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

Preference-guided Adaptation of Deformation Representations for Evolutionary Design Optimization

Andreas Johannes Richter, Stefan Menzel, Mario Botsch

2017

Preprint:

This is an accepted article published in IEEE Congress on Evolutionary Computation. The final authenticated version is available online at: https://doi.org/10.1109/CEC.2017.7969560 Copyright 2017 IEEE

Preference-guided Adaptation of Deformation Representations for Evolutionary Design Optimization

Andreas Richter* Stefan Menzel† Mario Botsch*
*Computer Graphics Group, Bielefeld University, Germany
†Honda Research Institute Europe, Germany

***Abstract*—A dynamic industrial design optimization requires high-quality optimization algorithms as well as adaptive representations to find the global solution for a given problem. For adapting the representation to changing environments or to new input we utilize the concept of evolvability, which in our interpretation consists of three criteria: *variability*, *regularity*, and *improvement potential*, where regularity and improvement potential characterize conflicting goals between exploration and exploitation. Our goal is the efficient adaptation of the representation according to a given preference weight between regularity and improvement potential. We propose a combination of two heuristics, Lloyd sampling and orthogonal least squares sampling, to initialize the adaptation process for a given preference weight. We show that this initialization improves the convergence speed of the adaptation process as well as the resulting fitness. We then realize a stepwise design optimization procedure by alternating the adaptation of the representation with optimization of the design. During the design optimization process we extract information which we exploit in the next adaptation phase. We show that an intermediate preference weight, balancing between regularity and improvement potential, allows to exploit this information and is robust to erroneous initial information. Thereby, we increase the performance of the whole design optimization process.**

I. Introduction

Modern industrial design development requires optimization processes that can cope with multi-disciplinary product specifications in dynamic environments. Biologically-inspired population-based evolutionary optimization algorithms are designed to handle these demands [1]. We focus on *deformation-based design optimization*: In an evolutionary optimization an input shape is deformed to improve with respect to some fitness function, e.g., aerodynamic drag in automotive design optimization. Hence, the deformation method/setup constitutes the representation in our scenario, i.e., the mapping from deformation parameters (genotype) to shape variations (phenotype).

It is well known that the initial deformation setup has a strong impact on the performance of the optimization process. The quality of such deformation representations, i.e., their potential performance, can be estimated through the concept of *evolvability* [1, 2], which can be quantified by the three sub-criteria *variability*, *regularity*, and *improvement potential* [3]. With regularity and improvement potential roughly corresponding to exploration and exploitation capabilities, respectively, these two quality criteria are conflicting.

The concept of evolvability can not only be used for analyzing existing setups, but also for their initial *generation*. In this case, we (evolutionary) optimize the *deformation setup* itself. Since our shape deformation framework is based on radial basis functions (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 performing a multi-objective optimization with respect to these two objectives and letting the designer choose a compromise setup on the Pareto front [4].

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 boundary conditions of a CFD simulation in automotive design. In this article, we perform the next step toward the goal of a truly dynamic evolutionary design optimization: In a static design optimization (non-varying fitness function or design target) we alternate between setup generation/adaptation (*where to place RBF kernels?*) and design optimization (*which RBF coefficients to use for shape deformation?*), as illustrated in Figure 1.

As a first contribution, we improve the automatic setup generation of [4]. Instead of computing the Pareto front through a multi-objective optimization and then selecting a setup on the Pareto front as a preference-based compromise between exploration and exploitation, we employ a preference-weighted single-objective optimization to derive the deformation setup. The involved optimization is initialized by an efficient combination of two kernel distribution strategies, namely *orthogonal least squares (OLS)* [5, 6] and *Lloyd sampling* [7], which is shown to improve the setup generation in terms of both quality and computational performance.

Tuning a deformation setup toward improvement potential (exploitation) 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 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. As a second contribution we analyze different preference weights for setup adaptation and show that an intermediate preference meets these two requirements.

978-1-5090-4601-0/17/$31.00 ©2017 IEEE

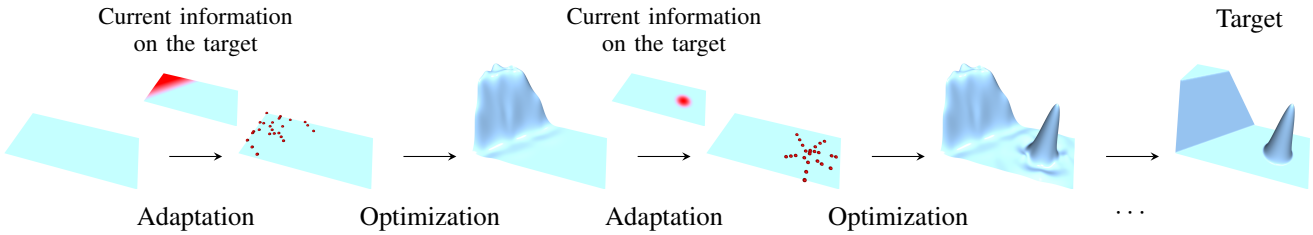

Fig. 1. Example of a stepwise design optimization process where the representation (RBF kernel distribution, red dots) is adapted in its domain (plane) according to information on the target (color coded estimation of the fitness gradient).

After discussing related work in Section II we give technical details to RBF deformations in Section III and their evaluation with evolvability in Section IV. Then, we discuss heuristics to initialize an adaptation process in Section V, which we embed in a stepwise design optimization in Section VI. We evaluate our concepts and show the results for heuristic initialization of the adaptation process and the results of the entire design optimization procedure in Section VII.

II. Related Work

The first task in a design optimization process with adaptive deformation representations is the generation of the initial deformation setup. Manual setup generation based on the designer's expertise [8] is employed for design optimization in general as well as for static setups for glider optimization [9], but those approaches are not suitable for adapting the deformation setup during an optimization process. Originally, deformation representations are employed in scattered data approximation, e.g., for approximating a target shape. In [6, 10–12] analytic data, e.g., gradient information of the given target or information from an additional test optimization, is utilized to initialize the deformation. These approaches are specialized to set up deformations purely focused on the approximation of one fixed target, and thus they neglect the exploration potential.

