scenario we increase complexity in a 3D template fitting procedure (as we did before). The goal in this design optimization scenario is to fit a template (sphere) to a scanned face (compare Chapter 5). Each vertex of the design has three degrees of freedom and exact gradient information is not available. In

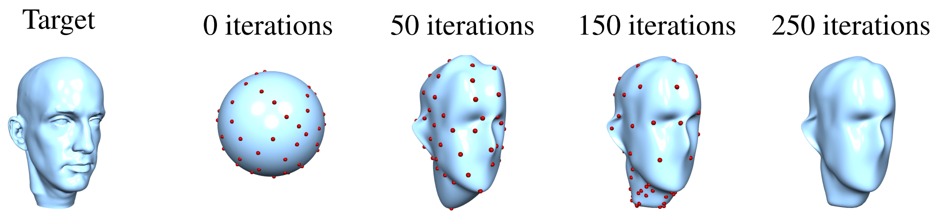


Figure 7.6: Example results of an adaptive template fitting process where the RBF kernels are adjusted on the deformed mesh with a compromise preference between regularity and improvement potential, i.e., between exploration and exploitation.

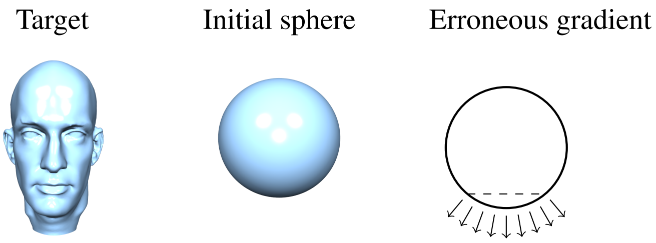


Figure 7.7: As erroneous gradient information for template fitting we use displacements just on the bottom of sphere (right) to initialize the adaptation of the kernel distribution.

contrast to the 1D function approximation scenario we distribute the kernels on the initial unit sphere or its deformed state. Furthermore, we increase the number of kernels from 25 to 75 to obtain plausible fits resulting in 150 parameters to be optimized for setup construction. In the design optimization phase we optimize 225 parameters because each vertex has three degrees of freedom. With the default settings of the shark library this leads to a (9,19)- and (10,20)-CMA-ES, respectively. Like in the function approximation scenario we alternate between setup construction and design optimization five times which took approximately three hours for the whole stepwise optimization due to the increase in complexity. We perform the fitting procedure for $k = 50$ iterations, exemplarily shown in Figure 7.6. To reduce random effects we perform the optimization process three times for three preferences, a compact Wendland kernel and the global triharmonic kernel.

As in the previous 1D test scenario we want to show that an intermediate weight ($\lambda = 0.5$) between improvement potential and regularity to adapt the kernel distribution overcomes bad initial estimations of the gradient and out-performs an adaptation strategy with a preference either set for improvement potential ($\lambda = 0.0$) or regularity ($\lambda = 1$). We intentionally construct erroneous gradient information by utilizing displacements in normal direction just on the bottom of the sphere (Figure 7.7). The convergence plots (Figure 7.8) show the same trend as in the function approximation scenario, namely that the stepwise design optimization with an intermediate preference weight between improvement potential and regularity repairs a low-quality estimation of the gradient and performs better than an adaptation strategy that either sets the kernels to obtain maximal regularity or sets them to obtain maximal improvement potential. An intermediate preference allows for a plausible fit of the whole scan and deforms important regions more accurately (Figure 7.9), thereby leading to an improved design.

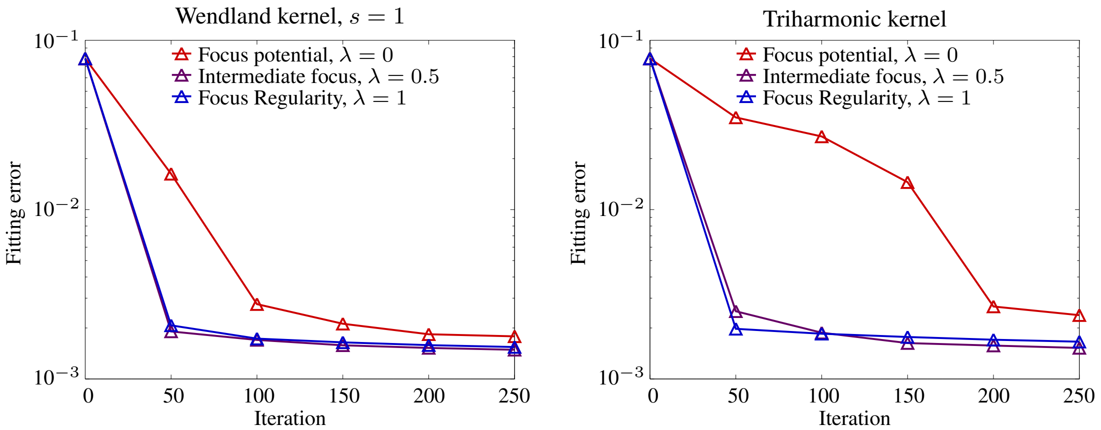
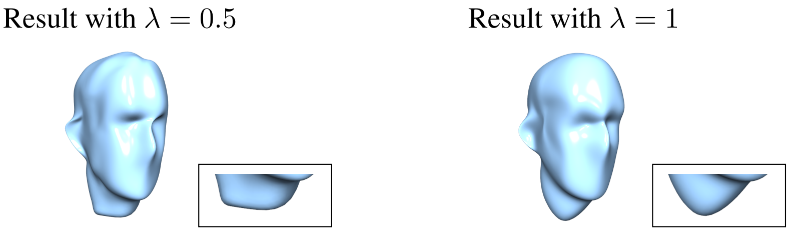


Figure 7.8: Convergence plot for the adaptive optimization procedure of the 3D template fitting. Choosing an intermediate preference (purple line) to setup the deformation out-performs a preference focus on regularity ($\lambda = 1$) or improvement potential ($\lambda = 0$).


Result with $\lambda = 0.5$

Result with $\lambda = 1$

Figure 7.9: Using an intermediate preference (left) between regularity and improvement potential to adapt setups is robust to erroneous initial gradient information and utilizes extracted information of the design optimization phase to improve the fit in an important region. In contrast, adaptation purely focusing regularity results in a plausible fit but neglects the extracted information (right).

With the advantages of an intermediate preference weight for a stepwise optimization process in mind we conduct a more detailed preference analysis for dynamic scenarios in the next section.

7.2 Dynamic Design Loop

A proper reaction to different dynamic changes is required to handle dynamic design problems efficiently. Here, we address how to describe the strength of a dynamic change and then we discuss the proper construction of the deformation setup which means choosing the best preference for trading-off between regularity and improvement potential.

To simulate dynamic changes in the 1D scenario we parameterize and vary the target height field $\mathbf{s}(t) = (s_t(x_1, y_1), \ldots, s_t(x_k, y_l))$ of [GJS12]. As depicted in Figure 7.10 the target height field consists of a plateau, a slope, and a bump. We parameterize the plateau with a height parameter h_1, and two variable points $\mathbf{P}_1, \mathbf{P}_2$ forming the edge. In addition to the edge $\mathbf{P}_1, \mathbf{P}_2$ we chose

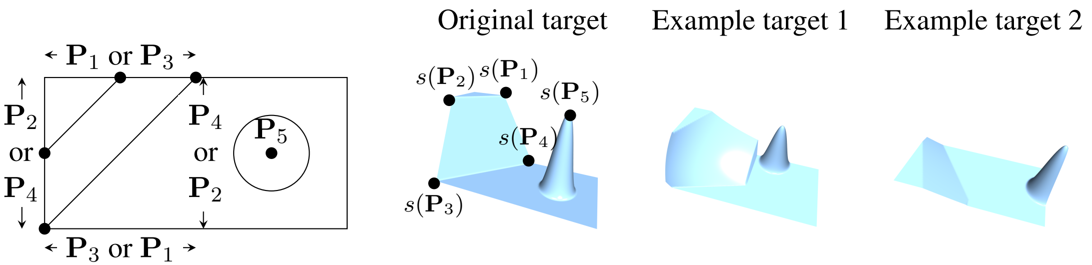


Figure 7.10: Example of a parameterized target height fields. The plateau at $\mathbf{P}_1, \mathbf{P}_2$, the slope between $\mathbf{P}_1\mathbf{P}_2, \mathbf{P}_3\mathbf{P}_4$ and the bump at $\mathbf{P}_5$ can be varied.

$\mathbf{P}_3, \mathbf{P}_4$ to bound the area of the slope. A point of the slope is defined as interpolation of the nearest point on the plateau and the "0"-level. The free parameters for the bump are its center $\mathbf{P}_5$, its radius r, and its height h_2. Choosing the parameters randomly as follows:

$$\mathbf{P}_1 \in ([0,1],1)^T, \mathbf{P}_2 \in (0,[0,1])^T, \mathbf{P}_3 \in ([0,1],0)^T, \mathbf{P}_4 \in (1,[0,1])^T, \mathbf{P}_5 \in ([1,2],[0,1])^T$$
$$h_1, h_2 \in [0,1] \quad r \in [0.1,.5]$$

and including the possibility that $\mathbf{P}_1$, $\mathbf{P}_3$ and $\mathbf{P}_2$, $\mathbf{P}_4$ can be flipped results in varying targets from almost congruent to almost distinct. Flipping these points simply defines the corner of the plateau $\mathbf{C}_{12} = (P_{2,x}, P_{1,y})^T$ which leads to the following definition of the height field (pseudo-code):

$$s(x,y) = \begin{cases} h_1 & \text{if } x \leq 1, (x,y) \in \triangle \mathbf{P}_1\mathbf{P}_2\mathbf{C}_{12} \\ \frac{d_{12} \cdot h_1}{d_{12}+d_{34}} & \text{if } x \leq 1, (x,y) \in \square \mathbf{P}_1\mathbf{P}_2\mathbf{P}_3\mathbf{P}_4 \text{ and} \\ & d_{ij} \text{ equals the distance of the segment } \mathbf{P}_i\mathbf{P}_j \text{ to } (x,y)^T \\ h_2 & \text{if } \left\|\mathbf{P}_5 - (x,y)^T\right\|_2^2 < r^2 \\ 0 & \text{otherwise} \end{cases}$$

Given this parameterization the fitness as the least squares distance of a deformed state $\mathbf{x} = \mathbf{Op}$ of the plane to the changing scalar field $s(t)$ can change over time:

$$f_t(\mathbf{p}) = \sum_{i=1}^{m} (o_\mathbf{p}(\mathbf{x}_i) - s_i(t))^2 = \|\mathbf{Op} - \mathbf{s}(t)\|^2 \tag{7.1}$$

The definition of improvement potential (Equation (4.6)) requires an estimation of the fitness gradient which can be computed analytically for the given fitness:

$$\mathbf{g}_t = \frac{\mathbf{Op} - \mathbf{s}(t)}{\|\mathbf{Op} - \mathbf{s}(t)\|} .$$

Note that in more complex scenarios the gradient can be estimated as the difference between two deformed states $\mathbf{X}_1, \mathbf{X}_2$ (in the Frobenius norm).

The similarity between two fitness gradients $\mathbf{g}_1, \mathbf{g}_2$ is an indicator for the strength of a changing dynamic. If both gradients are similar we assume no change in the dynamic, but if they vary

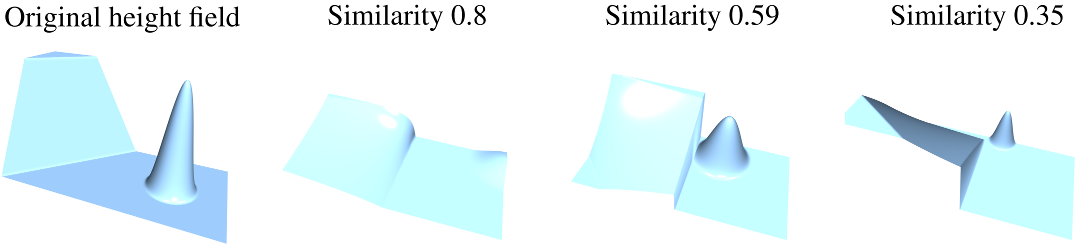


Figure 7.11: Randomized height fields compared to the original one. The similarity score measures the similarities between the fitness gradients for the original target and for the random ones.

heavily then the fitness might have changed such that the deformation setup has to be adjusted. We choose the dot product between the gradients as a measure of similarity:

$$\mathrm{sim}(\mathbf{g}_1, \mathbf{g}_2) = \frac{|\langle \mathbf{g}_1, \mathbf{g}_2 \rangle|}{\|\mathbf{g}_1\| \, \|\mathbf{g}_2\|} \ .$$

For the computation of the dot product in the 3D scenario we vectorize the gradient matrices and employ the Frobenius norm for normalization. In comparison to a p-norm, e.g., the Euclidean distance employed in [SZPJ13], this definition is scale-invariant which makes it suitable for linear deformations. If a deformation is able to approximate a vector then it can approximate any scaled version (even negatively scaled) with identical quality. A similarity of “1” means that the gradients are equivalent and thereby no change in the fitness occurred whereas a similarity of “0” describes orthogonal and thereby most distinct gradients, which indicates the most drastic change in the dynamic. But this gradient-based measure tempts to be unintuitive for a designer due to the scale-invariance. E.g. we would rank the height field in Figure 7.11, center right, higher than the left one.

However, we assume that if the dynamic of a scenario remains similar it is not necessary to adapt the deformation setup whereas the setup has to adapt to handle dynamic changes. By investigating the correlation between the similarity of the fitness before and after a dynamic change and the optimal choice of a preference to adjust the setup in test scenarios we obtain information for possible generalizations of the optimal adaptation settings in more complex scenarios.

Again, we mention the alternative interpretation of the similarity measure as reliability of information. A designer might have information about the standard height field, but the expected target is different, such that the original information is not reliable to a certain degree. Thus, an optimal preference weight to construct the deformation setup according to this derivation is required.

For the analysis we utilize the compromise setups for 51 preferences ($\lambda = 0, 0.02, 0.04, \ldots 1$) computed with BFGS in the previous chapter (Figure 6.11, left). These setups are constructed for the standard height field. A changing fitness is simulated with a new and randomly generated target, for which we compute the similarity to the standard one, and for which we test each setup. An analytic template fitting procedure as described in Chapter 5 enables the fitting of 10000 random targets which ensures to cover almost the whole range of similarities. Hence, each of the plots for the three tested kernels, the compact Wendland kernels with support 0.25 and 0.5 and the global triharmonic kernel, in Figure 7.12 (left) consists of 51000 data points. For each random target (fitness) we rank the preferences according to their success: a preference whose setup results in a better fit is ranked higher than a preference whose setup results in a worse fit. This is an appropriate approach because the different targets result in different optimal fitting scores, whereby they cannot be compared. We obtain a color coding of the rankings by averaging over each interval of similarities. E.g., the rankings of all tested targets with similarities in $\left[\frac{k-1}{20}, \frac{k}{20}\right]$ ($k = 1, \cdots, 20$ in the 1D scenario) or $\left[\frac{k-1}{10}, \frac{k}{10}\right]$ ($k = 1, \cdots, 10$ in the 3D scenario) are averaged. White indicates a very good similarity–weight relation and black refers to the worst relation for the different preference/target combinations in Figure 7.12.

