High Quality Mesh Morphing Using Triharmonic Radial Basis Functions

Daniel Sieger[1], Stefan Menzel[2], and Mario Botsch[1]

[1] Bielefeld University, Germany
{dsieger,botsch}@techfak.uni-bielefeld.de
[2] Honda Research Institute Europe, Offenbach, Germany
stefan.menzel@honda-ri.de

Summary. The adaptation of an existing volumetric simulation mesh to updated parameters of the underlying CAD geometry is a crucial component within automatic design optimization. By avoiding costly automatic or even manual (re-)meshing it enables the automatic generation and evaluation of new design variations, e.g., through FEM or CFD simulations. This is particularly important for stochastic global optimization techniques—such as evolutionary algorithms—which typically require a large number of design variations to be created and evaluated. In this paper we present a simple yet versatile method for high quality mesh morphing. Building upon triharmonic radial basis functions, our shape deformations minimize distortion and thereby implicitly preserve shape quality. Moreover, the same unified code can be used to morph tetrahedral, hexahedral, or arbitrary polyhedral volume meshes. We compare our method to several other recently proposed techniques and show that ours yields superior results in most cases.

Keywords: mesh morphing, mesh warping, design optimization

1 Introduction

Simulation-based fully automatic design optimization is becoming a cornerstone of the product development process of the automotive industry, aircraft construction, and naval architecture. A key component to the successful and efficient application of such an optimization process is the ability to adapt an existing volumetric simulation mesh according to an updated CAD geometry. The importance of such an automatic component is increased further when dealing with complex geometries that prohibit automatic mesh generation and require manual interaction by an expert instead, and/or when using stochastic global optimization techniques that require a large number of design variations to be created and evaluated in order to find the optimum.

The adaptation of an existing simulation mesh is addressed by mesh morphing or mesh warping: Given an initial CAD surface $\mathcal{G}$ and a volumetric mesh $\mathcal{M}$ of that geometry, a shape variation $\mathcal{G} \mapsto \mathcal{G}'$ is generated by modifying the geometric embedding of $\mathcal{G}$ while keeping its topology fixed. Mesh morphing then adapts the mesh $\mathcal{M}$ such that the updated version $\mathcal{M}'$ conforms to the updated boundary surface $\mathcal{G}'$. Analogously to the CAD modification, only the geometric embedding of $\mathcal{M}$ (i.e., its node positions) is modified in this process, while the mesh topology (i.e., its connectivity) stays fixed.

Mesh morphing techniques aim at preserving the element quality as much as possible, thereby allowing for as large as possible geometric changes before inevitably requiring some remeshing due to element inversion. Staten and coworkers recently proposed and evaluated several mesh morphing techniques, which they compared with respect to computational performance and element quality on different tetrahedral and hexahedral meshes [26].

Motivated by the work of Staten et al. [26], building on their results, and contributing to their benchmarking comparisons, we present and evaluate a meshless morphing technique based on triharmonic radial basis functions (RBFs). Our method yields highly smooth space warps that minimize distortion and thereby preserve element quality. While being computationally more expensive, our method offers the following compelling advantages: It is easy to understand and straightforward to implement; it can be applied to tetrahedral, hexahedral, or general polyhedral meshes; and finally, it more robustly achieves higher quality results.

2 Related Work

The recent comparison of mesh morphing methods published by Staten and colleagues [26] constitutes the starting point for our investigation. Based on a set of test scenarios involving varying complexity and topology the authors benchmark several techniques for warping volume meshes. Besides in the meshing community, mesh morphing or mesh deformation methods have also been a subject of intensive research in computer graphics. Within both fields the different morphing approaches can roughly be classified into four categories: methods based on generalized barycentric coordinates, mesh smoothing techniques, mesh-based variational methods that minimize certain fairness energies, and meshless warping approaches. Most techniques assume the updated positions of boundary nodes to be given and compute the new locations of interior nodes from these boundary constraints.

