Honda Research Institute EU
Honda Research Institute Europe GmbH
https://www.honda-ri.de/

Orthogonalization of Linear Representations for Efficient Evolutionary Design Optimization

Andreas Johannes Richter, Stefan Dresselhaus, Stefan Menzel, Mario Botsch

2018

Preprint:

This is an accepted article published in The Genetic and Evolutionary Computation Conference (GECCO) 2018. The final authenticated version is available online at: https://doi.org/10.1145/3205455.3205568

Orthogonalization of Linear Representations for Efficient Evolutionary Design Optimization

Andreas Richter
Computer Graphics Group, Bielefeld University
Bielefeld, Germany

Stefan Dresselhaus
Computer Graphics Group, Bielefeld University
Bielefeld, Germany

Stefan Menzel
Honda Research Institute Europe
Offenbach, Germany
stefan.menzel@honda-ri.de

Mario Botsch
Computer Graphics Group, Bielefeld University
Bielefeld, Germany
botsch@techfak.uni-bielefeld.de

The article has been accepted for publication in the GECCO 2018

"© 2018 Copyright held by the owner/author(s). Publication rights licensed to Association for Computing Machinery. This is the author's version of the work. It is posted here for your personal use. Not for redistribution. The definitive Version of Record was published in The Genetic and Evolutionary Computation Conference (GECCO) 2018, http://dx.doi.org/10.1145/3205455.3205568."

https://doi.org/10.1145/3205455.3205568

Orthogonalization of Linear Representations for Efficient Evolutionary Design Optimization

Andreas Richter
Computer Graphics Group, Bielefeld University
Bielefeld, Germany

Stefan Dresselhaus
Computer Graphics Group, Bielefeld University
Bielefeld, Germany

Stefan Menzel
Honda Research Institute Europe
Offenbach, Germany
stefan.menzel@honda-ri.de

Mario Botsch
Computer Graphics Group, Bielefeld University
Bielefeld, Germany
botsch@techfak.uni-bielefeld.de

ABSTRACT

Real-world evolutionary design optimizations of complex shapes can efficiently be solved using linear deformation representations, but the optimization performance crucially depends on the initial deformation setup. For instance, when modeling the deformation by radial basis functions (RBF) the convergence speed depends on the condition number of the involved kernel matrix, which previous work therefore tried to optimize through careful placement of RBF kernels. We show that such representation-specific techniques are inherently limited and propose a generic, representation-agnostic optimization based on orthogonalization of the deformation matrix. This straightforward black-box optimization projects *any* given linear deformation setup to optimal condition number without changing its design space, which, as we show through extensive numerical experiments, can boost the convergence speed of evolutionary optimizations by up to an order of magnitude.

CCS CONCEPTS

• **Mathematics of computing → Evolutionary algorithms**;

KEYWORDS

Orthogonalization, Evolutionary Design Optimization, Representation, Shape Deformation

ACM Reference Format:
Andreas Richter, Stefan Dresselhaus, Stefan Menzel, and Mario Botsch. 2018. Orthogonalization of Linear Representations for Efficient Evolutionary Design Optimization. In *GECCO '18: Genetic and Evolutionary Computation Conference, July 15–19, 2018, Kyoto, Japan.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3205455.3205568

1 INTRODUCTION

In contemporary industry practice, digital product development of complex designs involves a number of major ingredients, like efficient optimization algorithms, shape representations, and simulation tools eventually enriched by surrogate models as performance predictors. The set of optimization parameters is typically determined by the degrees of freedom of the underlying shape representation, but this strategy is intractable for highly complex geometries based on mesh structures. It has been shown, however, that complex geometric models can efficiently be modified using shape deformation or shape morphing techniques, using the deformation's controllers as optimization parameters [25]. This approach has successfully been applied to the optimization of various real-world objects [7, 28, 34].

Due to the black-box character of many simulation tools that are used for computing the object's performance, evolutionary optimization is favored because of its capability to find global optima without requiring gradient information. A typical vehicle design optimization that employs a computational fluid dynamics simulation (CFD) for aerodynamic performance computation is depicted in Figure 1. Because each performance evaluation of a design variant amounts to a computationally expensive CFD simulation, fast convergence to a high-quality solution (i.e., requiring few optimization iterations) is of high practical relevance.

The deformation representation has a strong influence on the convergence speed of the optimization. Deformation representations are therefore set up to provide the required design flexibility with a minimum number of control parameters. Linear deformations, such as kernel-based *radial basis functions* (RBFs) [28] or lattice-based *free-form deformation* (FFD) [26], are state-of-the-art methods to realize shape morphing. For an initial setup, RBF kernels or FFD control points have to be distributed on or around the shape, either uniformly spaced or in a target-adapted manner according to design heuristics or prior knowledge. Optimizing the initial setup then amounts to finding optimal positions for kernels or control points with respect to suitable quality criteria [22]. However, as we will demonstrate in this paper, this representation-specific approach is inherently limited when aiming at fast converging setups.