The plot of the Wendland kernel with support 0.25 (Figure 7.12, top left) visualizes a strong linear correlation (indicated by the "white diagonal") between the similarity of gradients and the most successful weights with a significant Pearson correlation coefficient $r = -0.63$. In contrast for the Wendland kernel with support 0.5 (Figure 7.12, center left) a linear relation tends to exist only for similarities larger than 0.8. For similarities lower than 0.8 the relation is constant and a preference focus on regularity seems to be most successful. The global triharmonic kernel has a similar constant relation between similarity and preference weights (Figure 7.12, bottom left) but with a larger constant range up to 0.9. Hence, the overall linear correlation for the global Wendland and triharmonic kernel is only very weak ($r = -0.22$, $r = -0.1$ respectively).

Based on these results we can only emphasize a focus on improvement potential (exploitation) for weakly changing environments, characterized by high similarity scores between the fitness gradients. In contrast, a focus on regularity (exploration) is superior for a strong change in the fitness. But, the transition between an optimal focus of regularity and improvement potential is different for the different kernel functions. For further investigation we conduct the tests in the more complex 3D scenario.

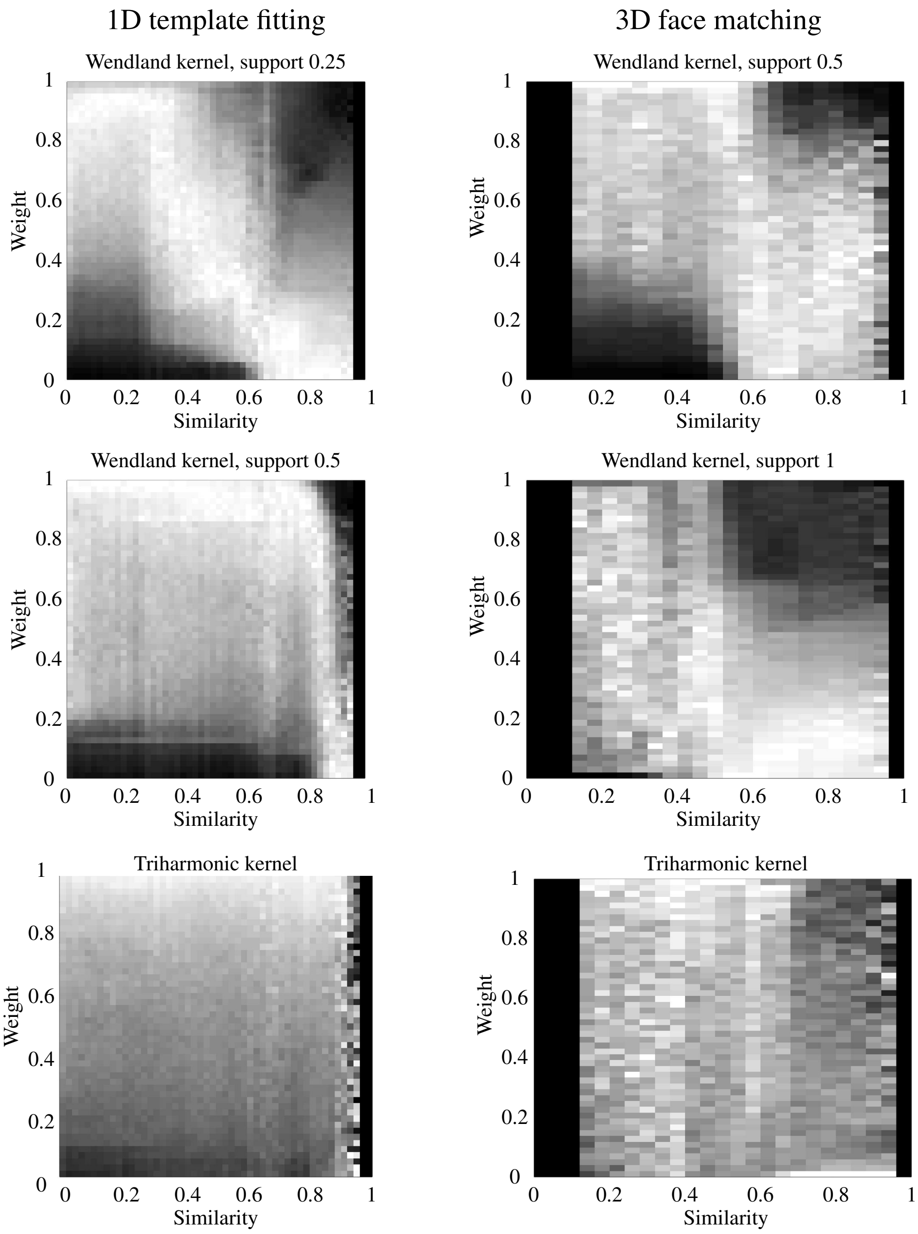
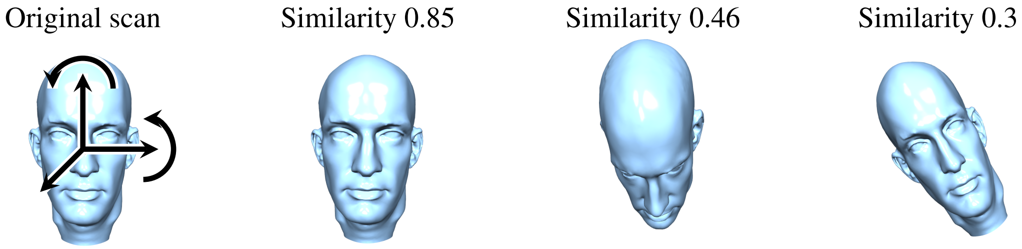


Figure 7.12: Optimal preference weights (highlighted in white) for 1D template fitting (left) and 3D face matching (right). For drastic changes in the fitness function (small similarity score) a focus on regularity (weight of 1) is superior. In contrast, for a small changes in the fitness function (high similarity score) a focus on improvement potential (weight of 0) is superior. But the range where and how to switch between the compromises is imprecise. For the 3D scenario the results are more noisy (especially the triharmonic kernel) because of a lower quality of the optimal compromise setups for the preference weights.



Original scan Similarity 0.85 Similarity 0.46 Similarity 0.3

Figure 7.13: Example modifications of a scanned face as the design target and the similarity scores with respect to the gradient of the original target.

Additionally to the dynamically changing correspondences in the 3D fitting scenario we vary the target scan by scaling the vertices in x,y,z direction (between 90% and 110%) and apply a rotation around the y and z axis (by up to 40°), e.g., Figure 7.13.

For the analysis of a relation between the similarity of a dynamic change and the preference weights to set up the center distribution we employ the constructed compromise setups shown in Figure 6.11 (right). We perform 1000 tests for 51 preference weights with 75 centers and three kernel functions, Wendland kernels with support 0.5 and 1 and a triharmonic kernel. Because of the more complex fitting procedure one test required approximately 20 minutes computation time compared to 10 seconds for the simpler 1D scenario.

But, the results are very noisy as shown in Figure 7.12 (right). The range of similarities (larger than 0.6) with a beneficial focus on exploitation seems to be larger than in the 1D approximation scenario (compare to Figure 7.12, left). Furthermore, this range is roughly similar for the three kernels. For the Wendland kernel with support 0.5 the plot in Figure 7.12 (top right) supports the assumption that the lower the similarity the better is a focus on regularity. In contrast for the Wendland kernel with support 1 (Figure 7.12, center right) there seems to be no optimal preference for similarities lower than 0.6. This tendency is even stronger for the global triharmonic kernel (Figure 7.12, bottom right).

In total the resulting plots for the 3D scenario in Figure 7.12 (right) are very noisy compared to the results of the function approximation. This has several reasons: First, the target geometry is more complicated resulting in noisy center distributions for the different preference weights especially for the triharmonic kernel (compare Figure 6.11). Second, the gradient information is only accurate in the beginning of the fitting procedure. Thus, we are working with inaccurate gradients anyway. Third, the "dynamic" fitness, constructed through a random target, is truly dynamic due to the changing correspondences during the fitting process, which causes noise. Forth and most important: regularity-optimal center distribution can accidentally fit a random target properly and distributions tuned for improvement potential can accidentally have very good regularity scores. This holds for the 1D test case, too, but it is especially true for the 3D scenario where center distributions with good improvement potential are almost uniformly spread over the initial sphere, which has to be deformed almost everywhere. Consequently, highly regular uniform center distributions result in good fits.

These facts disturb the analysis such that especially for the triharmonic kernel only a weak preference recommendation can be given for the 3D scenario. For a small change in the fitness function a focus on improvement potential (exploitation) is superior. For a strongly changing fitness a focus on regularity (exploration) is beneficial for the Wendland kernel with support 0.5, whereas almost all preference weights seem to be equivalent for the Wendland kernel with support 1 and the triharmonic kernel. However, the plots in Figure 7.12 show that an intermediate preference weight ($\lambda = 0.5$) works fine for all environments. Such a preference weight always had at least a medium ranking (grey color), such that we recommend its application if neither a static nor a heavily changing environment is detected.

Note, that we perform such a preference analysis again for an alternative formulation of the variability criterion in the next section. Although the utilized compromise setups have a better quality with the new definition, the preference analysis points to equivalent results.

7.3 Summary

The construction of optimal trade-offs for design representation is crucial for the performance of an evolutionary optimization process. With regularity and improvement potential as conflicting targets for the representation setups we cover exploration and exploitation capabilities of the setups. For these two criteria we targeted the optimal trade off for dynamic environments.

First, we analyzed a static and stepwise design optimization process as a first step. We showed in the two test scenarios, fitting of 1D height fields and fitting of 3D face scans, that an intermediate preference weight between regularity and improvement potential to construct a RBF deformation setup performs better than setting the preference to either construct highly regular setups or setups with optimal improvement potential. For our test cases, a compromise includes both: the potential to exploit information (via improvement potential) and the potential to explore the design (obtained indirectly from regularity). A compromise setup induces robustness towards imprecise or erroneous information in the beginning of an optimization process and it enables the exploitation of gathered information during the optimization process.

In the second step, we extended the coarse preference analysis to a fine-grained one for dynamic scenarios. We employed a similarity measure based on the (approximated) fitness gradient to quantify the strength of a dynamic change. In 1D template fitting a similar or static environment requires a focus on improvement potential (exploitation) for optimal performance whereas a quite drastic change in the environment's fitness made the focus on regularity (exploration) become superior. But the transition depends on the employed kernel function. For example the Wendland kernel (with small support) showed a rather linear correlation between the similarity of the fitness change and the optimal preference weight whereas for the global triharmonic kernel a focus on regularity is superior even for very constant fitness environments. In the more complex 3D scenario the results are very noisy: For the Wendland kernels we can observe a weak trend for superior regularity focus in changing environments and superior focus on improvement potential for constant environments. But the range for a superior focus on improvement potential is much larger than in the 1D scenario. Thus, we only could give a very coarse suggestion of the choice of preferences to set up RBF center distributions. We favor regularity for a strongly changing fitness, improvement potential for a constant fitness, and an intermediate focus otherwise.

The reasons for the weak results are diverse: One aspect in 3D are the Pareto-fronts of low quality. Moreover, because the regularity measure is problem independent it results in a variety of uniform distributions with identical regularity score but varying improvement potential. Thus, the chosen distribution can randomly result in good fits—or bad ones. This induces a large portion of random noise. To tackle these issues we question and analyze our concept of evolvability in the next chapter.

8 Extended Analysis of Regularity and Variability

In our tests in Chapter 6 we realized that optimal center distributions with respect to regularity rarely reach the maximum score of 1. Thus, a design optimization would converge slowly despite the optimization of regularity. Hence, targeting optimal center distributions is not sufficient for a good convergence speed.

Hence, we approach this problem from a different, more fundamental angle and propose a representation-agnostic optimization, that is based on the deformation matrix only. RBF, FFD, and shell deformation (like all linear deformation techniques) can be written in matrix notation (Equation (2.6)), where a constant deformation matrix maps the basis functions' coefficients to per-vertex displacements of the mesh geometry. Interestingly, the expected optimization performance can be estimated from the deformation matrix alone. Thus, we formulate an optimization based on the deformation matrix itself: For any given deformation setup, a matrix orthogonalization leads to provably optimal regularity while preserving both the design/phenotype space and the improvement potential. This optimization is computationally efficient, fully automatic, and can be applied in a black-box-manner to any linear deformation representation. It can boost convergence speed by up to an order of magnitude without affecting the optimization's outcome, as we demonstrate through extensive numerical experiments in the first part of this chapter. This research is published in [RDMB18]. We conducted our experiments for FFD based on an implementation of S. Dresselhaus [Dre17], the second author of [RDMB18].

Because this orthogonalization-approach is basically a transformation of the employed basis functions that are linked to the control points, any distribution of control points can be modified for optimal regularity. This makes the direct optimization of control points dispensable. Moreover, because any distribution of control points can be modified for optimal regularity, the connection between the uniform distribution of RBF centers and optimal regularity breaks apart. Thus, our attempt to link regularity and exploration has to be revoked and only variability remains to characterize exploration now. But, as we have discussed in Chapter 5 variability is basically constant for any distribution with a fixed amount of control points. To resolve this dilemma we scrutinize the definition of variability and propose a different approach in the second part of this chapter. Based on the new definition of variability we again construct optimal compromise setups with respect to improvement potential and perform a further preference analysis to improve the results from the previous chapter.

8.1 Matrix Orthogonalization for Optimal Regularity

RBF kernel distributions are optimized in Chapter 6 to improve the regularity of the initial deformation setup, requiring a computationally expensive evolutionary optimization by itself. But despite the high computation cost, the improvement in regularity (and hence in convergence speed) is rather modest. In a similar spirit, [MOS06, SMB12] switched from indirect to direct manipulation of RBF and FFD to improve causality and thereby increase convergence speed. We showed in Chapter 6 that these representation-specific methods provide only limited gain in convergence speed. In contrast, our method provably converts any linear deformation representation to optimal regularity through a suitable orthogonalization of the deformation matrix – without changing RBF kernels, FFD control points, or shell handles, and therefore without the need for an additional optimization procedure. In contrast to Chapter 6 we fix the solution space now and optimize its basis.

Regularity is strongly linked to the concepts of locality/causality, which aim for representations where small changes in the genotype result in small changes in the phenotype. This preservation of local neighborhoods in the genotype–phenotype mapping allows for more efficient exploratory evolutionary search [Rot06, WCT12]. However, typically the mutation or crossover operators are addressed with these concepts, e.g., with locality in genetic programming [GLMOB11] or grammatical evolution [TR14], or with causality for genetic representations in antenna design [CRLL12]. In contrast, we incorporate locality/causality into the representation. Not only does the orthogonalized representation setup feature optimal regularity, it also perfectly realizes locality/causality and, as a consequence, results in faster convergence of evolutionary optimization processes.

Our orthogonalization can be considered as a particular preconditioning technique, which are used in numerical analysis to improve the convergence of iterative solvers, e.g., [LMDM14, Bai15, dPVvZvB17]. Our orthogonalization employs the singular value decomposition (SVD), which is used in [GRO$^+$03] to increase the performance of evolutionary optimization. But while the decomposition is applied to the mutation operator in [GRO$^+$03], we optimize the underlying deformation representation.