The representation setup for an evolutionary design optimization process using adaptive B-splines is targeted in [13, 14]. The optimization alternates between the approximation of a shape and the adaptation of the representation. To test whether a representation is beneficial for the optimization task, this task is performed for a few iterations. Like this the performance of a representation is measured by the objective function of the actual optimization task, but consequently more optimization runs are required. In [15] the representation is optimized implicitly by adding its parameters to the approximation problem. This increases the number of optimization parameters and thus leads to a slow optimization process for computationally expensive fitness evaluations. Moreover, the criterion for a high-quality representation again solely depends on the optimization target, omitting further aspects like exploration potential.

We alternate between the adaptation of the deformation representation and the optimization of the design, similar to [13, 14, 16]. For adapting the deformation setup we employ the concept of evolvability, which quantifies the potential to converge to the desired target (exploitation) as well as the potential to explore the design space. To evaluate the exploitation potential we require information about the targeted design (i.e. the fitness function). In contrast to [13, 14, 16], 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 [17], where three different preferences (0, 0.5, and 1) are analyzed for competing targets in topology optimization. In the next section we give the technical details of our deformation representation.

III. RBF Representations

In a shape optimization scenario, for instance in automotive product design, the design model to be optimized is typically represented by a surface polygon mesh, where the n mesh vertices $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^3$ represent points on the 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 one million vertices, thus making the direct optimization of vertex positions intractable.

Even for highly complex shapes the actual *deformations* applied during optimization are typically rather simple, low-frequency functions, which can therefore be controlled by a small number of parameters. Both free-form deformation (FFD) [8, 18, 19] and radial basis functions (RBFs) [8, 9] have been successfully employed in design optimization. In this paper we utilize RBF deformations, since their kernel-based setup is more flexible than lattice-based FFD representations.

The deformation function $\mathbf{u}(\mathbf{x})$, which maps deformation parameters to shape variations, is added to each vertex $\mathbf{x}_i$ of the initial design $\mathbf{X} = (\mathbf{x}_1^\mathsf{T}, \ldots, \mathbf{x}_n^\mathsf{T})^\mathsf{T}$, resulting in a shape variant $\mathbf{X}' = (\mathbf{x}_1'^\mathsf{T}, \ldots, \mathbf{x}_n'^\mathsf{T})^\mathsf{T}$ (Figure 2). The deformation function $\mathbf{u}(\mathbf{x})$ has the form:

$$\mathbf{u}(\mathbf{x}) \;=\; \sum_{j=1}^{m} \mathbf{w}_j \, \varphi(\|\mathbf{c}_j - \mathbf{x}\|) \;=:\; \sum_{j=1}^{m} \mathbf{w}_j \, \varphi_j(\mathbf{x}) \quad . \qquad (1)$$

Here, $\varphi_j(\mathbf{x}) = \varphi(\|\mathbf{c}_j - \mathbf{x}\|)$ denotes the j-th scalar-valued radial basis function, which is centered at $\mathbf{c}_j \in \mathbb{R}^3$ and weighted by the coefficient $\mathbf{w}_j \in \mathbb{R}^3$.

As the kernel function $\varphi\colon \mathbb{R} \to \mathbb{R}$ we employ globally-supported triharmonic thin-plate splines, φ_{tri}, as well as

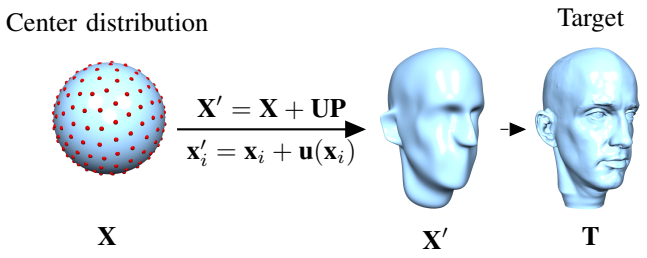

Fig. 2. The RBF deformation $\mathbf{u}$ transforms the initial mesh $\mathbf{X}$ to $\mathbf{X}'$ by translating each vertex $\mathbf{x}_i$ of $\mathbf{X}$ by the displacement $\mathbf{u}(\mathbf{x}_i)$. The distribution of the RBF kernels (red dots) is crucial for a high-performing fitting process of the mesh $\mathbf{X}$ to the target $\mathbf{T}$.

compactly-supported Wendland functions, φ_W, with support radii s varying from rather local to more global [20]:

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

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

Because the RBF deformation is linear in the weights we can express the deformation of a mesh $\mathbf{X}$ to $\mathbf{X}'$ in matrix notation

$$\mathbf{X}' = \mathbf{X} + \mathbf{UP}\,, \tag{2}$$

using an $(n \times m)$ deformation matrix $\mathbf{U}$, which is defined by the basis functions φ_j, and a $(m \times 3)$ parameter matrix $\mathbf{P}$, consisting of the weights $\mathbf{w}_j$ as deformation parameters (see [3] for details). The deformation matrix $\mathbf{U}$, which we call the deformation setup, depends on the employed kernel and the center distribution. These two aspects define the realizable deformations and thereby the performance of a design optimization process. Evaluating and constructing different deformation setups (different kernels and different center distributions) with the concept of evolvability allows us to initialize a high performing design optimization.

IV. Evolvability for Linear Deformation Setups

The biological concept of *evolvability* is a very promising approach to measure the *expected* performance of evolutionary processes [21]. We understand evolvability as a combination of three major attributes: *variability*, *regularity*, and *improvement potential* [2]. In [3] we proposed a mathematical formulation for the three attributes for linear deformation representations $\mathbf{U}$. Below we give a short summary of our model.

Variability $V(\mathbf{U})$ measures the potential of a deformation setup to explore the design space and is defined it as

$$V(\mathbf{U}) = \frac{\operatorname{rank}(\mathbf{U})}{n}\,, \tag{3}$$

where n is the number of vertices of a design and $\operatorname{rank}(\mathbf{U})$ denotes the rank of the matrix $\mathbf{U}$ [22]. In our scenario we assume a fixed number of centers, i.e., constant variability.