In this paper we approach the problem of setting up representations for fast convergence from a different, more fundamental angle and propose a representation-agnostic optimization that is based on the deformation matrix. RBF and FFD (like all linear deformation techniques) can be written in matrix notation, where a constant deformation matrix maps the basis functions' coefficients (the *genotype*) to per-vertex displacements of the mesh geometry (the *phenotype*). Interestingly, the expected optimization behavior

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
GECCO '18, July 15–19, 2018, Kyoto, Japan
© 2018 Copyright held by the owner/author(s). Publication rights licensed to Association for Computing Machinery.
ACM ISBN 978-1-4503-5618-3/18/07…$15.00
https://doi.org/10.1145/3205455.3205568

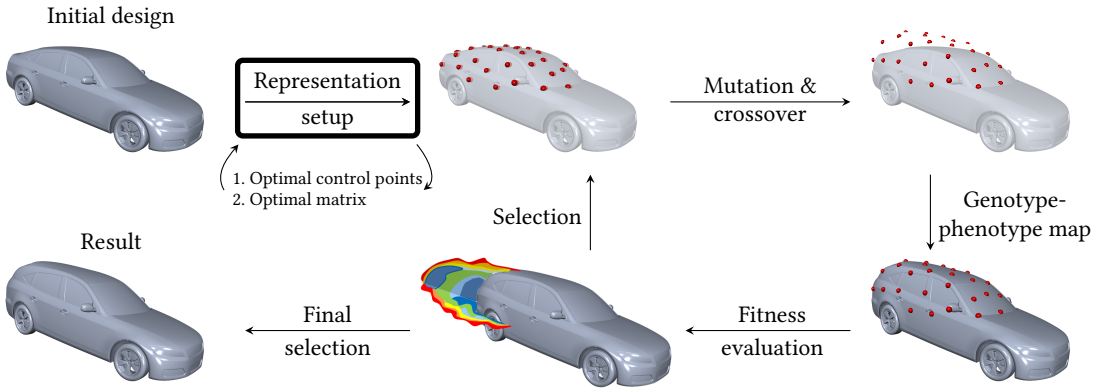

Figure 1: Evolutionary automotive design optimization: Our focus is on the optimal initial representation setup, i.e., an optimal genotype→phenotype mapping, which is determined by the distribution of control points (e.g., the RBF kernels shown in red) and defined by the numerical properties of the resulting deformation matrix.

can be estimated from the deformation matrix alone. Inspired by the concept of evolvability [30], we proposed the two criteria *regularity* and *improvement potential* and showed them to strongly correlate with convergence speed and quality of the final design – although for RBF-based optimizations only [21].

We first show that the criteria regularity and improvement potential generalize to other linear deformation methods besides radial basis functions, namely lattice-based free-form deformation and surface-based thin shell deformation (Section 3). Inspired by the general correlation between convergence speed and regularity, which itself is defined through the matrix condition number, we formulate an optimization based on the deformation matrix itself (Section 4): For any given deformation setup, a matrix orthogonalization leads to provably optimal regularity while preserving both the design/phenotype space and the improvement potential. Our 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 (Section 5).

2 RELATED WORK

For the utilization of evolutionary optimization in practical applications it is important to guarantee an adequate convergence speed to a high-performing solution. This is especially true for design optimization with computationally expensive fitness evaluations, such as computational fluid dynamics simulations for aerodynamics or finite element simulations for structural reliability in automotive optimization. Different acceleration strategies can be applied either alone or in combination. Surrogate models [16, 20] approximate the original fitness functions by a computationally less expensive one and thereby reduce the cost of each fitness evaluation, at the price of less accurate models and requiring sufficient training data. Choosing proper operators for mutation, recombination, and selection of an evolutionary algorithm tackles the problem from another angle [2]. Tuning these operators may lead to fewer iterations required for convergence depending on the shape of the fitness landscape.

In contrast, our focus is on enhancing representations, in particular linear deformations, to provide an optimal initial parameter set on which the evolution can efficiently optimize.

For design optimization based on linear deformations the setup is typically realized in a manual step, e.g., in [28] for automotive design optimization, in [7] for glider optimization, or in [34] for design optimization of trains. Manual approaches to distribute the representation's handles heavily rely on the designer's experience w.r.t. the deformation formulation, i.e., the potential of deformation variations, and the problem at hand, i.e., where to expect the largest fitness improvements. The manual representation setup fully defines the deformation matrix and thereby determines the convergence speed and potential fitness improvement. However, as we will show in Section 4, these matrices can be drastically improved by an automatic optimization technique.

Evaluating and optimizing representation setups requires suitable quality criteria. These criteria are motivated by prominent concepts, like evolvability [17, 30], locality [24], and causality [27, 32], which are known to promote an evolutionary search. The quality criteria *regularity* and *improvement potential*, derived from the concept of evolvability, have been shown in [21] to strongly correlate with the convergence speed and the resulting quality of an evolutionary design optimization, respectively. We apply these criteria for our analysis and further extend the RBF deformations of [21] by free-form deformation and shell deformations.

In [22] RBF kernel distributions are optimized 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, [19, 28] switched from indirect to direct manipulation of RBF and FFD to improve causality and increase convergence speed. We show 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 or FFD control points, and therefore without the need for an expensive optimization procedure.