Approaches based on *barycentric coordinates* determine the interior nodes as a linear (affine or convex) combination of the boundary nodes through a generalization of linear barycentric interpolation [28]. Examples are Wachspress coordinates [29], mean value coordinates [8], harmonic coordinates [14], and maximum entropy coordinates [27, 12]. The simplex-linear method of [26],

being a generalization of BMSWEEP [25], as well as its extension to natural neighbor interpolation [24], can also be associated to this category. While these approaches typically have simple geometric constructions and therefore are easy to implement and efficient to compute, the resulting morphs might not be smooth enough to reliably preserve element quality.

Mesh smoothing methods adjust interior node locations in order to explicitly optimize the quality of mesh elements [15, 23, 16], where the Mesquite framework [6] offers implementations based on mean ratio, untangling, and matrix condition number [15]. In the context of mesh warping the updated boundary nodes act as fixed constraints while the interior nodes are determined by the optimization process. The mesh smoothing methods evaluated in [26] worked well for small geometric changes, but were lacking robustness for larger scale modifications. In comparison, the LBWARP method [23], a weighted Laplacian smoothing based on the log-barrier technique, gives considerably better results, but is computationally more complex.

Mesh-based variational methods compute smooth harmonic or biharmonic deformations by solving Laplacian or bi-Laplacian systems [2, 11], which is numerically more robust than most mesh smoothing techniques. The finite element-based FEMWARP technique [2], which computes a harmonic deformation, was generalized from tetrahedra to hexahedra in [26], and turned out to be the most successful approach in Staten's benchmarks. Note that harmonic coordinates [14] (see above) are closely related to these approaches, since they are also derived by solving a Laplacian system. The boundary nodes' harmonic coordinate functions can be thought of as "response functions" of the Laplacian PDE. While the morphs produced by mesh-based variational methods tend to preserve element quality very well, they have to be custom-tailored to each mesh type (e.g., tetrahedral or hexahedral).

In contrast, *meshless morphing techniques* avoid this limitation by computing a space warp $\mathbf{d}\colon \mathbb{R}^3 \to \mathbb{R}^3$ that deforms the whole embedding space, thereby implicitly deforming each node of the mesh $\mathcal{M}$. After the initial freeform deformation (FFD) paper [22], many variants and extensions have been proposed. We refer the reader to the survey papers [3, 20, 10], which focus on mathematical formalisms for the different methods, on the interactive manipulation by a designer, and on shape deformation in the context of aerodynamic design optimization, respectively. However, spline-based FFD does not offer the same degree of smoothness as harmonic or biharmonic morphs, and it requires a rather tedious lattice setup.

We propose to combine the advantages of meshless approaches and mesh-based variational methods by using *triharmonic radial basis functions* (RBFs) for mesh morphing [4, 5, 13, 18]. Our RBF space warps are easy to setup and compute, can handle arbitrary polyhedral meshes, and offer a degree of smoothness equivalent or even superior to mesh-based variational techniques. Moreover, we show that the same concept can not only be employed for morphing the volumetric simulation mesh (Section 3), but also for morphing the updated surface node locations (Section 4).

3 RBF Volume Morphing

The input of the volume morphing are the *surface nodes* $\{\mathbf{s}_1, \ldots, \mathbf{s}_m\}$ and interior *volume nodes* $\{\mathbf{v}_1, \ldots, \mathbf{v}_n\}$ of the initial mesh $\mathcal{M}$, as well as the deformed *surface nodes* $\{\mathbf{s}'_1, \ldots, \mathbf{s}'_m\}$ of $\mathcal{M}'$, where the $\mathbf{s}_i$ and $\mathbf{s}'_i$ conform to the CAD geometries $\mathcal{G}$ and $\mathcal{G}'$, respectively. The goal is to find updated *volume node* positions $\{\mathbf{v}'_1, \ldots, \mathbf{v}'_n\}$, such that the element quality of the morphed mesh $\mathcal{M}'$ is as good as possible.