Regularity $R(\mathbf{U})$ is defined as

$$R(\mathbf{U}) = \kappa^{-1}(\mathbf{U})\,, \tag{4}$$

where κ is the condition number of a matrix [22]. The regularity of a deformation setup characterizes the expected convergence speed of an evolutionary optimization [3]. In [4] we showed that highly regular setups are uniformly distributed over the whole domain. Thus, regularity indirectly characterizes the exploration potential.

Improvement potential $P(\mathbf{U})$ measures a representation's potential to improve the fitness of a design, and therefore corresponds to exploitation. From a local point of view the most beneficial design variation is the (estimated) gradient $\mathbf{g}$ of the fitness function. Thus, we measure the improvement potential $P(\mathbf{U})$ as the approximation error with respect to this gradient, which leads to:

$$P(\mathbf{U}) = 1 - \left\|(\mathbf{I} - \mathbf{U}\mathbf{U}^{+})\mathbf{g}\right\|_2^2\,, \tag{5}$$

with $\mathbf{U}^{+}$ being the pseudo-inverse of $\mathbf{U}$ [22]. Because for complex design optimization applications the calculation of the fitness gradient is infeasible, we utilize designer knowledge or data from previous optimization runs to derive an estimation.

To setup deformations in the adaptation phase of a design optimization process we have to specify a preference between the conflicting targets regularity (exploration) and improvement potential (exploitation). Like in [4] we define an objective function balancing regularity and improvement potential according to a preference weight $\lambda \in [0, 1]$ as:

$$f_\lambda(\mathbf{U}) = \lambda R(\mathbf{U}) + (1 - \lambda) P(\mathbf{U})\ . \tag{6}$$

We use this objective function as the fitness for an evolutionary optimization to *setup* deformations in the *adaptation* phase of a design optimization process.

V. Adaptation Phase

In the adaptation phase of a design optimization process we optimize the deformation setup for a compromise between regularity (exploration) and improvement potential (exploitation) with an evolutionary method, aiming for Pareto optimal solutions. In [4] we showed that a weighted single-objective optimization (the covariance matrix adaptation evolution strategy, CMA-ES [23]) with equation (6) is capable of finding solutions close to the Pareto front much faster than a multi-objective optimization (the non-dominated sorting genetic algorithm II, NSGAII) with regularity and improvement potential (equations (4) and (5)) as objectives. Hence, we employ CMA-ES to adapt deformation setups.

Although evolutionary optimization processes have the possibility to find global optima, these optima may not be reached for real-world problems because the fitness landscape is very rugged or the dimension of the domain is too high [24]. The initialization of an evolutionary process has significant influence on its convergence speed and on the quality of its results. Thus, we aim at a good initialization to improve these two aspects. For a fixed kernel type and a fixed number of centers the optimization target for RBF deformation setups is the center distribution. In [4] we showed that heuristic center distributions for a preference solely on regularity or solely on improvement potential can be generated efficiently using Lloyd sampling and orthogonal least squares sampling (OLS), respectively. The resulting distributions are almost as good as

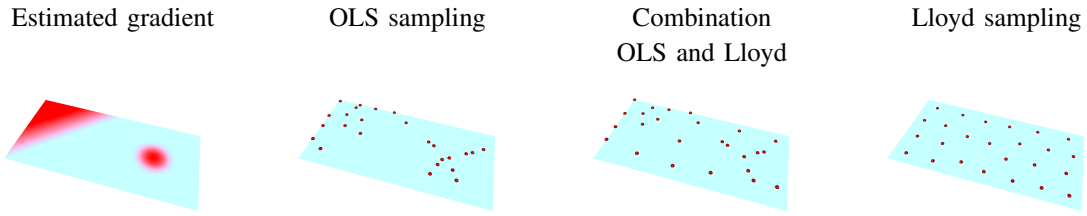

Fig. 3. OLS sampling results in adapted center distribution (center left) according to an estimated gradient (color coding left) whereas a Lloyd sampling results in a uniform distribution (right). The combination of OLS and Lloyd results in a compromise distribution (center right). We utilize these two methods for the initialization of an adaptation process of the center distribution.

optimized ones. Thus, these two methods are promising for the *initialization* of an evolutionary optimization of the center distribution. In the following, we *combine* Lloyd sampling and OLS sampling to construct center distributions according to a given preference weight, and use the resulting distributions as initialization for a single-objective optimization.

The Lloyd algorithm [7] (also known as k-means clustering [25]) distributes centers uniformly within a domain (Figure 3 right). The algorithm is initialized with a distribution and first clusters the domain: Each point of the domain is assigned to its closest center. Then, each center is moved to the barycenter of all domain points associated to it. This alternating procedure is iterated until convergence.

The OLS algorithm [5, 6] solves least squares problems in a greedy manner. In [3, 4] we motivated improvement potential (equation (5)) through the least squares approximation problem $\mathbf{g} = \mathbf{U}\mathbf{p}$ for parameters $\mathbf{p} = (p_1, \ldots, p_m)$. The OLS algorithm determines the influence of each parameter (or degree of freedom) on the approximation error to $\mathbf{g}$ and iteratively selects the parameter that reduces the least squares error the most. Like this it selects the most important parameters in a greedy manner. In our case each parameter p_j corresponds to an RBF center $\mathbf{c}_j$, such that OLS selects the RBF kernels best suited for approximating the estimated gradient $\mathbf{g}$ (see [4] for details). We initialize OLS with a large set of candidate centers (30×30) on a uniform grid and then select the best 25 ones, resulting in a gradient-adapted center distribution (Figure 3).

We now combine both methods according to a given preference $\lambda \in [0, 1]$, with weight λ for regularity and $(1 - \lambda)$ for improvement potential. Given m centers to be distributed we place $(1 - \lambda) \cdot m$ centers in a gradient-adapted manner (using OLS) and $\lambda \cdot m$ centers in a regular manner (using Lloyd sampling). For example, we distribute 25 centers with OLS for $\lambda = 0$ (Figure 3, center left), or we place 25 centers with the Lloyd sampling for $\lambda = 1$ (Figure 3, right), or we distribute 13 centers with OLS and 12 centers with Lloyd for $\lambda = 0.3$ (Figure 3, center right). In our implementation we first distribute $(1 - \lambda) \cdot m$ centers with OLS sampling and then distribute the remaining $\lambda \cdot m$ centers with the Lloyd algorithm. We initialize the Lloyd algorithm with the chosen OLS centers and fill up the remaining $\lambda \cdot m$ centers by farthest point sampling [26]: We add centers one by one such that they are as far away as possible from all other centers. Finally, we run the Lloyd algorithm as described above, but this time keep the OLS centers fixed. Note that the Lloyd algorithm moves centers slightly off a curved domain, e.g., a sphere; we simply project these centers back to the domain in the end.