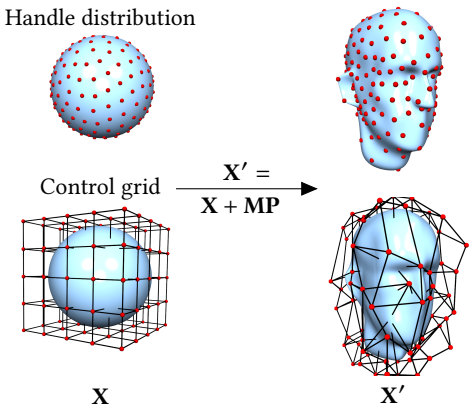

Figure 2: Displacements of kernel positions (RBF), handle positions (shells), or control points (FFD) constitute the optimization parameters P, which control the deformation of an initial design X to a design variant X′ through the linear deformation operator/matrix M.

Regularity is strongly linked to the concepts of locality/causality, which aim for representations where small changes in genotype result in small changes in phenotype. This preservation of local neighborhoods in the genotype–phenotype mapping allows for more efficient explorative evolutionary search [24, 32]. However, typically the mutation or crossover operators are addressed with these concepts, e.g., with locality in genetic programming [11] or grammatical evolution [29], or with causality for genetic representations in antenna design [6]. 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 [3, 9, 18]. Our orthogonalization employs the singular value decomposition, which is used in [13] to increase the performance of evolutionary optimization. But while the decomposition is apply to the mutation operator in [13], we optimize the underlying deformation representation.

3 PERFORMANCE CRITERIA FOR LINEAR DEFORMATION REPRESENTATIONS

In complex design optimization scenarios, geometric objects are typically represented as surface meshes, consisting of m vertices with positions $\mathbf{x}_1, \ldots, \mathbf{x}_m \in \mathbb{R}^3$, which are connected by edges forming polygonal faces (mostly triangles or quadrangles). The large number of vertices for real-world models disqualifies their direct usage as optimization parameters. Instead, one employs shape deformation techniques that can be controlled by a comparatively small number of parameters $\mathbf{p}_1, \ldots, \mathbf{p}_n$ with $n \ll m$.

Deformation techniques frequently employed in design optimization are free-form deformation (FFD) [14, 26] and radial-basis functions (RBFs) [33]. In addition, thin shell models are used to

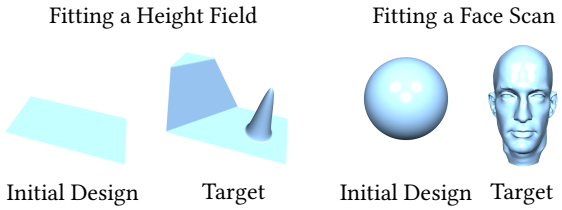

Figure 3: Two test scenarios of [21]: Deforming a plane to a height field (left) and a sphere to a face scan (right).

describe surface deformation in computational mechanics or computer graphics [4, 5]. These techniques are *linear deformations* and can therefore be written in matrix notation as

$$\mathbf{X}' = \mathbf{X} + \mathbf{MP}\,,$$

where the $(m \times n)$ matrix $\mathbf{M}$ constitutes the *deformation representation*. It maps the initial design $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_m)$ to a shape variant $\mathbf{X}'$ (the *phenotype*) through a displacement $\mathbf{MP}$ that is controlled by the parameter vector $\mathbf{P} = (\mathbf{p}_1, \ldots, \mathbf{p}_n)$ (the *genotype*). While the matrix $\mathbf{M}$ is constructed during initialization and kept constant, the parameters $\mathbf{P}$ are varied by a human designer or an optimization algorithm to produce shape variations (Figure 2).

In general, the matrix entry $\mathbf{M}_{i,j}$ 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* of [21], $\mathbf{M}_{i,j} = \phi\left(\left\|\mathbf{x}_i - \mathbf{c}_j\right\|\right)$, with the kernel ϕ positioned at center $\mathbf{c}_j$. For *free-form deformation*, which we additionally analyze in this paper, $\mathbf{M}_{i,j} = N^3_{j_1}(u_i) N^3_{j_2}(v_i) N^3_{j_3}(w_i)$, 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 [26] (implementation details in [10]). 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 [4, 5].

Since the geometric model $\mathbf{X}$ is given, setting up the deformation representation amounts to distributing kernel positions (RBF), defining the lattice of control points (FFD), or selecting some vertices as handle points (shells). For RBF deformations we distinguish between *indirect manipulation* (im), where the parameters $\mathbf{p}_j$ correspond to RBF coefficients, and *direct manipulation* (dm), where the parameters $\mathbf{p}_j$ are interpreted as displacements at the RBF kernels. For FFD the parameters $\mathbf{p}_j$ correspond to displacements of control points or they correspond to prescribed handle displacements for shells.

The parameters $\mathbf{p}_j$ then correspond to RBF coefficients (indirect manipulation), RBF kernel displacements (direct manipulation, see [21]), control point displacements (FFD), or prescribed handle displacements (shells), respectively.