Employing a space deformation approach, we treat the volume morphing as an abstract scattered data interpolation problem. We are looking for a deformation function $\mathbf{d}\colon \mathbb{R}^3 \to \mathbb{R}^3$ that (i) exactly interpolates the prescribed boundary displacements $\mathbf{d}(\mathbf{s}_i) = (\mathbf{s}'_i - \mathbf{s}_i)$ and (ii) smoothly interpolates these displacements into the mesh interior. Radial basis functions (RBFs) are well known to be suitable for solving this type of problem [30]. An RBF deformation is represented as a linear combination of radially symmetric kernels $\varphi_j(\mathbf{x}) = \varphi(\|\mathbf{x}_j - \mathbf{x}\|)$, located at centers $\mathbf{x}_j \in \mathbb{R}^3$ and weighted by $\mathbf{w}_j \in \mathbb{R}^3$, plus a linear polynomial that guarantees linear precision:

$$\mathbf{d}(\mathbf{x}) \;=\; \sum_{j=1}^{m} \mathbf{w}_j \varphi_j(\mathbf{x}) \;+\; \sum_{k=1}^{4} \mathbf{q}_k \pi_k(\mathbf{x})\,, \tag{1}$$

where $\{\pi_1, \pi_2, \pi_3, \pi_4\} = \{x, y, z, 1\}$ is a basis of the space of linear trivariate polynomials, weighted by coefficients $\mathbf{q}_k \in \mathbb{R}^3$. Note that the polynomial term is important, since it guarantees to find the optimal affine motion (translation, rotation, scaling) contained in the prescribed displacements $\mathbf{s}_i \mapsto \mathbf{s}'_i$.

The choice of the kernel function $\varphi\colon \mathbb{R} \to \mathbb{R}$ has a significant influence on the quality of the deformation. We propose to choose $\varphi(r) = r^3$, which by construction results in a triharmonic deformation function (i.e., $\Delta^3 \mathbf{d} = \mathbf{0}$), since the RBF kernels φ_j are fundamental solutions of the tri-Laplacian equation. As a consequence, the RBF deformation minimizes the fairness energy [30]

$$\iiint_{\mathbb{R}^3} \left\| \frac{\partial^3 \mathbf{d}}{\partial x^3} \right\|^2 + \left\| \frac{\partial^3 \mathbf{d}}{\partial x^2 \partial y} \right\|^2 + \ldots + \left\| \frac{\partial^3 \mathbf{d}}{\partial z^3} \right\|^2 \mathrm{d}x\,\mathrm{d}y\,\mathrm{d}z. \tag{2}$$

Because of this our RBF deformations are even slightly more smooth than the mesh-based harmonic or biharmonic morphs mentioned above. Compared to a biharmonic RBF warp ($\varphi(r) = r$) the triharmonic morph indeed turned out to provide a better element quality preservation in our experiments.

Satisfying the interpolation constraints $\mathbf{d}(\mathbf{s}_i) = (\mathbf{s}'_i - \mathbf{s}_i)$ amounts to placing RBF kernels at the constraint positions (i.e., $\mathbf{x}_j = \mathbf{s}_j$) and finding the coefficients $\mathbf{w}_j$ and $\mathbf{q}_k$ by solving the $(m+4) \times (m+4)$ linear system

$$\begin{pmatrix} \varphi_1(\mathbf{s}_1) & \cdots & \varphi_m(\mathbf{s}_1) & \pi_1(\mathbf{s}_1) & \cdots & \pi_4(\mathbf{s}_1) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \varphi_1(\mathbf{s}_m) & \cdots & \varphi_m(\mathbf{s}_m) & \pi_1(\mathbf{s}_m) & \cdots & \pi_4(\mathbf{s}_m) \\ \pi_1(\mathbf{s}_1) & \cdots & \pi_1(\mathbf{s}_m) & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \pi_4(\mathbf{s}_1) & \cdots & \pi_4(\mathbf{s}_m) & 0 & \cdots & 0 \end{pmatrix} \begin{pmatrix} \mathbf{w}_1^T \\ \vdots \\ \mathbf{w}_m^T \\ \mathbf{q}_1^T \\ \vdots \\ \mathbf{q}_4^T \end{pmatrix} = \begin{pmatrix} (\mathbf{s}'_1 - \mathbf{s}_1)^T \\ \vdots \\ (\mathbf{s}'_m - \mathbf{s}_m)^T \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{pmatrix}. \quad (3)$$