In [LM12] the SVD is employed in the design optimization context with FFD. The goal in this article is the construction of optimal FFD grids. The SVD is employed on the deformation matrix to determine a basis of the null space of the grid with respect to a certain point of the design. In this null space the grid points can move freely without causing a variation of this particular point

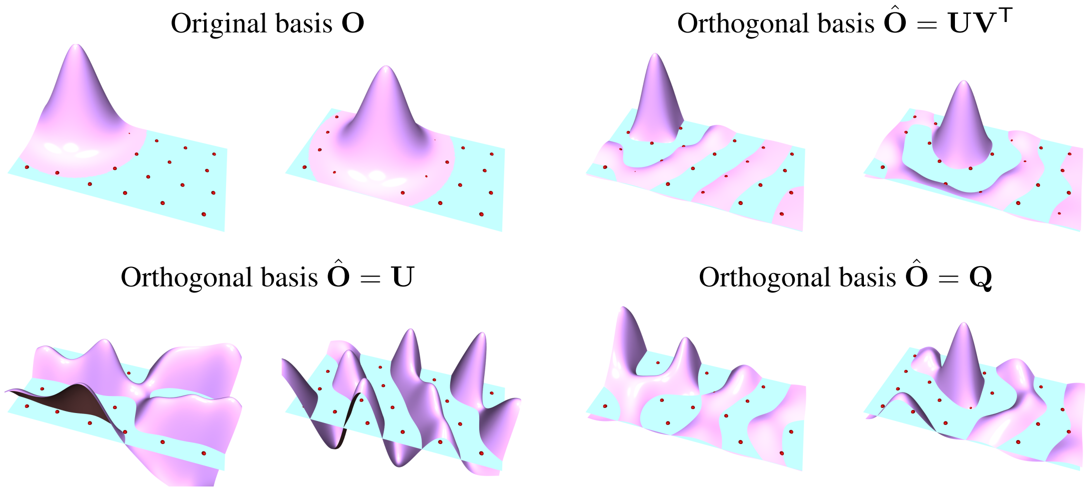

Original basis $\mathbf{O}$
Orthogonal basis $\hat{\mathbf{O}} = \mathbf{U}\mathbf{V}^\mathsf{T}$
Orthogonal basis $\hat{\mathbf{O}} = \mathbf{U}$
Orthogonal basis $\hat{\mathbf{O}} = \mathbf{Q}$

Figure 8.1: Visualization of two original basis functions and different orthogonal versions: Two columns of the deformation matrix $\mathbf{O}$ (top left). Their closest orthogonal approximation as $\mathbf{U}\mathbf{V}^\mathsf{T}$ of SVD (top right) are geometrically more similar to the original bases than using only the $\mathbf{U}$ matrix of SVD (bottom left) or the $\mathbf{Q}$ matrix of a QR decomposition (bottom right).

of interest. Thereby, an enhanced configuration of the control grid can be constructed. But, this approach again targets the optimal distribution of control points in contrast to an optimal deformation matrix, as we do now.

In general, the matrix entry $\mathbf{O}_{i,j}$ stores the jth basis function evaluated at vertex $\mathbf{x}_i$, and $\mathbf{p}_j$ is the coefficient or control parameter of that basis function. For example, for the RBF deformation, $\mathbf{O}_{i,j} = \varphi\left(\|\mathbf{x}_i - \mathbf{c}_j\|\right)$, with the kernel φ positioned at center $\mathbf{c}_j$. For FFD, which we additionally analyze here, $\mathbf{O}_{i,j} = N^3_{j_1}(u_i)N^3_{j_2}(v_i)N^3_{j_3}(w_i)$, a tri-cubic tensor-product B-spline function evaluated at (u_i, v_i, w_i), the local coordinates of $\mathbf{x}_i$ w.r.t. the FFD lattice [SP86, PT96] (implementation from [Dre17]). Finally, for linear thin shells there is no analytic expression. The jth column is the discrete response function to a virtual unit displacement of the jth control handle, computed by minimizing physical stretching and bending energies [BS08, BKP$^+$10].

In Chapter 6 we optimized RBF deformation setups with respect to regularity and improvement potential through optimization of kernel positions. However, even when optimizing solely for regularity, the resulting setups are still far from the optimal regularity value of one for larger kernel widths. This can be explained by analyzing the regularity definition: An optimal regularity requires *all* singular values σ_i to be one for optimal condition number $\kappa(\mathbf{O}) = \sigma_1/\sigma_n$ ($\sigma_1 > \cdots > \sigma_n$), which is true for orthogonal matrices only. For our non-square $m \times n$ matrix $\mathbf{O}$ with $m > n$, this requires the matrix columns $\mathbf{o}_1, \ldots, \mathbf{o}_n$ to be of unit length and mutually perpendicular, which (with slight misuse of notation) we call orthogonal, too. Because the RBF basis functions are not orthogonal w.r.t. the L_2 inner product, their discretization will in general not lead to orthogonal columns $\mathbf{o}_j$. The same is true for the B-spline basis of FFD and shells.

We approach the problem from a different angle, by directly optimizing the matrix $\mathbf{O}$, instead of indirectly manipulating it through careful placement of RBF kernels or FFD control points.

The columns $\mathbf{o}_j$ form a basis of the n-dimensional phenotype sub-space of $\mathbb{R}^m$, since any displacement $\mathbf{OP}$ can be written as a linear combination $\sum_{j=1}^{n} \mathbf{o}_j \mathbf{p}_j^{\mathsf{T}}$. Asking these basis vectors to be orthogonal and of unit length is in agreement with the concepts of *locality* [Rot06] and *causality* [SKVS97, WCT12], which are closely related to convergence speed. They emphasize that similar parameter variations should yield similar amounts of phenotype variation and that local neighborhoods should be preserved, both of which is achieved by orthogonal matrices. In the ideal case of an orthogonal deformation matrix, regularity, locality, and causality nicely coincide.

In practice, however, the deformation matrix $\mathbf{O}$ is not orthogonal, and we propose to orthogonalize it, which corresponds to a change of basis for the phenotype space. This can be achieved by several techniques, such as Gram-Schmidt orthogonalization, QR decomposition, or singular value decomposition (SVD) [GVL12]. Although computationally most expensive, we employ the SVD, since it is numerically most stable and yields the orthogonal matrix closest to the original one [Zha00], i.e., it changes the deformation basis the least, as visualized in Figure 8.1.

We decompose $\mathbf{O}$ using the thin SVD [GVL12] into $\mathbf{O} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^{\mathsf{T}}$, with orthogonal matrices $\mathbf{U} \in \mathbb{R}^{m \times n}$ and $\mathbf{V} \in \mathbb{R}^{n \times n}$ and a diagonal matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{n \times n}$ containing the singular values $\boldsymbol{\Sigma}_{i,i} = \sigma_i$. Removing the singular values $\boldsymbol{\Sigma}$ (or setting all σ_i to one) yields the orthogonalized deformation matrix (as the product of two orthogonal matrices):

$$\hat{\mathbf{O}} = \mathbf{U}\mathbf{V}^{\mathsf{T}}. \tag{8.1}$$

By construction $\kappa(\hat{\mathbf{O}}) = 1$, therefore $\hat{\mathbf{O}}$ has optimal regularity, locality, and causality, and we can expect faster convergence. Since the columns of $\hat{\mathbf{O}}$ span the same phenotype space as the columns of $\mathbf{O}$, the improvement potential is unchanged and the design optimization can reach the same optimum in both variants. In practice, starting from a given representation $\mathbf{O}$, we first compute its orthogonal version $\hat{\mathbf{O}}$ and transform the initial original parameters $\mathbf{P}$ to $\hat{\mathbf{P}}$ with

$$\hat{\mathbf{P}} = \mathbf{V}\boldsymbol{\Sigma}\mathbf{V}^{\mathsf{T}}\mathbf{P}. \tag{8.2}$$

Then we perform the evolutionary optimization (more efficiently) based on $\hat{\mathbf{O}}$. The resulting optimal parameter vector $\hat{\mathbf{P}}$ is finally mapped back to the original representation $\mathbf{O}$ as $\mathbf{P} = \mathbf{V}\boldsymbol{\Sigma}^{-1}\mathbf{V}^{\mathsf{T}}\hat{\mathbf{P}}$. This allows us to perform the optimization using the more efficient representation, but to convert the optimized parameters (exactly) back to the original representation, where they have their originally intended semantic meaning.

In order to analyze the orthogonalization's effect on the convergence speed in actual evolutionary optimization, we compare optimization runs *with* and *without* orthogonalization of the deformation representation for the experiments described in Chapter 5. For the two test scenarios (1D function approximation, 3D template fitting) and the different types of representations (RBF, FFD, shell) we run an evolutionary optimization for 100 random setups each. The RBF tests are further split up according to the employed kernel function and its support radius: W_2, W_5, W_{15} refer to Wendland kernels of support radius 2, 5, and 15; *Tri* refers to the triharmonic kernel; *dm* and *im* denotes direct or indirect deformation. Like before, we utilize a (1,10)-CMA-ES

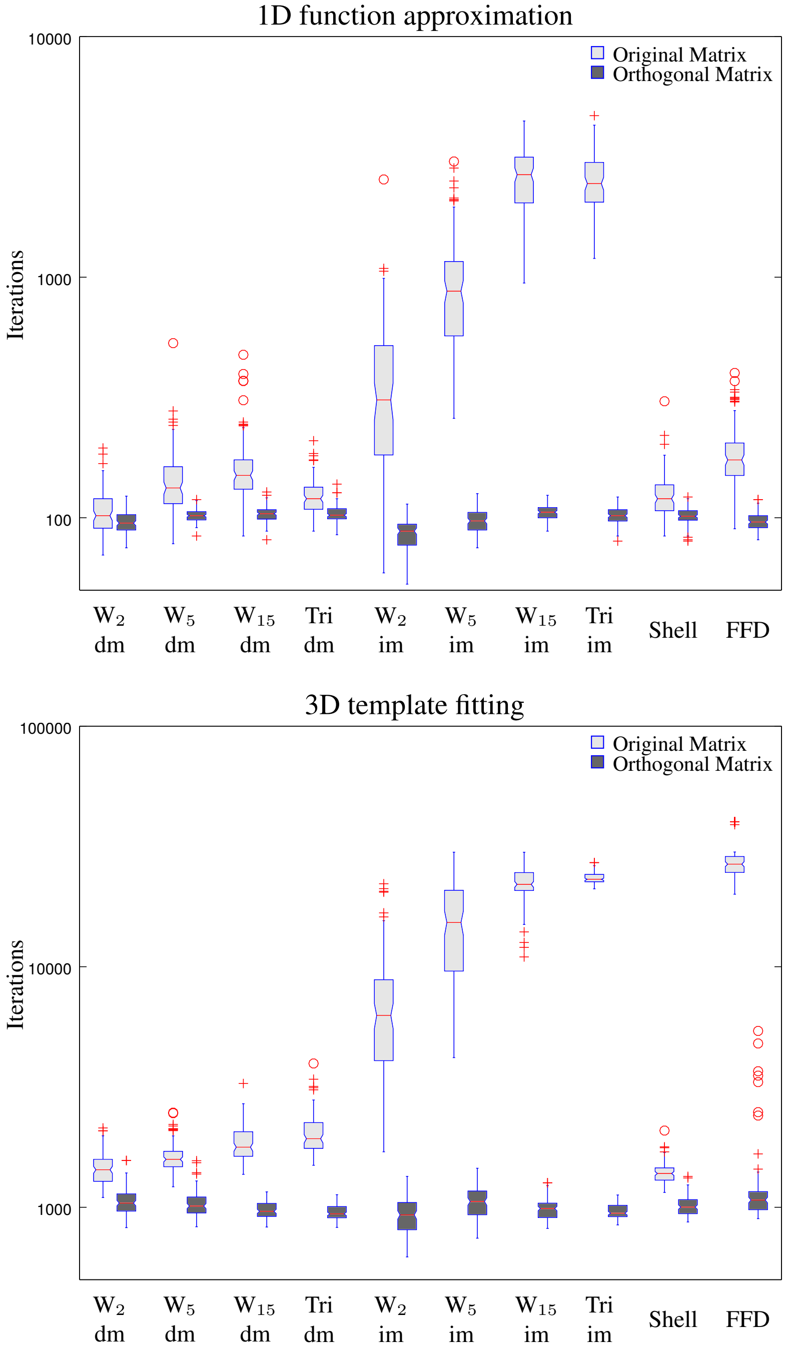


Figure 8.2: Pairwise comparison of the convergence speed of CMA-ES (#iterations until convergence) for different deformation representations (column pairs), for original representation $\mathbf{O}$ (light color) versus orthogonalized representation $\hat{\mathbf{O}}$ (dark color), for fitting 1D height fields (left) and 3D face scan (right), averaged over 100 random setups each.

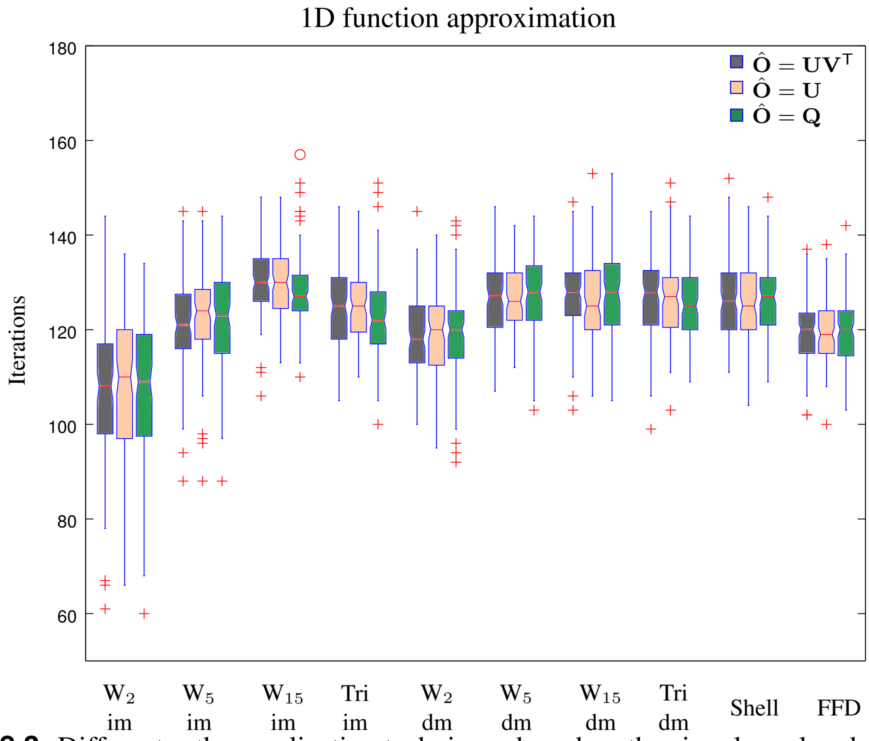


Figure 8.3: Different orthogonalization techniques based on the singular value decomposition $\mathbf{O} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\mathsf{T}$ or the QR decomposition $\mathbf{O} = \mathbf{Q}\mathbf{R}$ yield equivalent convergence speed.

of the shark library [IHMG08] with manually determined optimal initial step sizes s for each representation. For the 1D height field we chose $s = 0.001$ for the unmodified and $s = 0.01$ for the orthogonal setting. For fitting 3D faces we chose $s = 0.001$ and $s = 0.05$, respectively. Otherwise the default settings of the shark library are applied. The results in Figure 8.2 show that on average the orthogonalized setups converge faster than the unmodified ones, by more than an order of magnitude for representations with low initial regularity, such as FFD and im-RBF. These numerical experiments demonstrate that our orthogonalization approach – which raises high expectations for faster convergence due to the its optimal regularity – indeed meets these expectations.

Although we recommend the SVD-based orthogonalization $\hat{\mathbf{O}} = \mathbf{U}\mathbf{V}^\mathsf{T}$ from Equation (8.1) due to its numerical stability and the fact that it minimally modifies the input setup, Figure 8.3 shows that alternative orthogonalizations, such as using the $\mathbf{U}$-matrix of SVD or the $\mathbf{Q}$-matrix of QR decomposition, yield equivalent results in terms of convergence speed. This emphasizes the importance of the general concept of orthogonal representations, which is in agreement with the concepts regularity, locality [Rot06], and causality [SKVS97, WCT12].