Estimating the *expected* performance of the design optimization (or ultimately optimizing the initial setup for a better-performing optimization) requires suitable quality criteria for evaluating the deformation representation. We drew inspiration from the concept of evolvability [30] and defined the three criteria *variability*, *regularity*, and *improvement potential* in our previous work [21]. We only

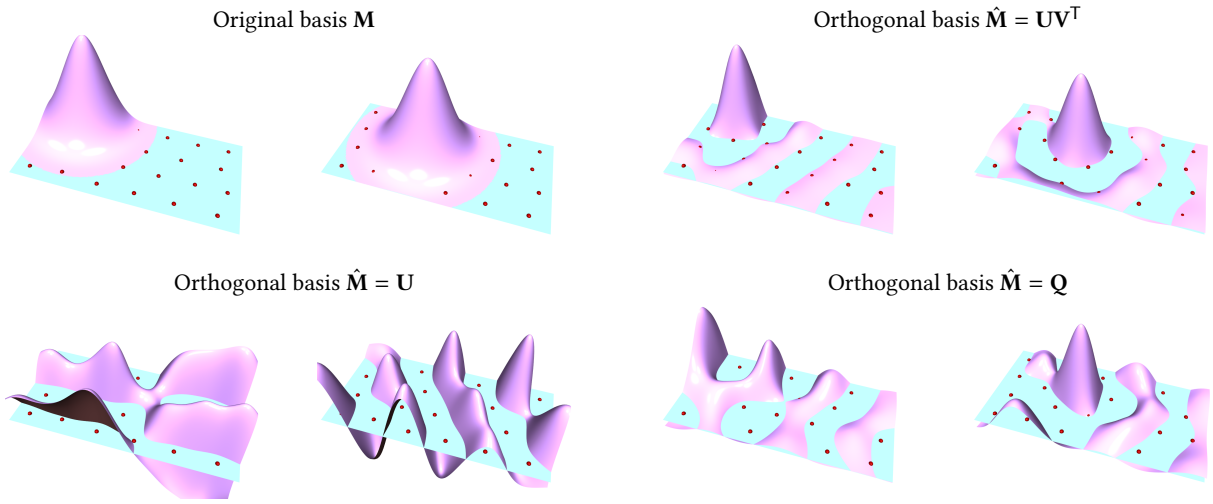

Figure 4: Visualization of two original basis functions and different orthogonal versions: Two columns of the deformation matrix M (top left). Their closest orthogonal approximation as $\mathbf{U}\mathbf{V}^\mathsf{T}$ of SVD (top right) are geometrically more similar to the original bases than using only the U matrix of SVD (bottom left) or the Q matrix of a QR decomposition (bottom right).

consider the latter two criteria (since variability basically measures the *constant* number of control parameters):

Regularity is defined as $R(\mathbf{M}) = 1/\kappa(\mathbf{M})$, where $\kappa(\mathbf{M})$ is the condition number of matrix $\mathbf{M}$ [12]. It characterizes the expected convergence speed of an evolutionary optimization utilizing the deformation representation $\mathbf{M}$.

Improvement potential measures a representation's potential to improve the fitness of a design. It is defined by how well the deformation setup $\mathbf{M}$ can approximate the (estimated, approximated) gradient $\mathbf{G}$ of the fitness function: $P(\mathbf{M}) = 1 - \left\|(\mathbf{I} - \mathbf{M}\mathbf{M}^+)\mathbf{G}\right\|_F^2$, with $\mathbf{M}^+$ being the pseudo-inverse of $\mathbf{M}$ and $\|\cdot\|_F$ the matrix Frobenius norm [12]. For instance, when the designer has a rough guess of the direction towards an optimal shape based on her experience or data analytics, then this direction can be used as the estimated gradient $\mathbf{G}$.

For RBF deformations, we showed strong correlations between regularity and convergence speed as well as between improvement potential and final fitness [21]. We confirm and extend our statistical analysis by adding free-form deformation and shell deformation to the RBF-based experiments, using the same two test scenarios: Deforming a plane to fit a target height field and deforming a sphere to fit a target face scan (Figure 3). We refer to [21] for details on these experiments and their evaluation.

For the height field experiment, we first generate 100 random setups each for RBF deformation (distributing 25 kernels), for shell deformation (selecting 25 control handles), and for free-form deformation (5×5 control grid). While distributing kernels and handles is straightforward, we ensure the feasibility of the control grids by making sure that they are not self-intersecting (see [10] for details). For each random deformation setup we evaluate its regularity and improvement potential, and then relate those to the total number of iterations and final fitness value of a CMA-ES optimization [1]. We utilize a (1,10)-CMA-ES of the shark library [15] with an initial step size s of 0.001 and default settings otherwise. The monotone Spearman's correlation coefficient [8, 31] between regularity and convergence speed is 0.71 for FFD and shells together, 0.93 for RBF only, and 0.91 for the complete set of RBF, FFD, and shells. The correlation between improvement potential and fitting quality is 0.72 (FFD, shells together), 0.74 (RBF only), and 0.75 (RBF, FFD, shells). All correlations are significant with a $p < 10^{-4}$.