After solving this linear system we can compute the morphed mesh $\mathcal{M}'$ by simply evaluating the RBF deformation at each volume node: $\mathbf{v}'_i = \mathbf{v}_i + \mathbf{d}(\mathbf{v}_i)$. This part can easily be parallelized and therefore is highly efficient. The computationally most expensive part is the solution of the linear system (3), which is dense due to the global support of $\varphi(r)$. While there are advanced techniques for efficiently solving this kind of systems, such as multipole expansion, multilevel approximation, or greedy center selection schemes [7, 30, 18], this was not necessary in all our test cases. Since most CAD geometries $\mathcal{G}$ and their corresponding volume meshes $\mathcal{M}$ are constructed from several solid components, we can simply perform the volume morphing individually for each of these (reasonably small) components. In all our examples this could be done using a standard dense linear system solver from the LAPACK library [1].

4 Surface Morphing

In order to establish a common baseline for comparison the benchmark tests of Staten and colleagues [26] were all based on the same surface morphing, i.e., they all shared the same boundary surface nodes for $\mathcal{M}$ and $\mathcal{M}'$. However, while the quality of our RBF volume morphing alone is already promising, the whole volume morph is completely determined by (and hence its quality is bounded by) the surface morph, i.e., the surface node positions $\mathbf{s}'_i \in \mathcal{M}'$. In order to further improve the quality of our results we propose a high quality surface morphing method that is also based on triharmonic RBFs.

Since the topology of the CAD surface $\mathcal{G}$ is assumed to stay constant, there is a one-to-one correspondence between the faces, curves, and corner vertices of $\mathcal{G}$ and $\mathcal{G}'$. Staten and colleagues exploit this fact for morphing curves: For each *curve node* $\mathbf{c}_i \in \mathcal{M}$, which is associated to a curve $\mathbf{f}\colon \mathbb{R} \to \mathbb{R}^3$ of the initial geometry $\mathcal{G}$, they find the parameter value u such that $\mathbf{c}_i = \mathbf{f}(u)$ and compute the morphed node as $\mathbf{c}'_i = \mathbf{f}'(u)$, where $\mathbf{f}' \subset \mathcal{G}'$ is the morphed curve corresponding to $\mathbf{f} \subset \mathcal{G}$. The morphed curves then act as boundary constraints for morphing the surface nodes, which was performed using either mesh smoothing or the weighted residual technique [26]. Both, however, lead to a certain amount of distortion or even inverted surfaces triangles, which in turn negatively impacts the volume morphing.

In our approach we extend the curve morphing idea to the surface case: For each surface node $\mathbf{s}_i$ we find its corresponding face $\mathbf{f}\colon \mathbb{R}^2 \to \mathbb{R}^3$ and its (u, v)-parameters, and define the morphed surface node as the corresponding point on the morphed face $\mathbf{f}'(u, v)$. We first describe how to find the (u, v)-parameters of a surface node $\mathbf{s}_i$, before explaining the actual mapping from $\mathbf{f}$ to $\mathbf{f}'$.

Given a surface node $\mathbf{s}_i$, finding its corresponding face $\mathbf{f}$ and (u, v) parameters in theory amount to projecting $\mathbf{s}_i$ onto each face $\mathbf{f}_k \in \mathcal{G}$ and selecting the closest one. Although this functionality is offered by most CAD kernels (OpenCASCADE [19] in our case), in practice these projections are both computationally intensive and numerically instable for complex, trimmed faces. We address both problems by densely sampling the CAD surface $\mathcal{G}$, which requires only robust and efficient evaluations and results in samples $\mathbf{x}_j = \mathbf{f}_k(u_j, v_j)$. For each surface node $\mathbf{s}_i$ we then find its closest sample point $\mathbf{x}_j$ and project $\mathbf{s}_i$ onto $\mathbf{f}_k$ with (u_j, v_j) as initial guess. When storing the samples $(\mathbf{x}_j, u_j, v_j, k)$ in a kD-tree [21], finding the closest sample for a given $\mathbf{s}_i$ takes less than 0.01ms for a very dense sampling of about 15M sample points. This approach significantly improved the efficiency and robustness of the projections.