Our approach offers an interesting view onto the concept of *indirect* versus *direct* manipulation. Indirect manipulation deforms a geometry by changing the coefficients of the basis function (e.g., RBF coefficients or spline control points), while direct manipulation prescribes displacements of some handle points on the surface and solves a linear system to determine the

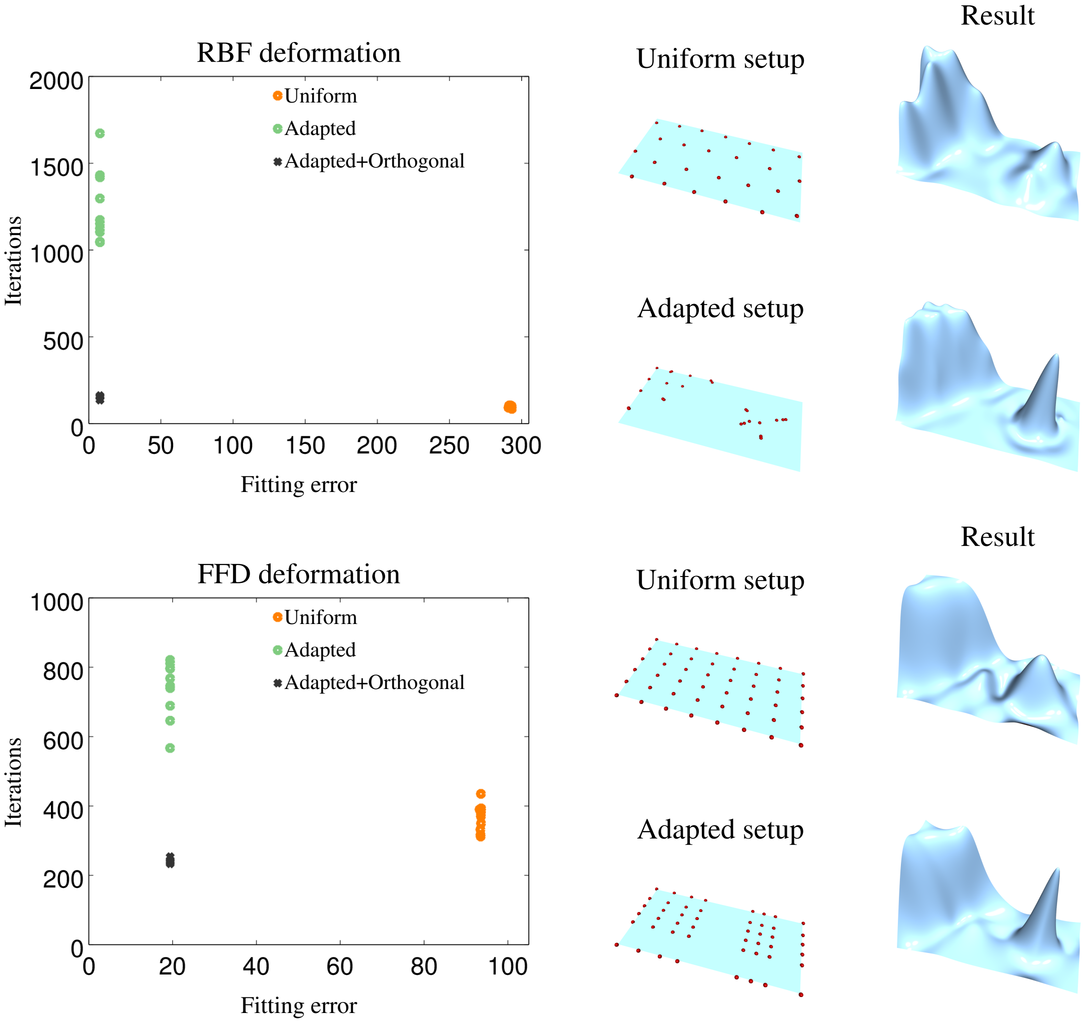


Figure 8.4: Matrix orthogonalization for custom-tailored setups. We analyze height field fitting, using RBF deformation (top) and FFD (bottom). The uniformly distributed setups (having higher regularity) converge faster than the target-adapted ones, but the adaptive setups (having higher improvement potential) achieve a better fitting quality with lower error. Orthogonalizing the target-adapted setup combines both advantages: low error *and* fast convergence.

coefficients that yield this desired deformation. Direct manipulation has better regularity and converges faster in a design optimization (Chapter 5 Figure 5.4, 5.8), as also confirmed by the experiments in this chapter (Figure 8.2). The switch from indirect to direct manipulation can be considered as a matrix preconditioning that improves regularity to a certain extend. In this view, our orthogonalization provides a superior alternative that projects the matrix to *optimal* regularity, thereby improving convergence speed even more (Figure 8.2).

While the previous experiments analyzed our orthogonalization technique for *random setups* only, we now demonstrate its practical relevance by applying it to *custom-tailored setups*. Fig-

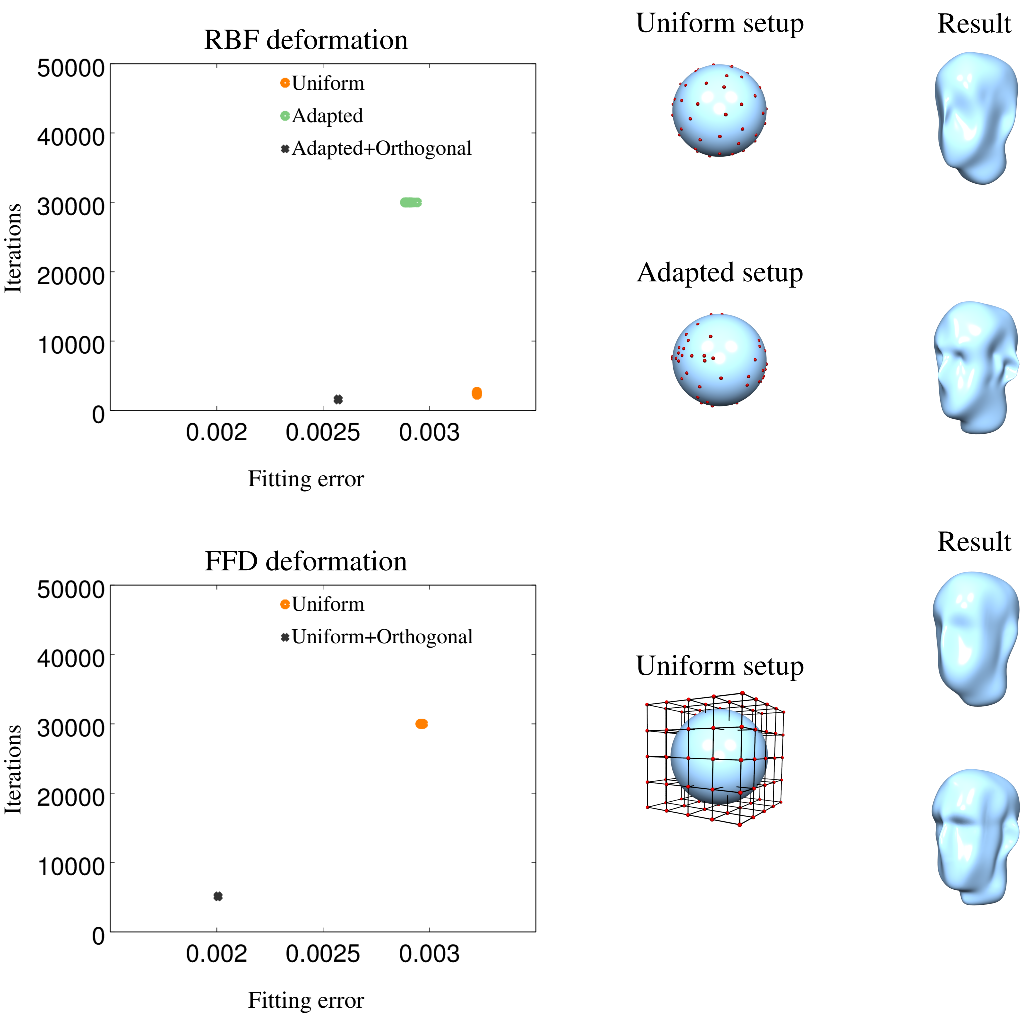


Figure 8.5: The results of the analysis of setups for 3D faces, using RBF deformation (top) and FFD (bottom), is equivalent to the 1D scenario. The uniformly distributed setups (having higher regularity) converge faster than the target-adapted ones, but the adaptive setups (having higher improvement potential) achieve a better fitting quality with lower error. Orthogonalizing the target-adapted setup combines both advantages: low error *and* fast convergence.

ure 8.4, 8.5 shows the results for fitting of 1D height fields and 3D face scans, of 10 trials using both RBF and FFD representations. The utilized uniform deformation setups have a higher regularity and therefore converge faster, but target-adapted setups have a higher improvement potential and achieve better fitting results. When setting up deformation representations by specifying RBF kernels or FFD control grids, one always has to find a compromise between these two extremes, as analyzed in detail in Chapter 6. In stark contrast, our matrix-based setup optimization does not face this problem, as it projects *any* input setup to optimal regularity without changing its phenotype space or improvement potential. Applying the setup orthogonalization to the target-adapted setups consequently preserves its ability to generate high-quality fitting results,

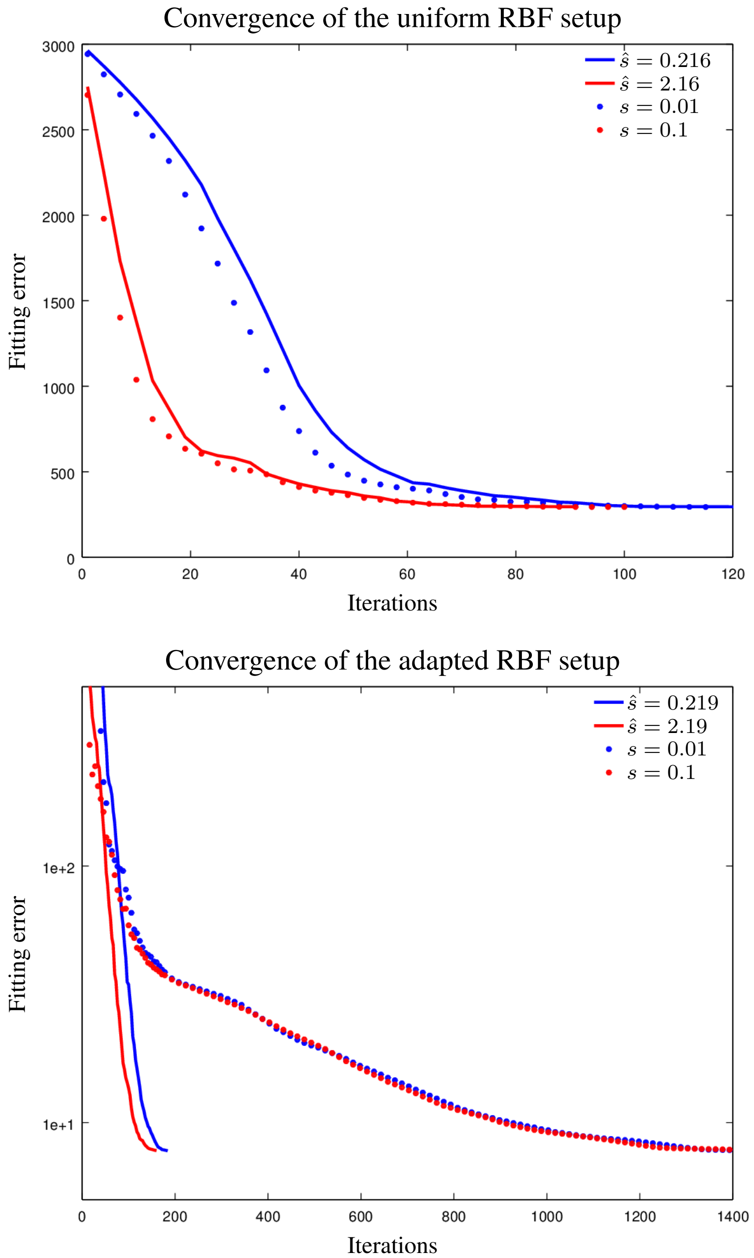


Figure 8.6: Convergence plots for the height field fitting scenario with a uniform RBF setup (left) and the adapted RBF setup (right). The initial step sizes s for the original representations $\mathbf{O}$ have been converted to $\hat{s}$ for the orthogonalized representations $\hat{\mathbf{O}}$ through Equation (8.3). For two initial step sizes s (red and blue), the convergence behavior for $\hat{\mathbf{O}}$ (solid line) matches that of the original setup $\mathbf{O}$ (dotted) in the beginning of the optimization. While the fast-converging uniform setup show an overall similar behavior (left), for the adaptive setups with low initial regularity our optimized setup converges faster (right).

but considerably reduces the number of iterations required to do so. The impact of orthogonal decompositions is stronger for the more complex face fitting scenario. The adaptive RBF setups did not fully converge due to their low regularity, and hence yield an error that is only slightly lower than the uniform setups. The orthogonalized adapted setups converge without problems and show the lowest errors. In the FFD example, we did not succeed in constructing an adaptive lattice of control points to produce better results than the uniform lattice. Therefore we only analyze the uniform lattice and its orthogonalized version, where the latter converges considerably faster to a considerably better solution.

For the previous experiments we hand-tuned the initial step sizes for the CMA-ES in order to focus the analysis on the actual representations. While for long-running optimizations the initial step size s has a minor effect, its choice is more important for fast-converging optimizations. Given an initial step size s for the original representation $\mathbf{O}$ (e.g., from the designer's knowledge), it can be converted to the orthogonal representation $\hat{\mathbf{O}}$ by compensating for the normalization of the matrix columns $\mathbf{o}_j$:

$$\hat{s} = s \cdot \frac{1}{n} \sum_{j=1}^{n} \|\mathbf{o}_j\| \ , \tag{8.3}$$

where $\|\mathbf{o}_j\|$ is the length of the jth column of $\mathbf{O}$ (see Figure 8.6).

Being able to modify any distribution of control points for optimal regularity breaks up its link to exploration. Following our interpretation of evolvability in Chapter 3 variability is the remaining criterion to characterize exploratory capabilities of the setup, now. But according to our definition of variability it characterizes the degrees of freedom (Chapter 4) and is independent of the distribution of control points (Chapter 5). Thus, we approach this dilemma with an alternative definition of variability in the next section. Moreover, we analyze the new definition for dynamic scenarios.

8.2 Redefinition of Variability

In Chapter 3 and Chapter 4 we motivated and defined variability as a measure for exploratory capabilities induced by a representation. An exact approximation of any phenotype with the representation would have the best exploration potential and thereby maximal variability. This theoretical fact is limited by the employed number of parameters because they are only few compared to the dimension of the phenotype. We motivated the definition of variability by computing the expected approximation error of any possible phenotype. If any phenotype can be approximated well, then variability should be good. This holds for the number of control points: increasing their amount increases variability. But, the variability definition (Equation (4.4)) can not distinguish between different distributions of control points with the same number.

Computing the expected approximation error has one significant drawback. Because the representation has to approximate any possible phenotype there are phenotypes which can be exactly approximated, and others which result in a large approximation error (Figure 8.7). Computing

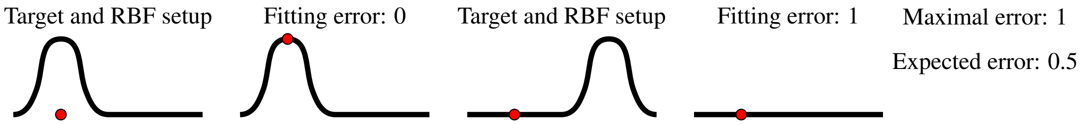


Figure 8.7: Exemplary visualization of the approximation quality of a particular RBF center "distribution". The target on the left can be exactly approximated in contrast to the target on the right. Computing the expected fitting error cancels the impact of bad approximation results.

the expected error cancels these extremes. This holds for each representation setup (as long as the parameters effect the phenotype, e.g., a deformation matrix of full rank) leading to identical variability.