The 3D face fitting works analogously, but we use 68 kernels/handles for RBF/shell deformation and a $4 \cdot 4 \cdot 4$ control lattice for FFD. The correlation between regularity and convergence speed is 0.77 (FFD and shells), 0.94 (RBF), and 0.94 (RBF, FFD, shells), and that between improvement potential and fitting quality is 0.62 (FFD and shells), 0.83 (RBF), and 0.8 (RBF, FFD, shells). Also here all results are significant with $p < 10^{-4}$.

While the correlations of FFD and shells are not as strong as those of RBFs, the total set of all experiments clearly demonstrates the general validity of the two criteria regularity and improvement potential, which motivates the optimization of deformation setups with respect to these measures.

4 MATRIX ORTHOGONALIZATION FOR OPTIMAL REGULARITY

In previous approaches [23] RBF deformation setups are optimized with respect to regularity and improvement potential through computationally expensive evolutionary 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 (compare [22, 23]). 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 $\kappa(\mathbf{M}) = \sigma_1/\sigma_n$ ($\sigma_1 > \cdots > \sigma_n$), which is true for orthogonal matrices only. For our non-square $m \times n$ matrix $\mathbf{M}$ with $m > n$,

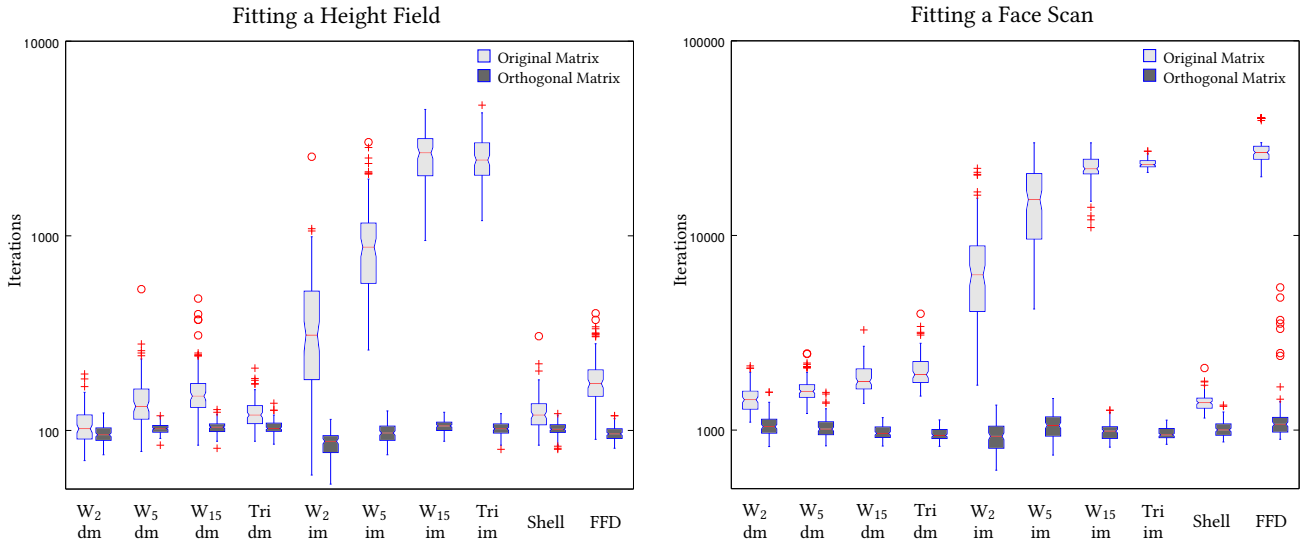

Figure 5: Pairwise comparison of the convergence speed of CMA-ES (#iterations until convergence) for different deformation representations (column pairs), for original representation M (light color) versus orthogonalized representation $\hat{\mathbf{M}}$ (dark color), for fitting height fields (left) and face scan (right), averaged over 100 random setups each. The RBF tests are 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.

this requires the matrix columns $\mathbf{m}_1, \ldots, \mathbf{m}_n$ 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 $\mathbf{m}_j$. 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 $\mathbf{M}$, instead of indirectly manipulating it through careful placement of RBF kernels or FFD control points. The columns $\mathbf{m}_j$ form a basis of the n-dimensional phenotype sub-space of $\mathbb{R}^m$, since any displacement $\mathbf{MP}$ can be written as a linear combination $\sum_{j=1}^{n} \mathbf{m}_j \mathbf{p}_j^\mathsf{T}$. Asking these basis vectors to be orthogonal and of unit length is in agreement with the concepts of *locality* [24] and *causality* [27, 32], 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 $\mathbf{M}$ 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) [12]. Although computationally most expensive, we employ the SVD, since it is numerically most stable and yields the orthogonal matrix closest to the original one [35], i.e., it changes the deformation basis the least, as visualized in Figure 4.