After finding the face $\mathbf{f}$ and the (u, v) parameters, the node $\mathbf{s}_i$ has to be moved to the corresponding point on the morphed CAD face $\mathbf{f}'$. This part is more challenging than for the curve case, since the geometric embedding of a face $\mathbf{f}\colon \Omega \to \mathbb{R}^3$ can be changed in two ways: (i) by modifying its geometric parameters, e.g., spline control points or cylinder radii, and (ii) by changing its parameter domain Ω, e.g., by adjusting its trimming curves. While (i) simply amounts to evaluating $\mathbf{f}'$ instead of $\mathbf{f}$, (ii) requires to morph the parameter values $(u, v) \in \Omega$ to $(u', v') \in \Omega'$.

In order to morph the parameter values to the updated parametric domain we exploit the versatility of our approach and construct a 2D triharmonic RBF deformation function $\mathbf{d}_k\colon \mathbb{R}^2 \to \mathbb{R}^2$ for each face $\mathbf{f}_k$ of the CAD model. To this end we uniformly sample the (u, v)-boundary curves of the faces $\mathbf{f}_k$ and $\mathbf{f}'_k$, resulting in 2D point samples $\{\mathbf{c}_1, \dots, \mathbf{c}_n\} \in \Omega$ and $\{\mathbf{c}'_1, \dots, \mathbf{c}'_n\} \in \Omega'$. Compared to the 3D case (1), computing the 2D triharmonic warp requires only minor changes: The 2D triharmonic kernel is $\varphi(r) = r^2 \ln(r)$, the polynomial part consists of the basis $\{\pi_1, \pi_2, \pi_3\} = \{x, y, 1\}$, and the coefficients $\mathbf{w}_j, \mathbf{q}_k$ are two-dimensional. With these changes, and replacing $\mathbf{s}_i$ by $\mathbf{c}_i$, a linear system analogous to (3) is solved for the RBF warp $\mathbf{d}_k$. After performing the parameter warp $(u', v') = (u, v) + \mathbf{d}_k(u, v)$ the morphed surface node can be computed as $\mathbf{s}'_i = \mathbf{f}'_k(u', v')$.

This method is easy to compute and produces high quality surface warps of minimal parametric distortion. Thanks to its meshless nature, it can be applied to all kinds of faces, such as simple non-trimmed rectangular faces, trimmed faces with curved boundaries, as well as faces with trimmed holes (see Bore and Pipe examples in the next section).

5 Results

In this section we compare our mesh morphing technique to the results recently published in [26]. The examples include meshes of varying topology, including structured and unstructured hexahedral as well as tetrahedral meshes. The complexity of the models ranges from ∼10-15k vertices for the Bore and Pipe models up to ∼130k vertices for the Courier model. The Canister model used in [26] was not available to us. A detailed description of the different geometric parameters for each model and how they are modified is given in [26]. We restrict our comparison to those methods that either delivered the best results (FEMWARP and LBWARP), or that were recommended by the authors for sake of simplicity and efficiency (Simplex-linear). In order to ensure comparability of the results we also measure element quality based on the scaled Jacobian as described in [15]. For our RBF volume morphing, we include results for both the original surface node locations from [26] (denoted RBF) as well as those obtained by our surface morphing (denoted RBF-S). Following [26] we investigate two different types of morphing—*relative* and *absolute* morphing. In the former case the mesh is updated incrementally from the initial design until the full parameter change is reached. In the latter case the initial mesh is directly warped to the corresponding parameter change. In the following subsections we present detailed results for the individual test cases. Selected cut-views of morphed volume meshes highlighting regions being particularly sensitive to degenerated elements are shown in Figure 8.