This general approach allows the efficient construction of center distributions for many different preferences and domains. Utilizing these center distributions as initialization for an evolutionary optimization process to adapt the centers increases its convergence speed and improves the quality of the resulting adapted center distribution (see Section VII-A). We incorporate this adaptive procedure in the design optimization process, which we discuss in the next section.

VI. Stepwise Design Optimization

A complex design optimization process involves a deformation representation that can react and adapt to new information. To adapt the center distribution we utilize the concept described in the previous section in an alternating process, where we switch between the adaptation phase of the setup and the optimization phase of the design (Figure 1). To adapt centers with respect to improvement potential we need estimated gradient information. As a coarse estimation any direction in which the design can be improved is possible. We extract such a direction during the optimization process of the design for the following adaptation phase.

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 and use the difference between the vertex positions $\mathbf{x}_i \in \mathbf{X}$ of the initial design and the result after k iterations $\mathbf{x}'_i \in \mathbf{X}'$ to compute a new estimation of the gradient (following the approach in [27]). 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 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.

In the next section we discuss the results of our concept to perform the adaptation phase and then we analyze the whole design optimization process.

VII. Evaluation

In this section we evaluate the proposed adaptive stepwise optimization on two test scenarios: In the first one we fit a plane to a target height field by minimizing the squared difference between both functions (see Figure 6 and [3] for the details). This simple scenario allows us to analytically compute gradient information and hence to perfectly control the error in the "estimated" gradients. Afterward we extend this experiment to the fitting of a sphere to a target head model, both represented as triangle meshes.

We first analyze the adaptation of deformation setups, i.e., the RBF center distribution, utilizing the combined Lloyd and OLS sampling proposed in Section V. Second, we analyze the stepwise optimization by alternating setup adaptation and shape optimization.

A. Adaptation Phase

We perform setup adaptation (i.e., kernel distribution) by minimizing equation (6) for RBF deformation matrices, and analyze this optimization for three different preference weights ($\lambda = 0.0$, 0.5, and 1). The optimization is performed with CMA-ES [23] from the shark 3.1 library [28]. In [4] we already showed the feasibility of such a single-objective optimization for constructing Pareto-optimal deformation setups. We optimize 25 RBF centers $\mathbf{c}_j$, each having 2D coordinates in the plane, leading to a 50-dimensional optimization problem. With the default settings of the shark library this results in a CMA-ES with 7 parents and 15 offspring per iteration. 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.

We evaluate five tests for both a random initialization and the heuristic OLS/Lloyd initialization, and for three kernel

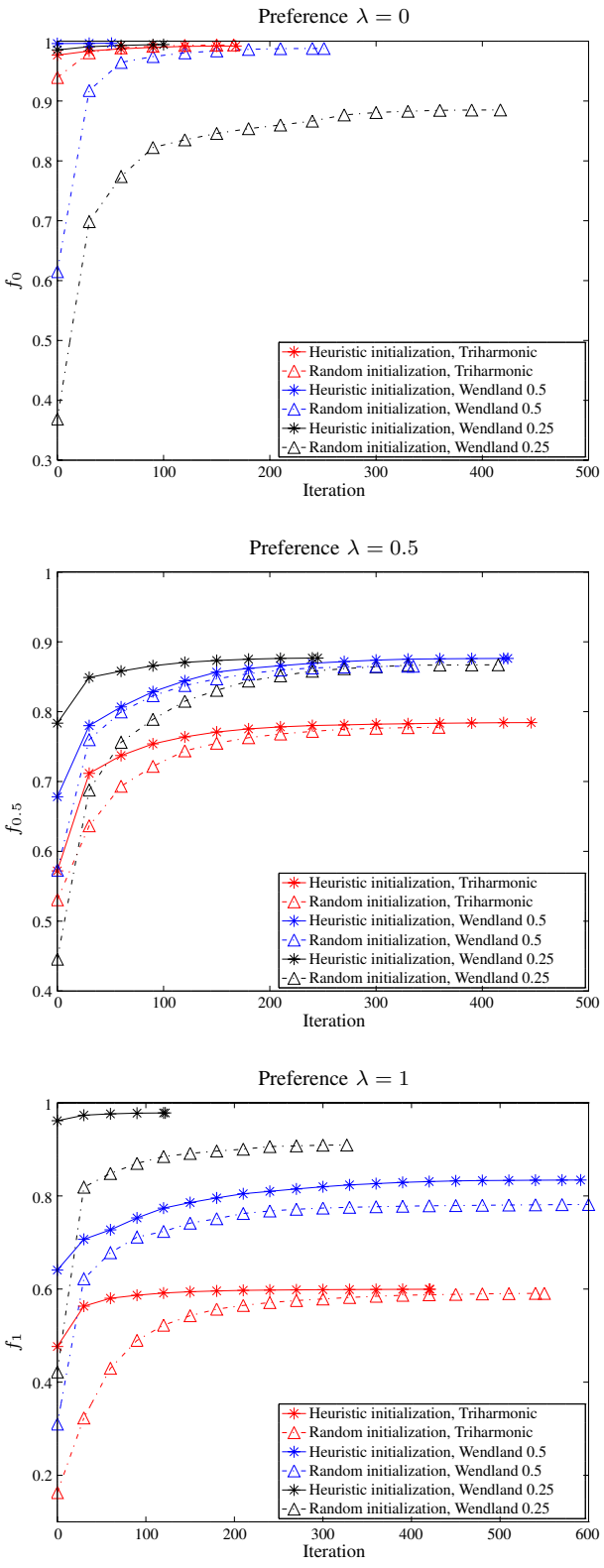

Fig. 4. A setup optimization with our heuristic initialization (solid lines) converges faster and yields better values than the optimization with a randomized initialization (dashed lines).