Instead of the expected approximation error we now compute the squared maximum error e_{max} for a displacement $\bar{\mathbf{o}} = (\bar{o}_1, \ldots, \bar{o}_m)$ ($\bar{o}_i \in [-1, 1]$ for mathematical convenience) of a design with the representation $\mathbf{O}$ (compare to Equation (4.1)):

$$e_{max} = \max_{\bar{o}_i \in [-1,1]} \left\| \bar{\mathbf{o}} - \mathbf{O}\mathbf{O}^{+}\bar{\mathbf{o}} \right\|^2 \ . \tag{8.4}$$

Although we don't have a general solution for this equation, which might not exist for arbitrary linear deformations, we can bound e_{max} with the so-called *fill distance* for standard indirect RBF deformations following [Wen04]. This quantity measures the maximum distance of any vertex of a design to its nearest center. Given the vertices of a discrete design $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$ and a RBF center distribution $\mathcal{C} = \{\mathbf{c}_1, \ldots, \mathbf{c}_n\}$ the fill distance $h_{\mathcal{X},\mathcal{C}}$ is defined as:

$$h_{\mathcal{X},\mathcal{C}} = \sup_{\mathbf{x} \in \mathcal{X}} \min_{\mathbf{c} \in \mathcal{C}} \left\| \mathbf{x} - \mathbf{c} \right\| \ .$$

To compute an upper bound for e_{max} we switch from our discrete setting to the point-wise continuous one to apply the results of [Wen04]. In our notation the best interpolant $\bar{o}_j$ of a variation at vertex j is the product of the j-th row of $\mathbf{O}\mathbf{O}^{+}$ and $\bar{\mathbf{o}}$ which is denoted with $s(\bar{o}_j)$ in [Wen04][1] . With this notation we bound e_{max} via:

$$\begin{aligned} e_{max} &= \max_{\bar{o}_i \in [-1,1]} \left\| \bar{\mathbf{o}} - \mathbf{O}\mathbf{O}^{+}\bar{\mathbf{o}} \right\|^2 \\ &= \max_{\bar{o}_i \in [-1,1]} \sum_{j=1}^{m} \left| \bar{o}_j - (\mathbf{O}\mathbf{O}^{+})_j \bar{\mathbf{o}} \right|^2 \\ &\leq \sum_{j=1}^{m} \max_{\bar{o}_i \in [-1,1]} \left| \bar{o}_j - (\mathbf{O}\mathbf{O}^{+})_j \bar{\mathbf{o}} \right|^2 \\ &= \sum_{j=1}^{m} \max_{\bar{o}_i \in [-1,1]} \left| \bar{o}_j - s(\bar{o}_j) \right|^2 \\ &\leq \sum_{j=1}^{m} \bar{c} \cdot F(h_{\mathcal{X},\mathcal{C}}) \ = \ c \cdot F(h_{\mathcal{X},\mathcal{C}}) \end{aligned}$$

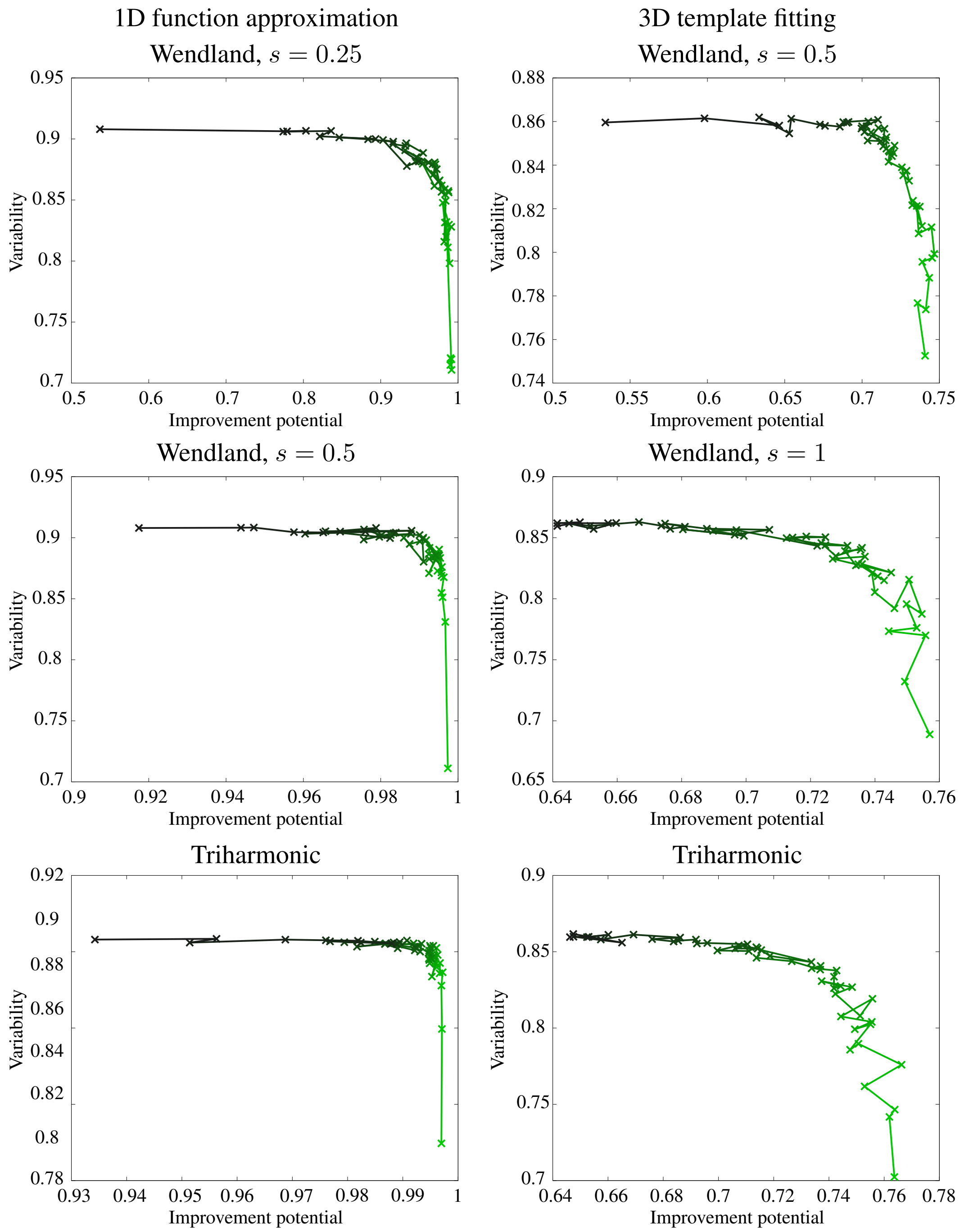


Figure 8.8: The solutions of a weighted single-objective optimization for 51 preferences weights trading of between variability and improvement potential. The results for the kernels of the 1D scenario are left; those of the 3D scenario are right. The preference weight is color-coded from a focus on variability (dark green) to improvement potential (light green).

with $c = \bar{c} \cdot m$, $\bar{c}$ being a constant number, and $F(h_{\mathcal{X},\mathcal{C}})$ being a monotonically increasing function in the fill distance. Thus, reducing the fill distance reduces the maximal approximation error e_{max} up to a theoretical limit of 0, which is achieved if every vertex of the design is equipped with a center. The upper limit of e_{max} is obtained if only one center is placed at a vertex of a design, which has the largest distance to the other vertices. We call this largest possible distance between any pairs of vertices the diameter $\varnothing\mathbf{X}$ of the design $\mathbf{X}$ and define the "new" variability based on the fill distance as:

$$V(\mathbf{O}(\mathcal{X},\mathcal{C})) = \frac{\varnothing\mathbf{X} - h_{\mathcal{X},\mathcal{C}}}{\varnothing\mathbf{X}} \; . \tag{8.5}$$

Like the definitions of regularity and improvement potential this new definition is scaled between 0 and 1 with 1 being the optimal value. Applying Equation (8.5) restricts the choice of the kernel functions. The proposed derivation only holds for symmetric basis functions, which are commonly employed anyway, like the radial Wendland and triharmonic kernels. It cannot be applied to differently mixed kernels or linearly combined basis functions, as in the case of direct manipulation.

For the realization of measures for exploration one has to decide to follow a genotype-based or phenotype-based approach as motivated in Chapter 3. Although, we emphasize and apply a phenotype-based approach because the target is to explore the phenotype, when having a second glance at Equation (8.5), we realize a definition based on both spaces. The maximum error in the phenotype can be bounded by a maximum distance—the fill distance—in the genotype, whereby both approaches coincide nicely.

For an efficient optimization of the new variability we apply the Lloyd sampling (discussed in Chapter 6) because this sampling results in a uniform center distribution, which naturally has a good fill distance and thereby a good variability score.

This link between Lloyd sampling and fill distance motivates us to discuss regularity again. Previously, in Chapter 7, we linked optimal regularity to a uniform (Lloyd) distribution and this to exploration. Now, we link variability to the Lloyd sampling. However, the connection between Lloyd sampling and exploration still remains such that we can simply exchange the place of regularity with variability for dynamic scenarios. Thus, we expect similar results for the preference analysis from the previous chapter.

To conduct a preference analysis we first need the optimal center distributions for different compromises between the alternative variability definition (Equation (8.5)) and improvement potential (Equation (4.6)). For simplicity we apply the CMA-ES with heuristic initialization for a weighted single-objective optimization as described in Chapter 6 to optimize 25 centers in the 1D scenario and 75 centers in 3D. We still employ soft constraints (Equation (6.4)) for the initial plane in the 1D tests because it enables a more efficient evolutionary search. Removing an individual directly, which violates a constraint only in one center, results in many discarded individuals if one center is on the edge of the valid area. If several centers of a parent population are at this edge the evolutionary search might get stuck. In contrast, a soft barrier takes these

[1] Pages 11, 181, 188

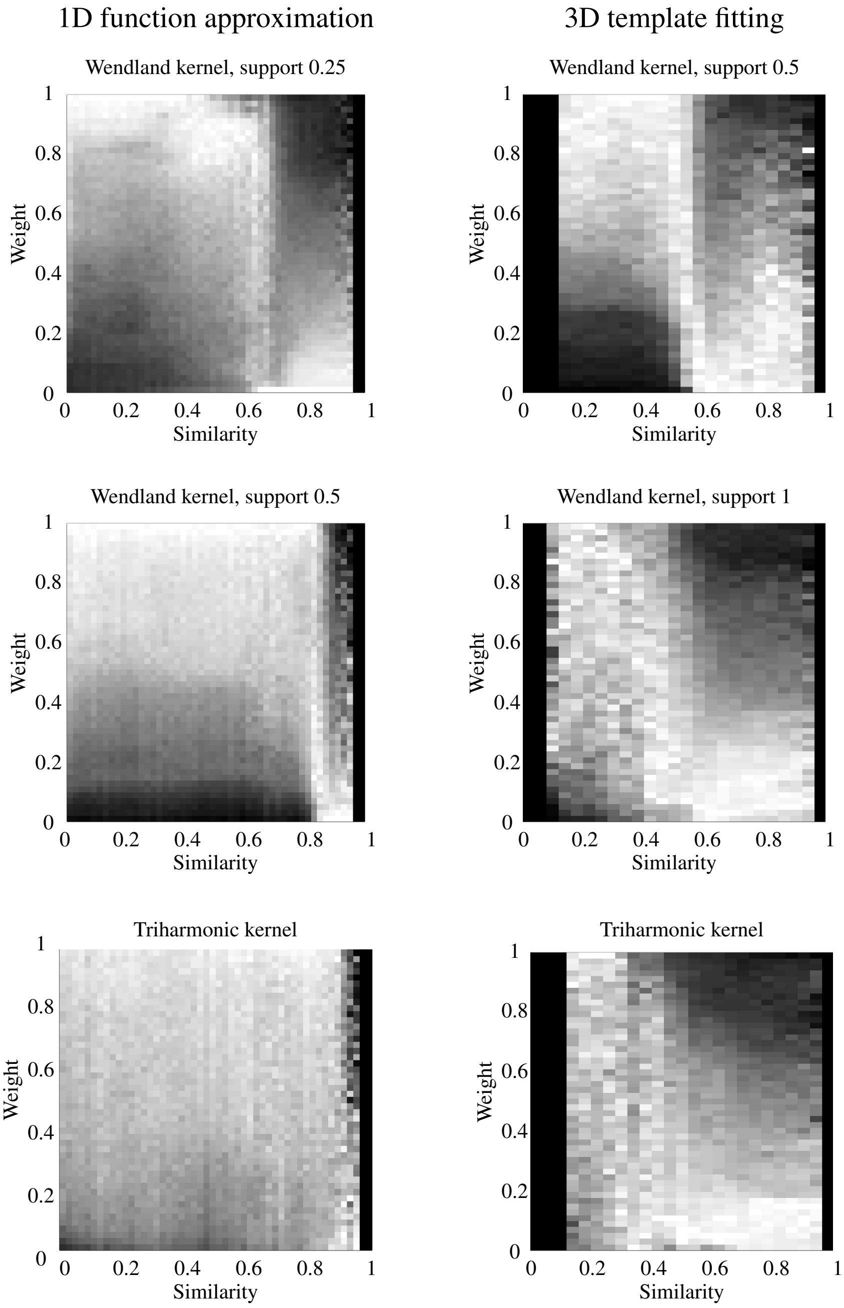


Figure 8.9: Optimal preference weights (highlighted in white) for 1D function approximation and 3D template fitting. For drastic changes in the fitness function (small similarity score) a focus on regularity (weight of 1) is superior. In contrast, for a small changes in the fitness function (high similarity score) a focus on improvement potential (weight of 0) is superior. But the range where and how to switch between the compromises remains rather imprecise.

slightly invalid individuals into account and thereby supports the search by pushing new offspring towards the valid area through the covariance updates. After the optimization the solution is projected back to the valid area as we did for the center optimization in Chapter 6. In Figure 8.8 the newly computed solutions for 51 different preference weights ($\lambda = 0, 0.02, \ldots, 1$, color-coded) are plotted. Especially, the solutions for triharmonic kernels are better distributed than the optimization for regularity (compare to Figure 6.11).

With these compromise setups we again perform the preference analysis for dynamic scenarios like in Chapter 7. Given a similarity of gradient information derived from various targets to be fitted, our goal is to determine which preference is the optimal choice for the different targets and different kernels. As before we rank the preferences according to their fitness score for each 10000 (1000) randomized targets in the 1D (3D) scenario and average the ranking for the similarities. The results in case of the triharmonic kernel in the 3D scenarios are less noisy, now. But overall, a similar trend as in the previous preference analysis can be observed (Figure 8.9). For a strongly changing dynamic, indicated by a low similarity score with respect to the previous fitness environment, a focus on exploration/variability, i.e., a uniform center distribution, is superior. In contrast, in a static environment a focus on exploitation/improvement potential, i.e., a fitness-adapted center distribution, is superior. Still, the range for an optimal intermediate focus, e.g., a preference weight of 0.5, is rather small and problem dependent. However, such an intermediate weight works fine for any dynamic change in our tests.