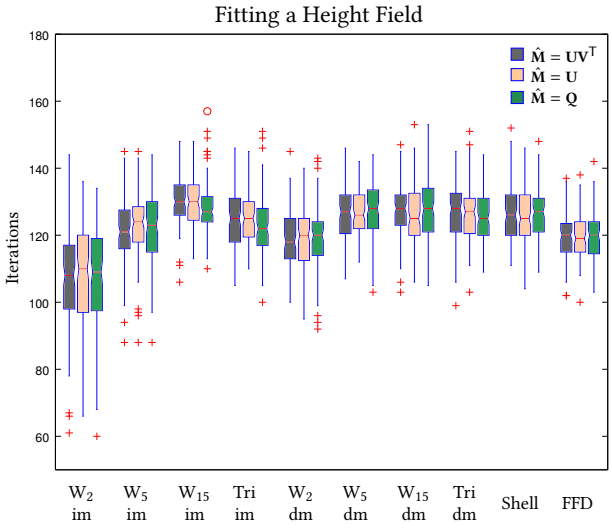

Figure 6: Different orthogonalization techniques based on the singular value decomposition $\mathbf{M} = \mathbf{U}\Sigma\mathbf{V}^\mathsf{T}$ or the QR decomposition $\mathbf{M} = \mathbf{QR}$ yield equivalent convergence speed.

We decompose $\mathbf{M}$ using the thin SVD [12] into $\mathbf{M} = \mathbf{U}\Sigma\mathbf{V}^\mathsf{T}$, with orthogonal matrices $\mathbf{U} \in \mathbb{R}^{m \times n}$ and $\mathbf{V} \in \mathbb{R}^{n \times n}$ and a diagonal matrix $\Sigma \in \mathbb{R}^{n \times n}$ containing the singular values $\Sigma_{i,i} = \sigma_i$. Removing the singular values Σ (or setting all σ_i to one) yields the orthogonalized deformation matrix (as the product of two orthogonal matrices):

$$\hat{\mathbf{M}} = \mathbf{U}\mathbf{V}^\mathsf{T}. \tag{1}$$

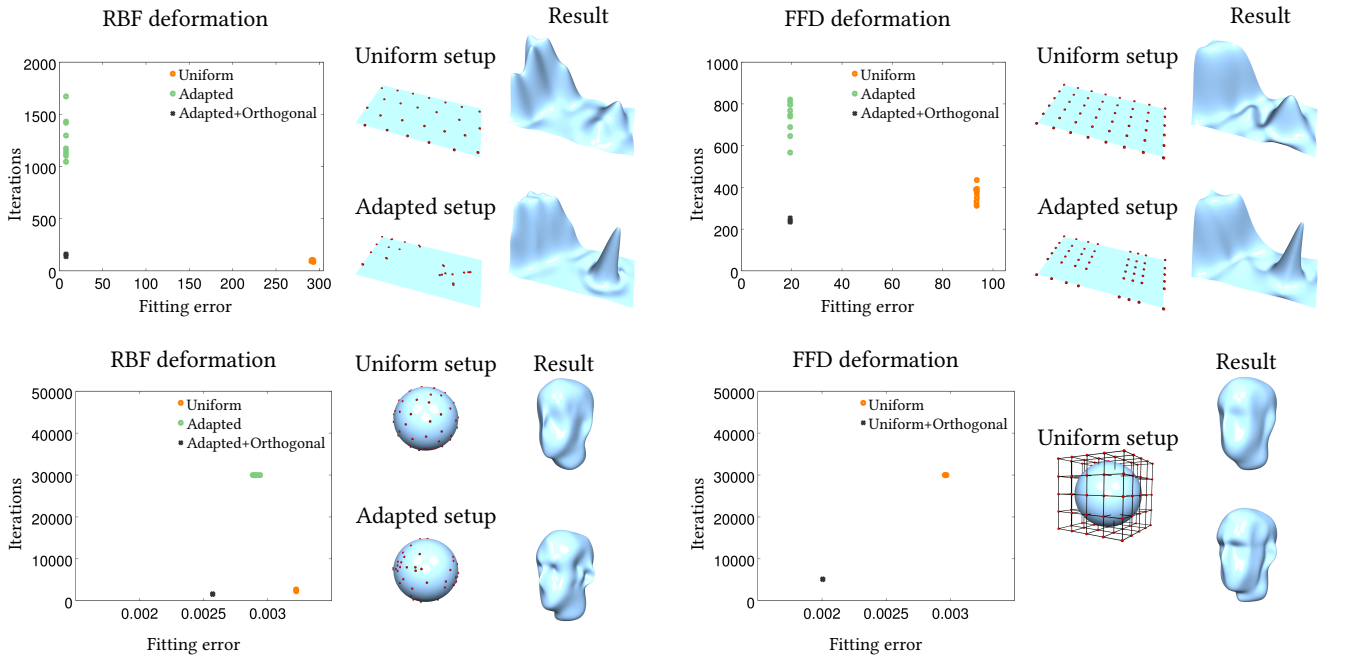

Figure 7: Matrix orthogonalization for custom-tailored setups. We analyze height field fitting (top) and face fitting (bottom), using RBF deformation (left) and FFD (right). The uniformly distributed setups (having higher regularity) converge faster than the target-adapted ones, but the adaptive setups (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.