5.1 Bore Model

The change of parameters in the Bore model tests the ability of the different methods to deal with scaling and rotation. An example morph from the initial mesh to the full parameter change is shown in Figure 1. As can be seen from Figure 2, our method is on par with or better than the FEMWARP and LBWARP methods. We note that the element inversion after 75% parameter change in case of the tetrahedral model is due to a defect in the morphed surface mesh of [26]. By using our more robust surface morph we are able to perform both the relative and the absolute morph up to a parameter change of 100% without any inverted elements.

5.2 Pipe Model

The change of parameters in the Pipe model tests the ability of the different methods to deal with nonlinear stretching. The initial and final shapes are illustrated in Figure 4. The detailed results are given in Figure 5. While our method provides superior results on the hex model, those on the tetrahedral model are comparable. However, in contrast to other methods ours does not result in inverted elements at 95% parameter change for the absolute morphing of the tetrahedral model. Again, the results obtained using our combined volume and surface morphing are superior. The difference in surface mesh element quality is also illustrated in Figure 3.

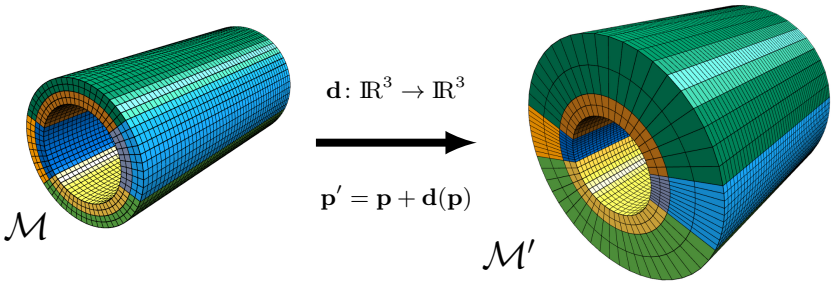

Fig. 1. Volume mesh morphing of the hexahedral Bore model using RBFs.

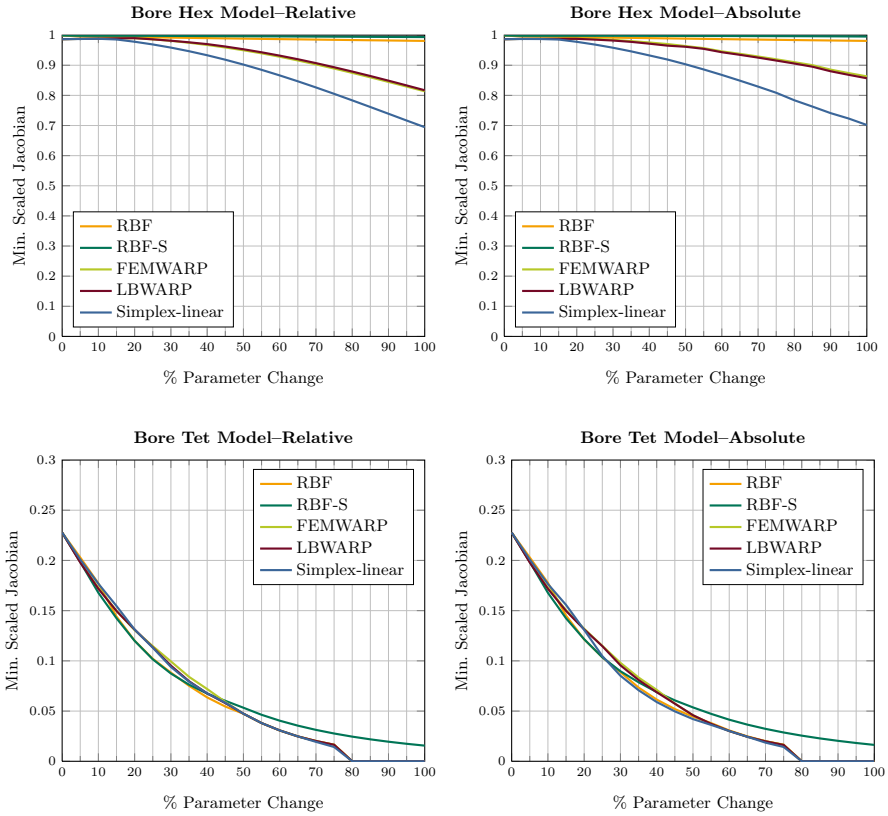

Fig. 2. Morphing results of the Bore model.