As mentioned before, there are various problems, which deteriorate precise results. (1) The Pareto-fronts as shown in Figure 8.8 are still not optimal. (2) Center distributions with optimal variability can accidentally have high improvement potential and vice versa. (3) The analyzed ranking does not take the absolute fitness score of the results into account. (4) The different kernels vary in their sensitivity with respect to their location. E.g., the fitting quality with a triharmonic kernel is almost independent of the kernel distribution whereas a local Wendland kernel is very sensitive.

8.3 Summary

For any design optimization, convergence speed and solution quality are crucial aspects to guarantee short developmental cycles of industrial products. This is especially true for evolutionary design optimizations with computationally expensive fitness functions. The choice of optimization parameters, i.e., the chosen representation, drastically influences the results. Optimally setting up linear deformation representations, which play a major role in practical design optimization of complex geometries, is therefore a (very challenging) problem of high practical relevance.

We argued that setting up a deformation representation by a careful placement of kernels or control points can lead to high-quality results for target-adapted setups, but is inherently limited for optimizing convergence speed, since the deformation basis functions do not yield an orthogonal deformation matrix. The resulting setups violate the design principles of regularity, locality, and causality, and will in general not provide high convergence speed.

Our automatic setup optimization is inspired by the concept of regularity. The proposed SVD-based orthogonalization can be applied to any linear deformation representation, is easy to implement, efficient to compute, and is optimal with respect to regularity, causality, and locality. The optimized setups showed performance improvements by up to an order of magnitude.

The applied SVD (or any orthogonalization) can be seen as a simple basis transformation of the phenotype space. However, the resulting basis vectors are unintuitive and to complicated to be construction manually.

Since the orthogonalized matrix spans the same phenotype space as the original matrix, our setup optimization does not negatively affect the optimization's results. For the user it acts as a perfect black-box: The designer provides an input representation, which is automatically orthogonalized; the optimization is efficiently performed using the orthogonal representation; and the final result is converted back to the original representation. Our approach has the potential to become a general recommendation for adapting any matrix in the context of linear deformation representations.

But as any deformation matrix can be set to optimal regularity through orthogonalization, the previously drawn link to exploration breaks apart. Thus, we fell back to an alternative approach to measure variability, that we linked to exploration instead. In case of RBFs we define variability with the fill distance. With this alternative definition of variability uniform center distributions achieved the best score (like in test before in Chapter 6) which perfectly fits the motivation behind exploration. Thus, we simply replaced the regularity criterion with variability from a conceptual point of view.

We again constructed compromise setups (RBF center distributions) with respect to variability and improvement potential targeting an improved preference analysis between exploration and exploitation. These compromises are better distributed, now. But because key problems of the preference analysis still remain, its results are only improved by a small margin. Thus we only can emphasize to employ a focus on exploration/variability for heavily changing dynamics and a focus on exploitation/improvement potential for an almost static environment. However, an intermediate preference between the criteria works fine in any scenario, which would be our recommendation if the variation in the dynamic is either unclear or intermediate.

9 Conclusion

Designers and engineers face a wide variety of challenges to come up with novel solutions for industrial design problems. The performance of an evolutionary optimization cycle, which is typically applied in these scenarios, depends significantly on the quality of the employed representation. *Linear deformation* methods, like Free-Form deformation (FFD) or deformations with radial basis functions (RBFs), as representations map the abstract genotype, i.e., the parameters, to meaningful phenotypes, i.e., design variations. The recurring tasks for designers are to choose the optimal distribution of control points, e.g., a FFD grid or the distributions of RBF centers, and to choose proper deformation functions, e.g., spline bases for FFD or radial symmetric kernels for RBF. This information is encoded into a deformation matrix. Ideally, the distribution of control points and the choice of the basis function, and thereby the deformation matrix, adapts to changing conditions to ensure a high-performing optimization process. Because of the impact of the representation/deformation matrix on the performance of the optimization we target optimal deformation matrices in our research.

Quality criteria are required to determine if a deformation matrix, or a deformation setup (as we call it), is optimal. Because industrial design optimization problems are complex, due to timely varying customer demands, manufacturing conditions, or even performance criteria, we analyzed the concept of complex system engineering. This concept motivates evolutionary optimization and the application of the meta-attribute *evolvability* to induce robustness and flexibility to the representation/deformation matrix during the developmental cycle.

Motivated by the meta-attribute evolvability we defined three novel quality criteria for linear deformations which are: *variability*, *regularity*, and *improvement potential*. We analyzed these criteria in two test scenarios. In a basic 1D scenario the design goal is to fit a plane to a target height field and in a more complex 3D scenario the objective is the optimal fit of a sphere to a scanned face.

Variability is independent of any fitness environment and thereby covers the exploratory capabilities of the deformation setup. According to this motivation highly variable setups should be suitable for environments with only limited information or no information at all. Based on two modeling approaches we defined variability basically as (1) the numbers of employed parameters (for any linear deformation) or as (2) the so-called fill distance (for RBF only). Our analysis reveals strong correlations between the number of parameters and the quality of an optimization process in the test scenarios. But, this approach (1) is independent of the distributions of control points as long as they are connected with the design. Thereby, it cannot be linked to the concept of exploration. This motivated the fill-distance as the alternative approach (2), which we finally link to exploration.

We motivated regularity as a general fitness independent criterion, like variability, to characterize the expected convergence speed of an evolutionary optimization algorithm. We define regularity basically as the condition number of the deformation matrix. Our analysis reveals strong correlations between the criterion and convergence speed of an evolutionary algorithm. Moreover, we discovered that an optimal regularity score or condition number of the deformation matrix, respectively, relies only on numerical properties of the deformation matrix and is independent of the distribution of control points. Through matrix orthogonalization with a singular value decomposition we tuned any deformation matrix for optimal regularity and showed improved convergence speed. Because this approach is independent of the center distribution it cannot be linked to exploration.

Of course a representation has to enable results of high quality based on available information about the fitness environment. Thus, we defined the third attribute improvement potential, which measures the potential of the representation to approximate (estimated) gradient information. Our analysis proofed strong correlation between the criterion and the fitting quality for the test cases even for imprecise gradient information. Improvement potential clearly corresponds to the concept of exploitation.

In essence, the defined criteria—variability, regularity, and improvement potential—are easy to apply to any linear deformation. Their local definitions do not rely on global information of the whole phenotype/design space, which makes them applicable for complex problems.

The comparison of FFD and RBF, to support a designer's choice which representation to employ, reveals no favorite with respect to the three criteria. As we have shown the fitting quality depends on the employed basis function, which in our tests showed no clear winner. And regularity can be tuned independent of the chosen type of linear deformation. However, orthogonalization for optimal regularity makes so-called direct manipulation approaches of FFD or RBF superfluous for algorithmic optimization. Even the comparison between local and global basis functions showed no clear winner in our tests. Although global basis functions (or functions with a large support) resulted in better fits for random distributions of control points, optimized distributions lead to equivalent fitting quality.

However, we recommend RBF deformations because of their simplicity and efficiency. An inherent drawback of FFD is the application of an additional optimization procedure to determine the local parametrization. Of course, this is much more inefficient than evaluating a simple function for RBFs. Moreover, the feasibility of a FFD control grid has to be guaranteed during its optimization. This limits the flexibility of the setup and results in additional constraints which would slow down the optimization of the deformation matrix.

Employing our proposed quality criteria we optimized RBF setups, more precisely their center distribution. Based on a multi-objective approach (running for 2 days) to compute Pareto-optimal distributions as compromises between exploration and exploitation we showed the capability of a more efficient weighted single-objective optimization to hit the Pareto front. To further enhance performance we drew inspiration of optimal center distributions of both ends of the Pareto front and discussed the Lloyd sampling, the orthogonal least squares (OLS) sampling, as well as an own approach for their combination. Although, these heuristics distribute centers almost immediately (1 minute), compared to a 2 hours running single-objective optimization, they lack the quality of a true optimization. However, employing the combination of Lloyd and OLS sampling as initialization for an optimization procedure improves its performance at almost no cost. Our attempt to further improve the quality of the setups and their computation time with a deterministic gradient-based optimization led to mediocre results. On the one hand the deterministic results are better for further analyses because they can be reproduced and the compromise solutions are partly more precise. On the other hand the gradient optimizer can get stuck in local optima easier than an evolutionary approach. However, we employed the deterministic approach for a detailed preference analysis.

Our goal was to choose the optimal compromise between exploration and exploitation dependent on the strength of the variation of the fitness function for dynamic design optimization scenarios. Our first tests in a static environments showed that an intermediate focus between exploration and exploitation is superior for imprecise gradient information. In a second series of tests we simulated a changing dynamic through a changing target to be fitted and thereby a varying fitness gradient. For very similar fitness gradients, inducing an almost static environment, a focus on exploitation leads to superior results, whereas for very distinct gradients, referring to a drastic change of the fitness, a focus on exploration is superior. But, intermediate preference weights were only better in a small range of similarity values and this range is highly problem dependent. However, an intermediate preference weight works fine for any environment. We can't give a more precise statement when to apply which preference weight according to our results. We figured out two key problems. First, a deformation setup tuned for exploration can accidentally exploit given information perfectly and vise versa. Second, the fitting quality of different setups cannot be compared because the fitness function and thereby the optimal value varies in the tests. This only allowed a simple ranking of the preferences, which does not give qualitative information to compare different fitness functions.

This leads us to general recommendations for future design scenarios. As mentioned before, we recommend RBF deformations for their simplicity. Given a kernel function we first optimize the center distribution with an evolutionary optimizer because of its robustness. If a designer expects a strong change in a dynamic environment a focus on exploration (variability) is promising. If a static dynamic is expected focusing exploitation (improvement potential) works best. Otherwise, or for imprecise information we recommend a 50–50 compromise. Moreover, for the exploitation cases we recommend local kernels, e.g., Wendland kernels with a small support radius

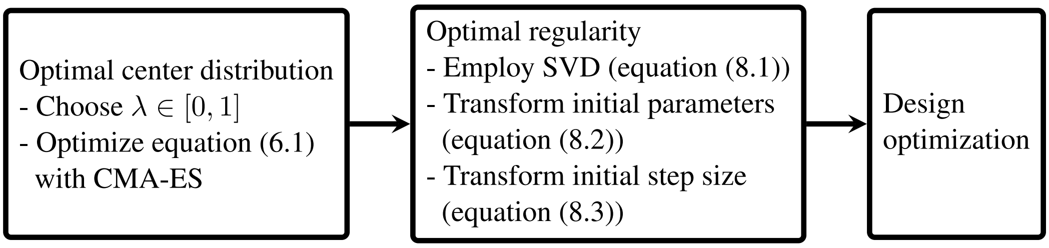

Optimal center distribution
- Choose $\lambda \in [0, 1]$
- Optimize equation (6.1) with CMA-ES
Optimal regularity
- Employ SVD (equation (8.1))
- Transform initial parameters (equation (8.2))
- Transform initial step size (equation (8.3))
Design optimization

Figure 9.1: Summary of our recommended steps to initialize a high-performing design optimization.

because they are more efficient than global kernels and lead to results of equivalent quality. For exploration or imprecise information we recommend global triharmonic kernels or Wendland kernels with a large support radius because their resulting quality of design solutions does not depend that much on the center distribution, which induces robustness. Independent of the optimal center distribution, or even independent of the employed linear deformation method, tuning the deformation matrix itself for optimal regularity and thereby enhancing the convergence speed is a must-do as a last step. Following these recommendations (summarized in Figure 9.1) the tuned deformation setup should enable an efficient evolutionary design optimization process.

But, our detailed analysis of optimal compromises relies only on RBF tests. A possibility to show the general character of our preference analysis is the analysis of optimal FFD setups. But, this requires a proper definition of variability for FFD, and possibly for any linear deformation first, which is an issue for future work. Another issue is to show if our proposed concept performs in an automotive design scenario as well as in our test cases. But, because our mathematical models are based on general problem-independent concepts their application in this more complex real-world application seems promising. Besides automotive product design further application areas (like train, ship, or aircraft optimization) for our models are closely related and can be tested. However, we can push the idea of generalization in another direction. We analyzed our models for deformation-based evolutionary design optimization with a CMA-ES as the algorithm. Different evolutionary methods for design problems can be tested to show the independence with respect to the chosen algorithm. Furthermore, alternative non-deformation-based, but still linear, scenarios can be evaluated with our quality criteria. So far variability, regularity, and improvement potential are defined for linear mappings only. Thus, the highest level of generalization is their transformation to arbitrary representations, e.g., continuous mappings.

9 Bibliography

[AH12] Anne Auger and Nikolaus Hansen. Tutorial CMA-ES: evolution strategies and covariance matrix adaptation. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 827–848, 2012.

[AHN14] Olivier Amoignon, Jiri Hradil, and Jan Navratil. A numerical study of adaptive FFD in aerodynamic shape optimization. In *Proceedings of 52nd Aerospace Sciences Meeting*, pages 899–909, 2014.

[AK14] Elaine Angelino and Varun Kanade. Attribute-efficient evolvability of linear functions. In *Proceedings of the 5th conference on Innovations in theoretical computer science*, pages 287–300, 2014.

[Alt94] Lee Altenberg. The evolution of evolvability in genetic programming. *Advances in genetic programming*, 3:47–74, 1994.

[ANMD16] Nikola Aulig, Emily Nutwell, Stefan Menzel, and Duane Detwiler. Preference-based topology optimization of body-in-white structures for crash and static loads, 2016. 14th International LS-DYNA Users Conference.

[AOCP14] Jorge A. Soria Alcaraz, Gabriela Ochoa, Martin Carpio, and Hector Puga. Evolvability metrics in adaptive operator selection. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 1327–1334, 2014.

[Aul11] Nikola Aulig. Evolutionary optimization of the robustness and evolvability of design solutions. Master's thesis, Technische Universität Darmstadt, 2011.

[AZB15] Jascha Achenbach, Eduard Zell, and Mario Botsch. Accurate face reconstruction through anisotropic fitting and eye correction. In *Proceedings of Vision, Modeling & Visualization*, pages 1–8, 2015.

[Bai15] Zhong-Zhi Bai. On preconditioned iteration methods for complex linear systems. *Journal of Engineering Mathematics*, 93(1):41–60, 2015.

[BAK13] Bogdan Burlacu, Michael Affenzeller, and Michael Kommenda. On the evolutionary behavior of genetic programming with constants optimization. In *Computer Aided Systems Theory - EUROCAST 2013*, volume 8111 of *Lecture Notes in Computer Science*, pages 284–291. Springer, 2013.

[BBR09] Robert Brcina, Stephan Bode, and Matthias Riebisch. Optimisation process for maintaining evolvability during software evolution. In *Proceedings of IEEE International Conference and Workshop on the Engineering of Computer Based Systems*, pages 196–205, 2009.

[BCCV09] Maroun Bercachi, Philippe Collard, Manuel Clergue, and Sebastien Verel. Do not choose representation just change: An experimental study in states based EA. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 1799–1806, 2009.

[BdS10] Nigel P. A. Browne and Marcus V. dos Santos. Adaptive representations for improving evolvability, parameter control, and parallelization of gene expression programming. *Applied Computational Intelligence and Soft Computing*, 2010, 2010.

[BFM00] Thomas Bäck, David B. Fogel, and Zbigniew Michalewicz. *Evolutionary computation 1: Basic algorithms and operators*, volume 1. CRC press, Boca Raton, Florida, USA, 2000.

[BKP$^+$10] Mario Botsch, Leif Kobbelt, Mark Pauly, Pierre Alliez, and Bruno Lévy. *Polygon mesh processing*. CRC press, Boca Raton, Florida, USA, 2010.

[BLOA06] Jesse D. Bloom, Sy T. Labthavikul, Christopher R. Otey, and Frances H. Arnold. Protein stability promotes evolvability. In *Proceedings of the National Academy of Sciences*, pages 5869–5874, 2006.