By construction $\kappa(\hat{\mathbf{M}}) = 1$, therefore $\hat{\mathbf{M}}$ has optimal regularity, locality, and causality, and we can expect faster convergence (see next section). Since the columns of $\hat{\mathbf{M}}$ span the same phenotype space as the columns of $\mathbf{M}$, the improvement potential is unchanged and the optimization can reach the same optimum in both variants. In practice, starting from a given representation $\mathbf{M}$, we first compute its orthogonal version $\hat{\mathbf{M}}$ and then perform the evolutionary optimization (more efficiently) based on $\hat{\mathbf{M}}$. The resulting optimal parameter vector $\hat{\mathbf{P}}$ is finally mapped back to the original representation $\mathbf{M}$ 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.

5 RESULTS AND EVALUATION

In order to analyze the orthogonalization's effect on the convergence speed in actual evolutionary optimizations, we compare optimization runs *with* and *without* orthogonalization of the deformation representation for the experiments described in Section 3. For the two test scenarios (height field, face scan) and the different types of representations (RBF, FFD, shell) we run an evolutionary optimization for 100 random setups each. Following [21], 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 (see [21] for details). Like before, we utilize a (1,10)-CMA-ES of the shark library [15] with manually determined optimal initial step sizes s for each representation. For the 1D height field (Figure 3) we chose $s = 0.001$ for the unmodified and $s = 0.01$ for the orthogonal setting. For fitting 3D faces (Figure 3) we chose $s = 0.001$ and $s = 0.05$, respectively. Otherwise the default settings of the shark library are applied. The results in Figure 5 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 $\hat{\mathbf{M}} = \mathbf{U}\mathbf{V}^{\mathsf{T}}$ from Equation (1) due to its numerical stability and the fact that it minimally modifies the input setup, Figure 6 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 [21], locality [24], and causality [27, 32].

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 coefficients that yield this desired deformation. Direct manipulation has better regularity and converges faster in a design optimization [21], as also confirmed

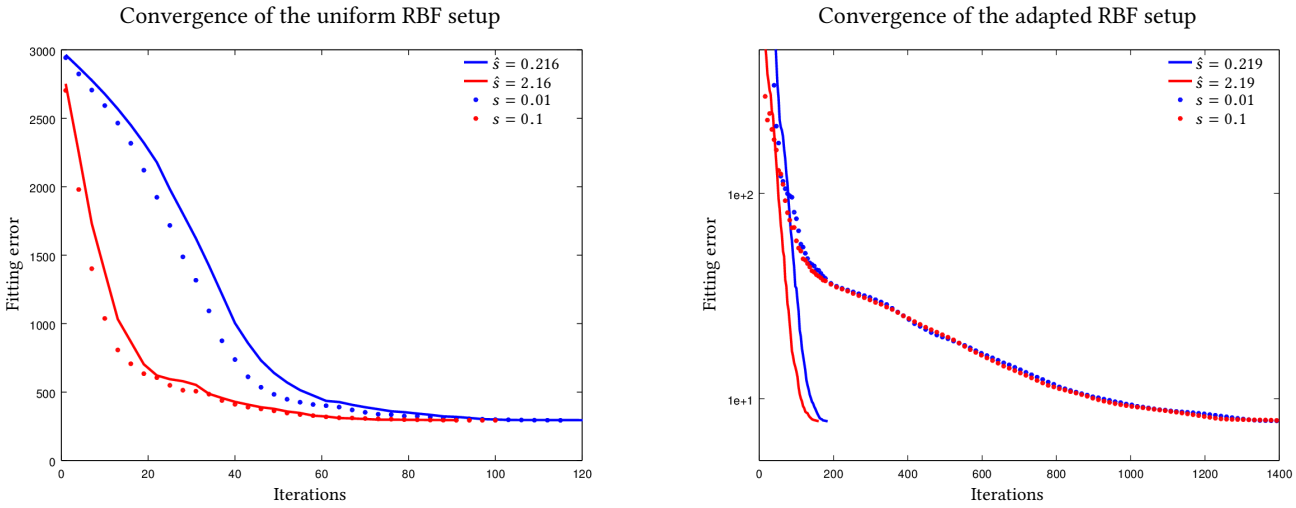

Figure 8: Convergence plots for the height field fitting scenario with a uniform RBF setup (left) and the adapted RBF setup (right). The initial step sizes s for the original representations $\mathbf{M}$ have been converted to $\hat{s}$ for the orthogonalized representations $\hat{\mathbf{M}}$ through Equation (2). For two initial step sizes s (red and blue), the convergence behavior for $\hat{\mathbf{M}}$ (solid line) matches that of the original setup $\mathbf{M}$ (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).

by our experiments (Figure 5). The switch from indirect to direct manipulation can be considered as a matrix preconditioning that improves regularity to a certain extend [21]. In this view, our orthogonalization provides a superior alternative that projects the matrix to *optimal* regularity, thereby improving convergence speed even more (Figure 5).