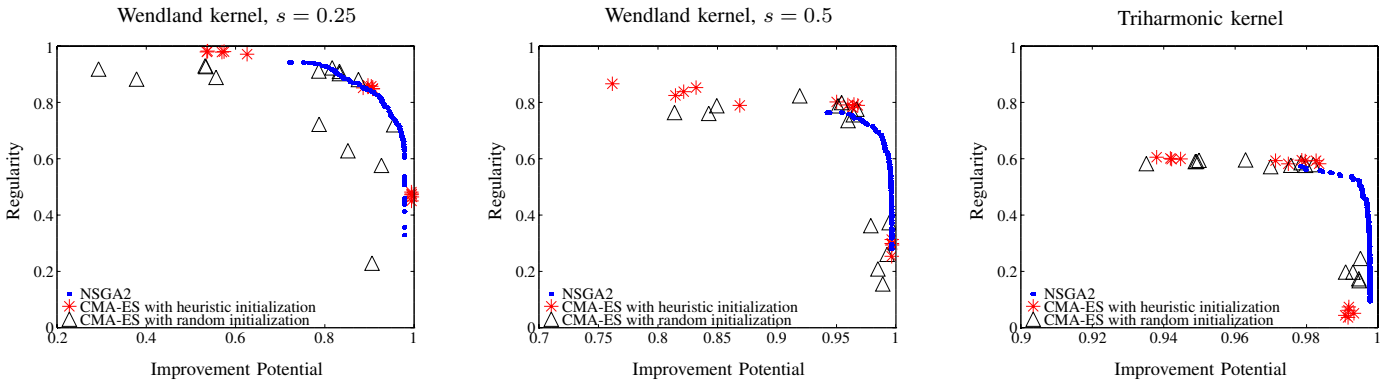

Fig. 5. Comparison of heuristic (red stars) and random (black triangles) initialization for the single-objective optimization of the kernel distribution, and the solutions computed through multi-objective optimization from [4]. The single-objective optimization out-performs the two-objective optimization in most tests, and heuristic initialization leads to improved setups on average.

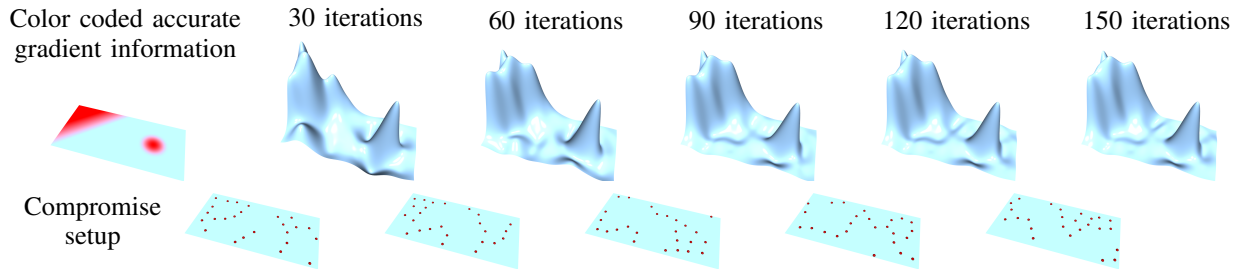

Fig. 6. Stepwise design optimization initialized with accurate gradient information in the beginning. A compromise preference between improvement potential and regularity is chosen for setup adaptation, leading to good fitting quality and convergence speed.

types: a compact Wendland kernel with support radii 0.25 and 0.5, and a global triharmonic kernel [20]. The improvement potential for the initial center distribution was derived from the exact analytic gradient.

In Figure 4 we show convergence plots of the fitness of the deformation setup $\mathbf{U}$, either being solely improvement potential $f_0(\mathbf{U}) = P(\mathbf{U})$, a compromise between improvement potential and regularity $f_{0.5}(\mathbf{U}) = 0.5 \cdot P(\mathbf{U}) + 0.5 \cdot R(\mathbf{U})$, or solely regularity $f_1(\mathbf{U}) = R(\mathbf{U})$. The fitness values are averaged over five trials. For each kernel the heuristic initialization (solid lines in Figure 4) out-performs random initializations (dashed lines in Figure 4). 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 NSGAII (Figure 5, blue dots) reveals that a weighted single-objective optimization with heuristic initialization (red stars) shows a very high performance compared to the two-objective optimization, except for the values of improvement potential for the triharmonic kernel, which are slightly worse (Figure 5 right). Moreover, heuristic initialization leads to better setups than to random initialization for the Wendland kernels and results in setups of similar quality for the triharmonic kernel. These results and the largely superior performance of the single-objective optimization confirm the benefit of our proposed OLS/Lloyd initialization for setup adaptation.

B. *Stepwise Design Optimization: 1D Function Approximation*

In this section we analyze the stepwise design optimization for the height field approximation scenario, where we again evaluate three different preference weights ($\lambda = 0$, 0.5, and 1) for the adaptation phase. The design optimization process alternates between setup adaptation (Section V) and optimization of the fit (Section VI). For the former we distribute 25 RBF kernels with two coordinates with a (7,15)-CMA-ES, for the latter 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 adaptation five times, which we denote as one test (see Figure 6). Each test took approximately 40 minutes on an Intel Xeon, 8×3.60 GHz, with 8 GB of memory.

To reduce the random effects of the setup adaptation and the fitting optimization we perform five tests for each of the three kernel functions (Wendland kernel with support radius 0.25 and 0.5, and a 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 6). 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 we sketch such an estimation compared to the accurate gradient direction.

In the case of accurate gradient information we expect that a preference $\lambda = 0$, i.e., a focus on exploitation, results in the

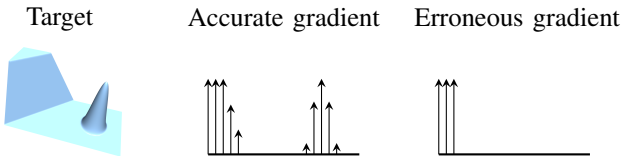

Fig. 7. Sketch of an accurate gradient direction derived from the target and an erroneous gradient which completely ignores the peak in the target.