[BP03] Mark A. Bedau and Norman H. Packard. Evolution of evolvability via adaptation of mutation rates. *Biosystems*, 69(2):143–162, 2003.

[Bro09] John F. Y. Brookfield. Evolution and evolvability: celebrating Darwin 200. *Biology Letters*, 5(1):44–46, 2009.

[Bro13] Rachael L. Brown. What evolvability really is. *British Journal for the Philosophy of Science*, 65(3):549–572, 2013.

[BS07] Hans-Georg Beyer and Bernhard Sendhoff. Robust optimization–a comprehensive survey. *Computer methods in applied mechanics and engineering*, 196(33):3190–3218, 2007.

[BS08] Mario Botsch and Olga Sorkine. On linear variational surface deformation methods. *IEEE transactions on visualization and computer graphics*, 14(1):213–230, 2008.

[BSJ11] Gerrit Becker, Michael Schäfer, and Antony Jameson. An advanced NURBS fitting procedure for post-processing of grid-based shape optimizations. In *Proceedings of 49th Aerospace Sciences Meeting*, pages 891–909, 2011.

[BV04] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge University Press, 2004.

[CBG$^+$14] Emiliano Costa, Marco E. Biancolini, Corrado Groth, Ubaldo Cella, Gregor Veble, and Matej Andrejasic. RBF-based aerodynamic optimization of an industrial glider. In *Proceedings of International CAE Conference*, 2014.

[CBL89] Sheng Chen, Stephen A Billings, and Wan Luo. Orthogonal least squares methods and their application to non-linear system identification. *International Journal of control*, 50(5):1873–1896, 1989.

[CC08] Chin-Chieh Chiang and John P. Chandler. An approximate equation for the condition numbers of well-scaled matrices. In *Proceedings of the IAJC-IJME International Conference*, volume 36, 2008.

[CJS13] Tianyou Chai, Yaochu Jin, and Bernhard Sendhoff. Evolutionary complex engineering optimization: opportunities and challenges. *IEEE Computational Intelligence Magazine*, 8(3):12–15, 2013.

[CL11] Bor-Sen Chen and Ying-Po Lin. On the interplay between the evolvability and network robustness in an evolutionary biological network: A systems biology approach. *Evolutionary Bioinformatics*, 7:201–233, 2011.

[CL13a] Bor-Sen Chen and Ying-Po Lin. A unifying mathematical framework for genetic robustness, environmental robustness, network robustness and their trade-off on phenotype robustness in biological networks. Part I: Gene regulatory networks in systems and evolutionary biology. *Evolutionary Bioinformatics*, 9:43–68, 2013.

[CL13b] Bor-Sen Chen and Ying-Po Lin. A unifying mathematical framework for genetic robustness, environmental robustness, network robustness and their trade-offs on phenotype robustness in biological networks. Part III: Synthetic gene networks in synthetic biology. *Evolutionary Bioinformatics*, 9:87–109, 2013.

[CL13c] Bor-Sen Chen and Ying-Po Lin. A unifying mathematical framework for genetic robustness, environmental robustness, network robustness and their tradeoff on phenotype robustness in biological networks. Part II: Ecological networks. *Evolutionary Bioinformatics*, 9:69–85, 2013.

[ČLM13] Matej Črepinšek, Shih-Hsi Liu, and Marjan Mernik. Exploration and exploitation in evolutionary algorithms: A survey. *ACM Computing Surveys (CSUR)*, 45(3):35–42, 2013.

[CRLL12] Timur Chabuk, James Reggia, Jason Lohn, and Derek Linden. Causally-guided evolutionary optimization and its application to antenna array design. *Integrated Computer-Aided Engineering*, 19(2):111–124, 2012.

[CWMC18] Pablo Catalán, Andreas Wagner, Susanna Manrubia, and José A Cuesta. Adding levels of complexity enhances robustness and evolvability in a multilevel genotype–phenotype map. *Journal of The Royal Society Interface*, 15(138), 2018.

[CXP$^+$09] Jie Chen, Bin Xin, Zhihong Peng, Lihua Dou, and Juan Zhang. Optimal contraction theorem for exploration–exploitation tradeoff in search and optimization. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 39(3):680–691, 2009.

[Dan90] Wayne W. Daniel. *Applied nonparametric statistics*. Houghton Mifflin, Boston, Massachusetts, USA, 1990.

[DMF10] P. Durr, C. Mattiussi, and D. Floreano. Genetic representation and evolvability of modular neural controllers. *IEEE Computational Intelligence Magazine*, 5(3):10–19, 2010.

[DPAM02] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*, 6(2):182–197, 2002.

[dPVvZvB17] Frits de Prenter, Clemens V. Verhoosel, Gert J. van Zwieten, and E. Harald van Brummelen. Condition number analysis and preconditioning of the finite cell method. *Computer Methods in Applied Mechanics and Engineering*, 316:297–327, 2017.

[DPWP10] Jeremy A. Draghi, Todd L. Parsons, Günter P. Wagner, and Joshua B. Plotkin. Mutational robustness can facilitate adaptation. *Nature*, 463(7279):353–355, 2010.

[Dre17] Stefan Dresselhaus. Evaluation of the performance of randomized FFD control grids. Master's thesis, Bielefeld University, 2017.

[DS10] Constantine Dovrolis and Jeffrey Todd Streelman. Evolvable network architectures: What can we learn from biology? *ACM SIGCOMM Computer Communication Review*, 40(2):72–77, 2010.

[ELPZ97] Yuval Eldar, Michael Lindenbaum, Moshe Porat, and Yehoshua Y. Zeevi. The farthest point strategy for progressive image sampling. *IEEE Transactions on Image Processing*, 6(9):1305–1315, 1997.

[Fel09] Vitaly Feldman. Robustness of evolvability. In *Proceedings of Conference on Learning Theory*, pages 277–292, 2009.

[Fle13] Roger Fletcher. *Practical methods of optimization*. John Wiley & Sons, 2013.

[FLS10] Christina Fang, Jeho Lee, and Melissa A. Schilling. Balancing exploration and exploitation through structural design: The isolation of subgroups and organizational learning. *Organization Science*, 21(3):625–642, 2010.

[FS11] Regina Frei and Giovanna Di Marzo Serugendo. Concepts in complexity engineering. *International Journal of Bio-Inspired Computation*, 3(2):123–139, 2011.

[FW06] Marie-Anne Felix and Andreas Wagner. Robustness and evolution: concepts, insights and challenges from a developmental model system. *Heredity*, 100(2):132–140, 2006.

[GJLA14] Sam F. Greenbury, Iain G. Johnston, Ard A. Louis, and Sebastian E. Ahnert. A tractable genotype–phenotype map modelling the self-assembly of protein quaternary structure. *Journal of The Royal Society Interface*, 11(95):249–261, 2014.

[GJS12] Carlotta Giannelli, Bert Jüttler, and Hendrik Speleers. THB-splines: The truncated basis for hierarchical splines. *Computer Aided Geometric Design*, 29(7):485–498, 2012.

[GLMOB11] Edgar Galván-López, James McDermott, Michael O'Neill, and Anthony Brabazon. Defining locality as a problem difficulty measure in genetic programming. *Genetic Programming and Evolvable Machines*, 12(4):365–401, 2011.

[GMH$^+$08] Lars Graening, Stefan Menzel, Martina Hasenjäger, Thomas Bihrer, Markus Olhofer, and Bernhard Sendhoff. Knowledge extraction from aerodynamic design data and its application to 3d turbine blade geometries. *Journal of Mathematical Modelling and Algorithms*, 7(4):329–344, 2008.

[GRO$^+$03] Jesús González, Ignacio Rojas, Julio Ortega, Héctor Pomares, Francisco Javier Fernandez, and Antonio F. Díaz. Multiobjective evolutionary optimization of the size, shape, and position parameters of radial basis function networks for function approximation. *IEEE Transactions on Neural Networks*, 14(6):1478–1495, 2003.

[GS00] Matthew R. Glickman and Katia Sycara. Evolvability and static vs. dynamic fitness. In *Workshop Proceedings of Artificial Life VII*, 2000.

[GSAL16] Sam F. Greenbury, Steffen Schaper, Sebastian E. Ahnert, and Ard A. Louis. Genetic correlations greatly increase mutational robustness and can both reduce and enhance evolvability. *PLoS computational biology*, 12(3):e1004773, 2016.

[GVL12] Gene H. Golub and Charles F. Van Loan. *Matrix computations*. Johns Hopkins University Press, Baltimore, Maryland, USA, 2012.

[GY00] J. Barry Gomm and Ding Li Yu. Selecting radial basis function network centers with recursive orthogonal least squares training. *IEEE Transactions on Neural networks*, 11(2):306–314, 2000.

[Han06] Nikolaus Hansen. The CMAt evolution strategy: a comparing review. In *Towards a new evolutionary computation*, pages 75–102. Springer, 2006.

[HB10] Ting Hu and Wolfgang Banzhaf. Evolvability and speed of evolutionary algorithms in light of recent developments in biology. *Journal of Artificial Evolution and Applications*, 2010:1–28, 2010.

[HFK$^+$06] Cecilia Haskins, Kevin Forsberg, Michael Krueger, D. Walden, and D. Hamelin. Systems engineering handbook, 2006. International Council On Systems Engineering.

[HHK92] William M. Hsu, John F. Hughes, and Henry Kaufman. Direct manipulation of free-form deformations. In *Proceedings of ACM SIGGRAPH*, pages 177–184, 1992.

[HMK16] Ary A. Hoffmann, Juha Merilä, and Torsten N. Kristensen. Heritability and evolvability of fitness and nonfitness traits: lessons from livestock. *Evolution*, 70(8):1770–1779, 2016.

[Hou92] David Houle. Comparing evolvability and variability of quantitative traits. *Genetics*, 130(1):195–204, 1992.

[HPH11] Thomas F. Hansen, Christophe Pélabon, and David Houle. Heritability is not evolvability. *Evolutionary Biology*, 38(3):258–277, 2011.

[HT11] Pauline C. Haddow and Andy M. Tyrrell. Challenges of evolvable hardware: past, present and the path to a promising future. *Genetic Programming and Evolvable Machines*, 12(3):183–215, 2011.

[HTL$^+$15] Thomas T. Hills, Peter M. Todd, David Lazer, A. David Redish, Iain D. Couzin, Cognitive Search Research Group, et al. Exploration versus exploitation in space, mind, and society. *Trends in cognitive sciences*, 19(1):46–54, 2015.

[IHMG08] Christian Igel, Verena Heidrich-Meisner, and Tobias Glasmachers. Shark. *The Journal of Machine Learning Research*, 9:993–996, 2008.

[IJH$^+$13] Benjamin Inden, Yaochu Jin, Robert Haschke, Helge Ritter, and Bernhard Sendhoff. An examination of different fitness and novelty based selection methods for the evolution of neural networks. *Soft Computing*, 17(5):753–767, 2013.

[ITN10] Hisao Ishibuchi, Noritaka Tsukamoto, and Yusuke Nojima. Diversity improvement by non-geometric binary crossover in evolutionary multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 14(6):985–998, 2010.

[JGPS09] Yaochu Jin, Robin Gruna, Ingo Paenke, and Bernhard Sendhoff. Evolutionary multi-objective optimization of robustness and innovation in redundant genetic representations. In *Proceedings of IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making*, pages 38–45, 2009.

[JGS09] Yaochu Jin, Robin Gruna, and Bernhard Sendhoff. Pareto analysis of evolutionary and learning systems. *Frontiers of Computer Science in China*, 3(1):4–17, 2009.

[Jin11] Yaochu Jin. Surrogate-assisted evolutionary computation: Recent advances and future challenges. *Swarm and Evolutionary Computation*, 1(2):61–70, 2011.

[JOS00] Yaochu Jin, Markus Olhofer, and Bernhard Sendhoff. On evolutionary optimization with approximate fitness functions. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 786–793, 2000.

[JT10] Yaochu Jin and Jens Trommler. A fitness-independent evolvability measure for evolutionary developmental systems. In *Proceedings of IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology*, pages 1–8, 2010.

[Kal00] Tatiana G. Kalganova. *Evolvable Hardware Design of Combinational Logic Circuits*. PhD thesis, Napier University Edinburgh, Scotland, 2000.

[Kel03] Carl T. Kelley. *Solving nonlinear equations with Newton's method*, volume 1. Siam, 2003.

[KG98] Marc Kirschner and John Gerhart. Evolvability. *Proceedings of the National Academy of Sciences*, 95(15):8420–8427, 1998.

[KG15] Smail Khalfallah and Adel Ghenaiet. Radial basis function-based shape optimization of centrifugal impeller using sequential sampling. In *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, pages 648–665, 2015.

[KVK+14] Junil Kim, Drieke Vandamme, Jeong-Rae Kim, Amaya Garcia Munoz, Walter Kolch, and Kwang-Hyun Cho. Robustness and evolvability of the human signaling network. *PLoS Computational Biology*, 10(7):e1003763, 2014.

[Lap10] André Laplume. *Heuristics for strategic ambidexterity: Balancing exploration and exploitation over time in varying environments*. PhD thesis, University of Manitoba, Winnipeg, Canada, 2010.

[Llo82] Stuart P Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.

[LM12] Henry Lehmann and Stefan Menzel. Evolvability as concept for the optimal design of free-form deformation control volumes. In *Proceedings of IEEE Congress on Evolutionary Computation*, pages 1–8, 2012.

[LMDM14] Christapher Lang, David Makhija, Alireza Doostan, and Kurt Maute. A simple and efficient preconditioning scheme for heaviside enriched XFEM. *Computational Mechanics*, 54(5):1357–1374, 2014.

[LOM+13] Minh Nghia Le, Yew Soon Ongand, Stefan Menzel, Yaochu Jin, and Bernhard Sendhoff. Evolution by adapting surrogates. *Evolutionary computation*, 21(2):313–340, 2013.

[LP07] Martin H. Luerssen and David M. W. Powers. Evolvability and redundancy in shared grammar evolution. In *Proceedings of IEEE Congress on Evolutionary Computation*, pages 370–377, 2007.

[LS11a] Joel Lehman and Kenneth O. Stanley. Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary Computation*, 19(2):189–223, 2011.

[LS11b] Joel Lehman and Kenneth O. Stanley. Improving evolvability through novelty search and self-adaptation. In *Proceedings of IEEE Congress on Evolutionary Computation*, pages 2693–2700, 2011.

[LS13] Joel Lehman and Kenneth O. Stanley. Evolvability is inevitable: Increasing evolvability without the pressure to adapt. *PloS one*, 8(4):e62186, 2013.

[LSN15] Stavros N. Leloudas, Giorgos A. Strofylas, and Ioannis K. Nikolos. Airfoil optimization using area-preserving free-form deformation. In *Proceedings of ASME 2015 International Mechanical Engineering Congress and Exposition*, 2015.

[LWK15] Alexander E. Lobkovsky, Yuri I. Wolf, and Eugene V. Koonin. Evolvability of an optimal recombination rate. *Genome biology and evolution*, 8(1):70–77, 2015.

[LWS16] Joel Lehman, Bryan Wilder, and Kenneth O. Stanley. On the critical role of divergent selection in evolvability. *Frontiers in Robotics and AI*, 3:45–51, 2016.

[Mac67] James MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, pages 281–297, 1967.

[MBBY06] Ali A. Mina, Dan Braha, and Yaneer Bar-Yam. *Complex Engineered Systems: Science Meets Technology*, chapter Complex Engineered Systems: A New Paradigm, pages 1–21. Springer, 2006.

[Men11] Stefan Menzel. Evolvable free-form deformation control volumes for evolutionary design optimization. In *Proceedings of IEEE Congress on Evolutionary Computation*, pages 1388–1395, 2011.