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*. Figure 7 shows the results for height field fitting and face scan fitting averaged over 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 [22]. 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, 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 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 $\mathbf{M}$ (e.g., from the designer's knowledge), it can be converted to the orthogonal representation $\hat{\mathbf{M}}$ by compensating for the normalization of the matrix columns $\mathbf{m}_j$:

$$\hat{s} = s \cdot \frac{1}{n} \sum_{j=1}^{n} \left\| \mathbf{m}_j \right\| , \tag{2}$$

where $\left\| \mathbf{m}_j \right\|$ is the length of the jth column of $\mathbf{M}$ (see Figure 8).

6 CONCLUSION

For any design optimization, convergence speed and solution quality are crucial aspects to guarantee short product development cycles. This is especially true for real-world 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 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, which we showed to generalize to free-form deformation and shell deformation beyond the previously evaluated RBF deformations [21]. 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.

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.

In the future we will analyze our representation optimization for more complex geometric models and more complex fitness functions, such as the automotive design optimization shown in Figure 1. Additionally, because regularity and improvement potential are currently only defined for linear deformation representations, we are anxious to generalize these criteria to non-linear representations.

ACKNOWLEDGMENTS

Andreas Richter gratefully acknowledges the financial support from Honda Research Institute Europe (HRI-EU). Mario Botsch is supported by the Cluster of Excellence Cognitive Interaction Technology "CITEC" (EXC 277) at Bielefeld University, funded by the German Research Foundation (DFG).

REFERENCES

[1] Anne Auger and Nikolaus Hansen. 2012. Tutorial CMA-ES: evolution strategies and covariance matrix adaptation.. In *GECCO '12: Proceedings of the 14th Annual Conference Companion on Genetic and Evolutionary Computation*. 827–848.

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

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

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

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

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

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

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

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

[10] Stefan Dresselhaus. 2017. *Evaluation of the Performance of Randomized FFD Control Grids*. Master's thesis. Bielefeld University.

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

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

[13] Jesús González, Ignacio Rojas, Julio Ortega, Héctor Pomares, Francisco Javier Fernandez, and Antonio F Díaz. 2003. 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 (2003), 1478–1495.

[14] William M. Hsu, John F. Hughes, and Henry Kaufman. 1992. Direct Manipulation of Free-Form Deformations. In *Proceedings of ACM SIGGRAPH*. 177–184.

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

[16] Yaochu Jin and Jürgen Branke. 2005. Evolutionary optimization in uncertain environments—a survey. *IEEE Transactions on evolutionary computation* 9, 3 (2005), 303–317.

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

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

[19] Stefan Menzel, Markus Olhofer, and Bernhard Sendhoff. 2006. Direct Manipulation of Free Form Deformation in Evolutionary Design Optimisation. In *Proceedings of the International Conference on Parallel Problem Solving From Nature*. 352–361.

[20] Yew S. Ong, Prasanth B. Nair, and Andrew J. Keane. 2003. Evolutionary optimization of computationally expensive problems via surrogate modeling. *American Institute of Aeronautics and Astronautics (AIAA) journal* 41, 4 (2003), 687–696.

[21] Andreas Richter, Jascha Achenbach, Stefan Menzel, and Mario Botsch. 2016. Evolvability as a Quality Criterion for Linear Deformation Representations in Evolutionary Optimization. In *Proceedings of IEEE Congress on Evolutionary Computation*. 901–910.

[22] Andreas Richter, Jascha Achenbach, Stefan Menzel, and Mario Botsch. 2017. Multi-objective Representation Setups for Deformation-based Design Optimization. In *Proceedings of 9th International Conference on Evolutionary Multi-Criterion Optimization*. 514–528.

[23] Andreas Richter, Stefan Menzel, and Mario Botsch. 2017. Preference-guided Adaptation of Deformation Representations for Evolutionary Design Optimization. In *Proceedings of IEEE Congress on Evolutionary Computation*. 2110–2119.

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

[25] Jamshid A. Samareh. 2001. Survey of shape parameterization techniques for high-fidelity multidisciplinary shape optimization. *American Institute of Aeronautics and Astronautics (AIAA) journal* 39, 5 (2001), 877–884.

[26] Thomas W. Sederberg and Scott R. Parry. 1986. Free-Form Deformation of Solid Geometric Models. In *Proceedings of ACM SIGGRAPH*. 151–159.

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

[28] Daniel Sieger, Stefan Menzel, and Mario Botsch. 2012. A Comprehensive Comparison of Shape Deformation Methods in Evolutionary Design Optimization. In *Proceedings of the International Conference on Engineering Optimization*.

[29] Ann Thorhauer and Franz Rothlauf. 2014. On the Locality of Standard Search Operators in Grammatical Evolution. In *Proceedings of the International Conference on Parallel Problem Solving From Nature*. 465–475.

[30] Günter P. Wagner and Lee Altenberg. 1996. Perspectives: Complex Adaptations and the Evolution of Evolvability. *Evolution* 50, 3 (1996), 967–976.

[31] Iain Weir. 2015. Spearman's correlation. (2015). http://www.statstutor.ac.uk/resources/uploaded/spearmans.pdf

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

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

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

[35] Zhengyou Zhang. 2000. A Flexible New Technique for Camera Calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22, 12 (2000), 1330–1334.