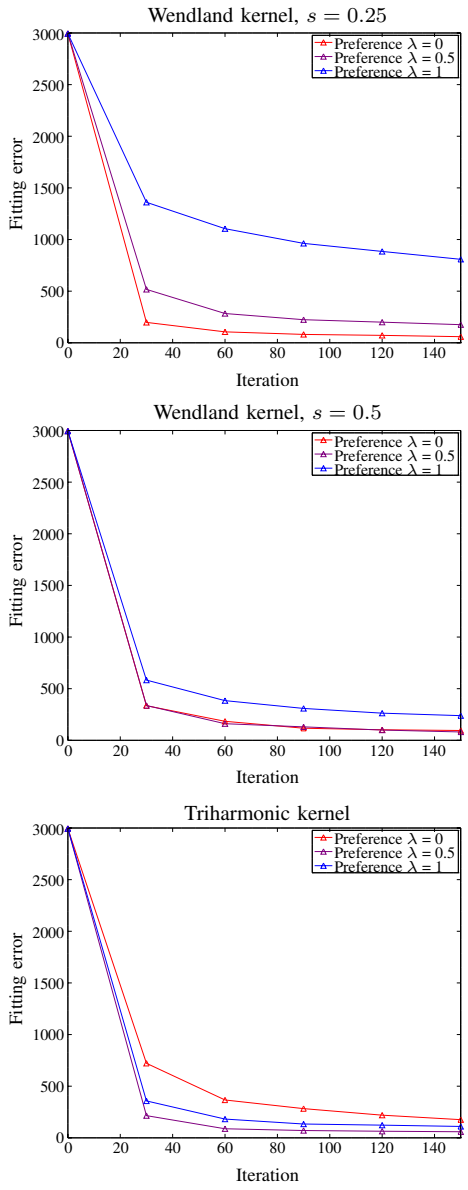

Fig. 8. Convergence plot for the adaptive optimization procedure for 1D function approximation initialized with accurate gradient information. Choosing an intermediate preference (purple line) to set up the deformation result in good fitting quality for Wendland kernels (top, center) and out-performs a pure focus on exploitation for triharmonic kernels (bottom).

TABLE I. 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

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 8, top). For the larger Wendland kernel, the intermediate preference is on par with a preference on improvement potential (Figure 8, center), and for triharmonic kernels it even out-performs this preference (Figure 8, bottom). As shown in Table I, 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 9 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 9 (top) the centers are only placed near the plateau during the whole design optimization process. Choosing a compromise between improvement potential and regularity 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 9, bottom.

In Figure 10 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

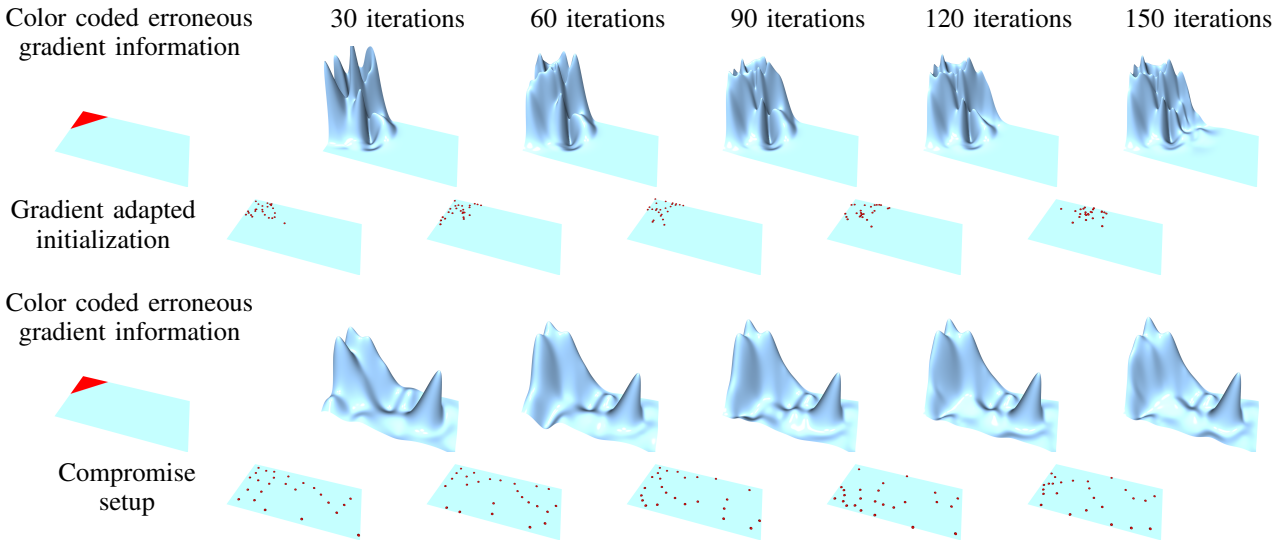

Fig. 9. Design optimization with an erroneous estimation of the initial gradient given by a designer (left). Top: Adapting purely to improvement potential "exploits" the misleading erroneous gradient information, resulting in fits of low quality. Bottom: A compromise between improvement potential and regularity manages to repair the initially misleading gradients.