[MH17] Christine Mayer and Thomas F. Hansen. Evolvability and robustness: a paradox restored. *Journal of theoretical biology*, 430:78–85, 2017.

[MJ96] Ron MacCracken and Kenneth I. Joy. Free-form deformations with lattices of arbitrary topology. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 181–188, 1996.

[MLC16] Henok Mengistu, Joel Lehman, and Jeff Clune. Evolvability search: directly selecting for evolvability in order to study and produce it. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 141–148, 2016.

[MN16] Nicola Milano and Stefano Nolfi. Robustness to faults promotes evolvability: Insights from evolving digital circuits. *PloS one*, 11(7):e0158627, 2016.

[Mor17] Matthew Moreno. Evolvability: What is it and how do we get it? Bachelor thesis, University of Puget Sound, 2017.

[MOS05] Stefan Menzel, Markus Olhofer, and Bernhard Sendhoff. Application of free form deformation techniques in evolutionary design optimisation. In *Proceedings of 6th World Congress on Structural and Multidisciplinary Optimization*, 2005.

[MOS06] Stefan Menzel, Markus Olhofer, and Bernhard Sendhoff. Direct manipulation of free form deformation in evolutionary design optimisation. In *Proceedings of the International Conference on Parallel Problem Solving From Nature*, pages 352–361, 2006.

[MPN17] Nicola Milano, Paolo Pagliuca, and Stefano Nolfi. Robustness, evolvability and phenotypic complexity: Insights from evolving digital circuits, 2017. arXiv, 1712.04254.

[MS08] Stefan Menzel and Bernhard Sendhoff. Representing the change—free form deformation for evolutionary design optimization. In *Evolutionary computation in practice*, pages 63–86. Springer, 2008.

[MT10] Joanna Masel and Meredith V. Trotter. Robustness and evolvability. *Trends in Genetics*, 26(9):406–414, 2010.

[MY09] Pierre Maréchal and Jane J. Ye. Optimizing condition numbers. *SIAM Journal on Optimization*, 20(2):935–947, 2009.

[OBS04] Yutaka Ohtake, Alexander Belyaev, and Hans-Peter Seidel. 3D scattered data approximation with adaptive compactly supported radial basis functions. In *Proceedings of IEEE International Conference on Shape Modeling Applications*, pages 31–39, 2004.

[OJS01] Markus Olhofer, Yaochu Jin, and Bernhard Sendhoff. Adaptive encoding for aerodynamic shape optimization using evolution strategies. In *Proceedings of IEEE Congress on Evolutionary Computation*, pages 576–583, 2001.

[OSL11] Mauro Onori, Daniel Semere, and Bengt Lindberg. Evolvable systems: An approach to self-X production. *International Journal of Computer Integrated Manufacturing*, 24(5):506–516, 2011.

[OWCK16] Charles Ofria, Michael J. Wiser, and Rosangela Canino-Koning. The evolution of evolvability: Changing environments promote rapid adaptation in digital organisms. In *Proceedings of the European Conference on Artificial Life 13*, pages 268–275, 2016.

[Pig08] Massimo Pigliucci. Is evolvability evolvable? *Nature Reviews Genetics*, 9(1):75–82, 2008.

[PL00] Théodore Papadopoulo and Manolis IA Lourakis. Estimating the jacobian of the singular value decomposition: Theory and applications. In *Proceedings of the European Conference on Computer Vision*, pages 554–570, 2000.

[PM11] Joshua L. Payne and Jason H. Moore. Robustness, evolvability, and accessibility in the signal-integration space of gene regulatory circuits. In *Proceedings of the European Conference on Artificial Life*, pages 662–669, 2011.

[PMW14] Joshua L. Payne, Jason H. Moore, and Andreas Wagner. Robustness, evolvability, and the logic of genetic regulation. *Artificial Life*, 20(1):111–126, 2014.

[PT96] Les Piegl and Wayne Tiller. *The NURBS Book*. Monographs in Visual Communication. Springer Berlin Heidelberg, 1996.

[PW14] Joshua L. Payne and Andreas Wagner. The robustness and evolvability of transcription factor binding sites. *Science*, 343(6173):875–877, 2014.

[R C15] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2015.

[RAMB16] Andreas Richter, Jascha Achenbach, Stefan Menzel, and Mario Botsch. Evolvability as a quality criterion for linear deformation representations in evolutionary optimization. In *Proceedings of IEEE Congress on Evolutionary Computation*, pages 901–910, 2016.

[RAMB17] Andreas Richter, Jascha Achenbach, Stefan Menzel, and Mario Botsch. Multi-objective representation setups for deformation-based design optimization. In *Proceedings of 9th International Conference on Evolutionary Multi-Criterion Optimization*, pages 514–528, 2017.

[RBM15] Andreas Richter, Mario Botsch, and Stefan Menzel. Evolvability of representations in complex system engineering: a survey. In *Proceedings of IEEE Congress on Evolutionary Computation*, pages 1327–1335, 2015.

[RDMB18] Andreas Richter, Stefan Dresselhaus, Stefan Menzel, and Mario Botsch. Orthogonalization of linear representations for efficient evolutionary design optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 1356–1363, 2018.

[RM06] Joseph Reisinger and Risto Miikkulainen. Selecting for evolvable representations. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 1297–1304, 2006.

[RM07] Joseph Reisinger and Risto Miikkulainen. Acquiring evolvability through adaptive representations. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 1045–1052, 2007.

[RMB17] Andreas Richter, Stefan Menzel, and Mario Botsch. Preference-guided adaptation of deformation representations for evolutionary design optimization. In *Proceedings of IEEE Congress on Evolutionary Computation*, pages 2110–2119, 2017.

[Rot06] Franz Rothlauf. *Representations for Genetic and Evolutionary Algorithms*. Springer, Berlin, Germany, 2006.

[RSM05] Joseph Reisinger, Kenneth O. Stanley, and Risto Miikkulainen. Towards an empirical measure of evolvability. In *Proceedings of workshop on Genetic and evolutionary computation*, pages 257–264, 2005.

[Rus12] Popa Rustem. *Genetic Algorithms: An Overview with Applications in Evolvable Hardware*. InTech Open Access Publisher, 2012.

[SAESF17] Jorge A. Soria-Alcaraz, Andrés Espinal, and Marco A. Sotelo-Figueroa. Evolvability metric estimation by a parallel perceptron for on-line selection hyperheuristics. *IEEE Access*, 5:7055–7063, 2017.

[Sch07] Robert Schaback. A practical guide to radial basis functions, 2007. online: http://num.math.uni-goettingen.de/schaback/teaching/texte.html.

[SD94] Nidamarthi Srinivas and Kalyanmoy Deb. Muiltiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary computation*, 2(3):221–248, 1994.

[SdLSA$^+$10] Aurora Torres Soto, Eunice Esther Ponce de Leon-Senti, Arturo Hernández Aguirre, María Dolores Torres Soto, and Elva Díaz-Díaz. A robust evolvable system for the synthesis of analog circuits. *Computación y Sistemas*, 13(4):409–421, 2010.

[SIHE14] Luís F. Simões, Dario Izzo, Evert Haasdijk, and Agoston Endre Eiben. Self-adaptive genotype-phenotype maps: neural networks as a meta-representation. In *International Conference on Parallel Problem Solving from Nature*, pages 110–119, 2014.

[SKL05] Emanuele Stomeo, Tatiana Kalganova, and Cyrille Lambert. Analysis of genotype size for an evolvable hardware system. In *Proceedings of International Enformatika Conference Prague*, pages 74–79, 2005.

[SKVS97] Bernhard Sendhoff, Martin Kreutz, and Werner Von Seelen. A condition for the genotype-phenotype mapping: Causality. In *Proceedings of the International Conference on Genetic Algorithms*, pages 73–80, 1997.

[SMB12] Daniel Sieger, Stefan Menzel, and Mario Botsch. A comprehensive comparison of shape deformation methods in evolutionary design optimization. In *Proceedings of the International Conference on Engineering Optimization*, 2012.

[SMB14] D. Sieger, S. Menzel, and M. Botsch. RBF morphing techniques for simulation-based design optimization. *Engineering with Computers*, 30(2):161–174, 2014.

[SMB15] Daniel Sieger, Stefan Menzel, and Mario Botsch. On shape deformation techniques for simulation-based design optimization. In *New Challenges in Grid Generation and Adaptivity for Scientific Computing*, pages 281–303. Springer, 2015.

[SMC12] Tom Seaton, Julian F. Miller, and Tim Clarke. An ecological approach to measuring locality in linear genotype to phenotype maps. In *Proceedings of the European Converence on Genetic Programming*, pages 170–181, 2012.

[SN14] David Shorten and Geoff Nitschke. How evolvable is novelty search? In *Proceedings of the IEEE International Conference on Evolvable Systems*, pages 125–132, 2014.

[SP86] Thomas W. Sederberg and Scott R. Parry. Free-form deformation of solid geometric models. In *Proceedings of ACM SIGGRAPH*, pages 151–159, 1986.

[Sri13] Atul Kumar Srivastava. Novel crossover for evolvable hardware. *World Applied Sciences Journal*, 21(10):1408–1414, 2013.

[Ste04] Kim Sterelny. *Modularity in development and evolution*, chapter Symbiosis, Evolvability and Modularity, pages 490–516. University of Chicago Press, 2004.

[Suz03] Hideaki Suzuki. An example of design optimization for high evolvability: string rewriting grammar. *Biosystems*, 69(2):211–221, 2003.

[SZPJ13] Chaoli Sun, Jianchao Zeng, Jengshyang Pan, and Yaochu Jin. Similarity-based evolution control for fitness estimation in particle swarm optimization. In *Proceedings of the IEEE symposium on Computational Intelligence in Dynamic and Uncertain Environments (CIDUE)*, pages 1–8. IEEE, 2013.

[TDB97] Lloyd N. Trefethen and III David Bau. *Numerical Linear Algebra*. Society for Industrial and Applied Mathematics, 1997.

[TM14] Danesh Tarapore and Jean-Baptiste Mouret. Comparing the evolvability of generative encoding schemes. In *Artificial Life: Proceedings of the International Conference on the Synthesis and Simulation of Living Systems*, pages 55–62, 2014.

[Tor04] Jim Torresen. An evolvable hardware tutorial. In Jürgen Becker, Marco Platzner, and Serge Vernalde, editors, *Field Programmable Logic and Application*, Lecture Notes in Computer Science, pages 821–830. Springer, 2004.

[Tou03] Marc Toussaint. On the evolution of phenotypic exploration distributions. In *Proceedings of the Foundations of Genetic Algorithms*, pages 169–182, 2003.

[TR14] Ann Thorhauer and Franz Rothlauf. On the locality of standard search operators in grammatical evolution. In *Proceedings of the International Conference on Parallel Problem Solving From Nature*, pages 465–475, 2014.

[UD11] Mihaela Ulieru and René Doursat. Emergent engineering: a radical paradigm shift. *International Journal of Autonomous and Adaptive Communications Systems*, 4(1):39–60, 2011.

[UNS12] Arnold B. Urken, Arthur "Buck" Nimz, and Tod M. Schuck. Designing evolvable systems in a framework of robust, resilient and sustainable engineering analysis. *Advanced Engineering Informatics*, 26(3):553–562, 2012.

[Val07] Leslie Valiant. Evolvability. In *Mathematical Foundations of Computer Science 2007*, pages 22–43. Springer, 2007.

[Val12] Paul Valiant. Distribution free evolvability of polynomial functions over all convex loss functions. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, pages 142–148, 2012.

[VC14] Roby Velez and Jeff Clune. Novelty search creates robots with general skills for exploration. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 737–744, 2014.

[VGJS11] Anhtu Vuong, Carlotta Giannelli, Bert Jüttler, and Bernd Simeon. A hierarchical approach to adaptive local refinement in isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 200(49):3554–3567, 2011.

[WA96] Günter P. Wagner and Lee Altenberg. Perspectives: Complex adaptations and the evolution of evolvability. *Evolution*, 50(3):967–976, 1996.

[WA11] James M. Whitacre and Sergei P. Atamas. The diversity paradox: How nature resolves an evolutionary dilemma, 2011. arXiv, 1112.3115.

[Wag05] Andreas Wagner. Robustness, evolvability, and neutrality. *FEBS Letters*, 579(8):1772–1778, 2005.

[Wag08] Andreas Wagner. Robustness and evolvability: a paradox resolved. *Proceedings of the Royal Society B: Biological Sciences*, 275(1630):91–100, 2008.

[Wag17] Andreas Wagner. Information theory, evolutionary innovations and evolvability. *Philosophical Transactions of the Royal Society A*, 372(1735), 2017.

[WBM14] Yifei Wang, Joanna Bryson, and Stephen Matthews. Evolving evolvability in the context of environmental change: A gene regulatory network (grn) approach. In *Artificial Life: Proceedings of the International Conference on the Synthesis and Simulation of Living Systems*, 2014.

[WCT12] Thomas Weise, Raymond Chiong, and Ke Tang. Evolutionary optimization: Pitfalls and booby traps. *Journal of Computer Science and Technology*, 27(5):907–936, 2012.

[Wei09] Thomas Weise. Global optimization algorithms-theory and application. *Self-published*, 2009.

[Wei15] Iain Weir. Spearman's correlation, 2015. Retrieved from statstutor: http://www.statstutor.ac.uk/resources/uploaded/spearmans.pdf.

[Wen04] Holger Wendland. *Scattered data approximation*. Cambridge University Press, Cambridge, England, 2004.

[Whi10] James M. Whitacre. Degeneracy: a link between evolvability, robustness and complexity in biological systems. *Theoretical Biology and Medical Modelling*, 7(1):1–17, 2010.

[WHK15] Andrew M. Webb, Julia Handl, and Joshua Knowles. How much should you select for evolvability. In *Proceedings of the 2015 European Conference on Artificial Life*, pages 487–494, 2015.

[WLZY17] Mang Wang, Bin Li, Guofu Zhang, and Xin Yao. Population evolvability: Dynamic fitness landscape analysis for population-based metaheuristic algorithms. *IEEE Transactions on Evolutionary Computation*, pages 1–14, 2017.

[WRBY10] James M. Whitacre, Philipp Rohlfshagen, Axel Bender, and Xin Yao. The role of degenerate robustness in the evolvability of multi-agent systems in dynamic environments. In *Proceedings of the International Conference on Parallel Problem Solving From Nature*, pages 284–293, 2010.

[WRBY12] James M. Whitacre, Philipp Rohlfshagen, Axel Bender, and Xin Yao. Evolutionary mechanics: new engineering principles for the emergence of flexibility in a dynamic and uncertain world. *Natural computing*, 11(3):431–448, 2012.

[WS98] Andrew R Webb and Simon Shannon. Shape-adaptive radial basis functions. *IEEE Transactions on Neural Networks*, 9(6):1155–1166, 1998.

[YSTY16] Zhenyu Yang, B Sendhoff, Ke Tang, and Xin Yao. Target shape design optimization by evolving B-splines with cooperative coevolution. *Applied Soft Computing*, 48:672–682, 2016.

[Zha00] Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1330–1334, 2000.

[ZWS05] Jianmin Zheng, Yimin Wang, and Hock Soon Seah. Adaptive T-spline surface fitting to z-map models. In *Proceedings of the 3rd international conference on Computer graphics and interactive techniques in Australasia and South East Asia*, pages 405–411, 2005.

[ZYSD16] Ye Zhang, Guowei Yang, Zhenxu Sun, and Guo Dilong. A general shape optimization method based on FFD approach with application to a high-speed train. *Journal of Multidisciplinary Engineering Science and Technology*, 3:6181–6188, 2016.