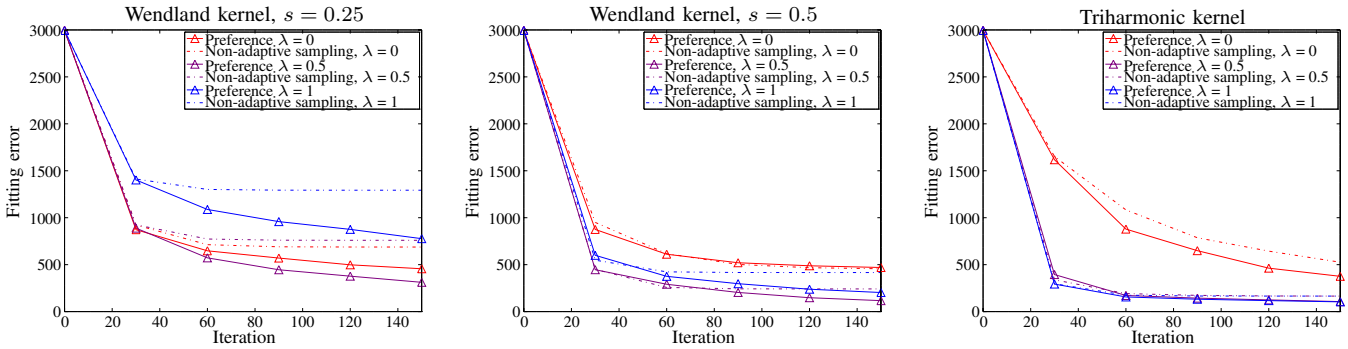

Fig. 10. Convergence plot for the adaptive optimization procedure for 1D function approximation initialized with an estimated gradient given by a designer. An adaptive process (solid lines) performs better than a non-adaptive process (dashed lines). Choosing an intermediate preference (solid purple line) handles the coarse estimation of the gradient and out-performs a pure preference focus on improvement potential (solid red line) or regularity (solid blue line).

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 10, dashed lines), which demonstrates its benefit for design optimization.

C. Stepwise Design Optimization: 3D Template Fitting

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 in [3, 4]). The goal in this design optimization scenario is to fit a template (sphere) to a scanned face (Figure 2). Each vertex of the design has three degrees of freedom and exact gradient information is not available (see [3] for details). In 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 in the adaptation phase. 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 adaptation 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 11. 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 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

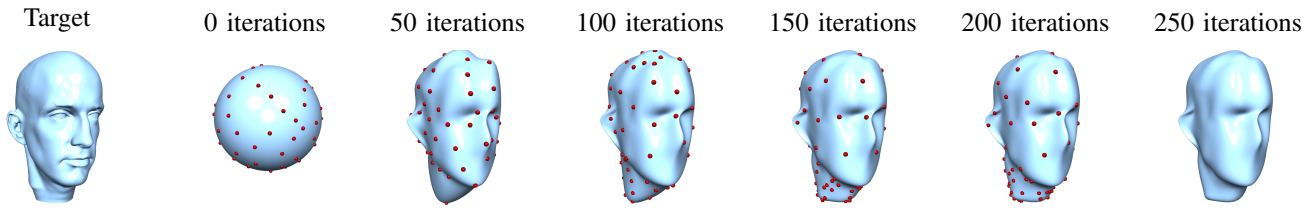

Fig. 11. Example results of an adaptive template fitting process where the RBF kernels are adjusted on the deformed mesh with a compromise preference between regularity and improvement potential, i.e., between exploration and exploitation.

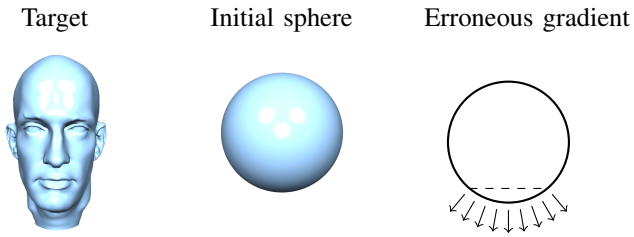

Fig. 12. As erroneous gradient information for template fitting we use displacements just on the bottom of sphere (right) to initialize the adaptation of the kernel distribution.

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 12). The convergence plots (Figure 13) 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 14), thereby leading to an improved design.

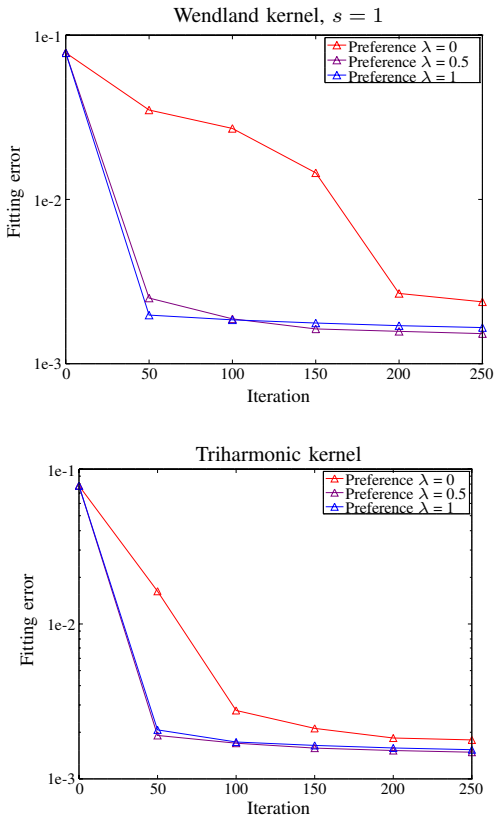

Fig. 13. Convergence plot for the adaptive optimization procedure of the 3D template fitting. Choosing an intermediate preference (purple line) to setup the deformation out-performs a preference of 0 or 1.

VIII. Summary and Future Work

The adaptation strategy of a representation is crucial for the performance of an evolutionary optimization process. The concept of evolvability reveals powerful quality criteria for setups, especially regularity and improvement potential, to be utilized in an adaptation process. Regularity and improvement potential are conflicting targets, which therefore have to be carefully weighted.

First, we analyzed heuristics to initialize the adaptation (optimization) process of a deformation representation. We successfully combined the Lloyd sampling for regular setups and the OLS sampling for setups with high improvement potential in order to construct initial deformation setups according to a given preference weight. With this initialization we increase the computational performance and the resulting quality of an evolutionary adaptation procedure.

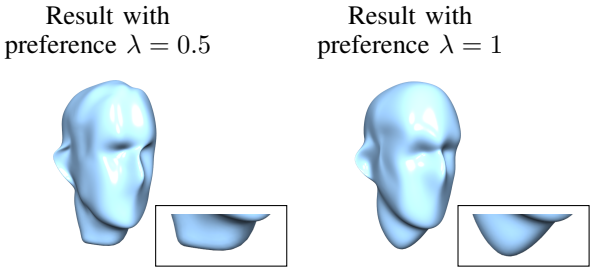

Fig. 14. Using an intermediate preference (left) between regularity and improvement potential to adapt setups is robust to erroneous initial gradient information and utilizes extracted information of the design optimization phase to improve the fit in an important region. In contrast, adaptation purely focusing regularity results in a plausible fit but neglects the extracted information (right).

Second, we integrated this procedure into a stepwise design optimization process. We showed in 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 adapt the 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).

In this article we performed all tests in a static environment and used a fixed preference for the whole stepwise design optimization process. Our next goal is to realize a flexible adjustment of the preference during a dynamic design optimization process, first in our test scenarios and finally in an automotive design optimization.

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] A. A. Mina, D. Braha, and Y. Bar-Yam, *Complex Engineered Systems: Science Meets Technology*. Springer, 2006, ch. Complex Engineered Systems: A New Paradigm, pp. 1–21.

[2] A. Richter, M. Botsch, and S. Menzel, "Evolvability of representations in complex system engineering: a survey," in *Proceedings of IEEE Congress on Evolutionary Computation*, 2015, pp. 1327–1335.

[3] A. Richter, J. Achenbach, S. Menzel, and M. Botsch, "Evolvability as a quality criterion for linear deformation representations in evolutionary optimization," in *Proceedings of IEEE Congress on Evolutionary Computation*, 2016, pp. 901–910.

[4] A. Richter, J. Achenbach, S. Menzel, and M. Botsch, "Multi-objective representation setups for deformation-based design optimization," in *Proceedings of 9th International Conference on Evolutionary Multi-Criterion Optimization*, 2017, p. to appear.

[5] S. Chen, S. A. Billings, and W. Luo, "Orthogonal least squares methods and their application to non-linear system identification," *International Journal of control*, vol. 50, no. 5, pp. 1873–1896, 1989.

[6] J. B. Gomm and D. L. Yu, "Selecting radial basis function network centers with recursive orthogonal least squares training," *IEEE Transactions on Neural networks*, vol. 11, no. 2, pp. 306–314, 2000.

[7] S. P. Lloyd, "Least squares quantization in PCM," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, 1982.

[8] D. Sieger, S. Menzel, and M. Botsch, "A comprehensive comparison of shape deformation methods in evolutionary design optimization," in *Proceedings of the International Conference on Engineering Optimization*, 2012.

[9] E. Costa, M. E. Biancolini, C. Groth, U. Cella, G. Veble, and M. Andrejasic, "RBF-based aerodynamic optimization of an industrial glider," in *Proceedings of International CAE Conference*, 2014.

[10] A.-V. Vuong, C. Giannelli, B. Jüttler, and B. Simeon, "A hierarchical approach to adaptive local refinement in isogeometric analysis," *Computer Methods in Applied Mechanics and Engineering*, vol. 200, no. 49, pp. 3554–3567, 2011.

[11] J. Zheng, Y. Wang, and H. S. 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*, 2005, pp. 405–411.

[12] Y. Ohtake, A. Belyaev, and H.-P. Seidel, "3D scattered data approximation with adaptive compactly supported radial basis functions," in *Proceedings of IEEE International Conference on Shape Modeling Applications*, 2004, pp. 31–39.

[13] M. Olhofer, Y. Jin, and B. Sendhoff, "Adaptive encoding for aerodynamic shape optimization using evolution strategies," in *Proceedings of IEEE Congress on Evolutionary Computation*, 2001, pp. 576–583.

[14] Z. Yang, B. Sendhoff, K. Tang, and X. Yao, "Target shape design optimization by evolving B-splines with cooperative coevolution," *Applied Soft Computing*, vol. 48, pp. 672–682, 2016.

[15] L. F. Simões, D. Izzo, E. Haasdijk, and A. E. Eiben, "Self-adaptive genotype-phenotype maps: neural networks as a meta-representation," in *International Conference on Parallel Problem Solving from Nature*, 2014, pp. 110–119.

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

[17] N. Aulig, E. Nutwell, S. Menzel, and D. Detwiler, "Preference-based topology optimization of body-in-white structures for crash and static loads," 2016, 14th International LS-DYNA Users Conference.

[18] W. M. Hsu, J. F. Hughes, and H. Kaufman, "Direct manipulation of free-form deformations," in *Proceedings of ACM SIGGRAPH*, 1992, pp. 177–184.

[19] S. Menzel, M. Olhofer, and B. Sendhoff, "Direct manipulation of free form deformation in evolutionary design optimisation," in *Proceedings of the International Conference on Parallel Problem Solving From Nature*, 2006, pp. 352–361.

[20] H. Wendland, *Scattered data approximation*. Cambridge University Press, 2004.

[21] G. P. Wagner and L. Altenberg, "Perspectives: Complex adaptations and the evolution of evolvability," *Evolution*, vol. 50, no. 3, pp. 967–976, 1996.

[22] G. H. Golub and C. F. Van Loan, *Matrix computations*. Johns Hopkins University Press, 2012.

[23] A. Auger and N. Hansen, "Turorial: CMA-ES: Evolution strategies and covariance matrix adaptation," in *Proceedings of the Conference on Genetic and Evolutionary Computation*, 2011, pp. 991–1010.

[24] T. Weise, R. Chiong, and K. Tang, "Evolutionary optimization: Pitfalls and booby traps," *Journal of Computer Science and Technology*, vol. 27, no. 5, pp. 907–936, 2012.

[25] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, 1967, pp. 281–297.

[26] Y. Eldar, M. Lindenbaum, M. Porat, and Y. Y. Zeevi, "The farthest point strategy for progressive image sampling," *IEEE Transactions on Image Processing*, vol. 6, no. 9, pp. 1305–1315, 1997.

[27] L. Graening, S. Menzel, M. Hasenjäger, T. Bihrer, M. Olhofer, and B. Sendhoff, "Knowledge extraction from aerodynamic design data and its application to 3d turbine blade geometries," *Journal of Mathematical Modelling and Algorithms*, vol. 7, no. 4, pp. 329–344, 2008.

[28] C. Igel, V. Heidrich-Meisner, and T. Glasmachers, "Shark," *The Journal of Machine Learning Research*, vol. 9, pp. 993–996, 2008.