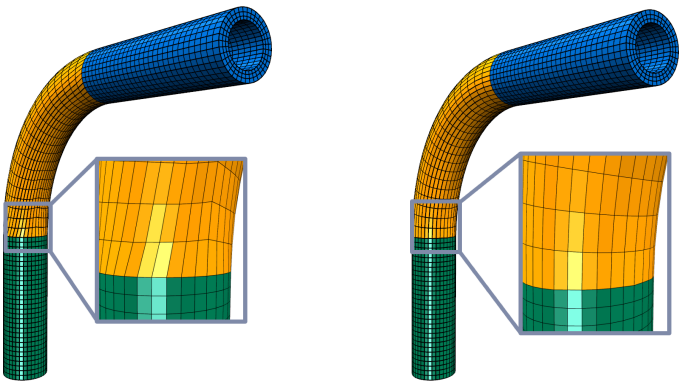

Fig. 3. Comparison of element quality in the morphed surface mesh. While there are distortions in the meshes of Staten et al. [26] (left), our surface (right) is perfectly aligned to the updated CAD geometry.

5.3 Courier Model

The Courier model is the most complex model in our comparison. In contrast to previous examples the hexahedral mesh of this model is an unstructured one. Especially in case of the absolute tetrahedral mesh morphing all methods presented in [26] result in inverted elements as soon as a change of parameter values of 65% is reached. In contrast, our method results in inverted elements only after a parameter change of 75%. Unfortunately, due to a mismatch between CAD geometry and initial tetrahedral mesh we could not apply our surface morphing method for the tetrahedral Courier model.

5.4 Performance

The execution time in seconds for a single morphing step is given in Table 1. Since in our method the performance does not differ between absolute and relative morphing we only report timings for absolute morphing. In general our method is computationally more expensive than those investigated by Staten et al. However, in all but one case our method allows to perform an absolute morph to the full parameter change without resulting in inverted elements, while other methods might only reach this goal by falling back to several steps of relative morphing, thereby becoming computationally more expensive. We also note that the timing for the tetrahedral Courier model is due to the fact that the model—unlike the hexahedral one—is only decomposed into four large components. Finally, we note that in our current implementation we morph each component in a serial manner. Exploiting the parallel nature of the problem would further increase the performance of our method.

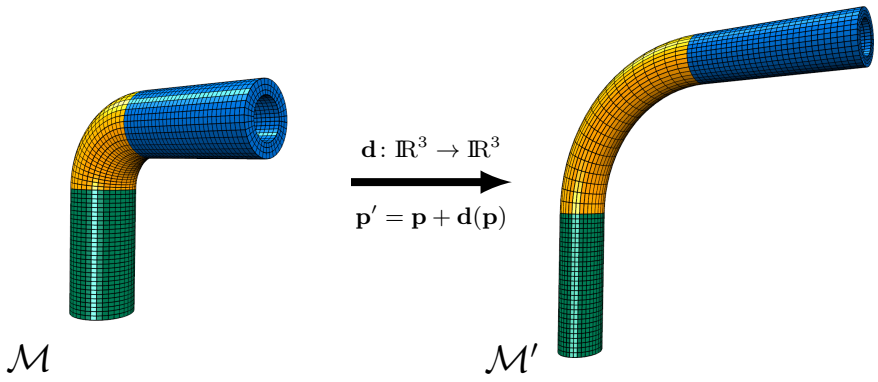

Fig. 4. Volume mesh morphing of the hexahedral Pipe model using RBFs.

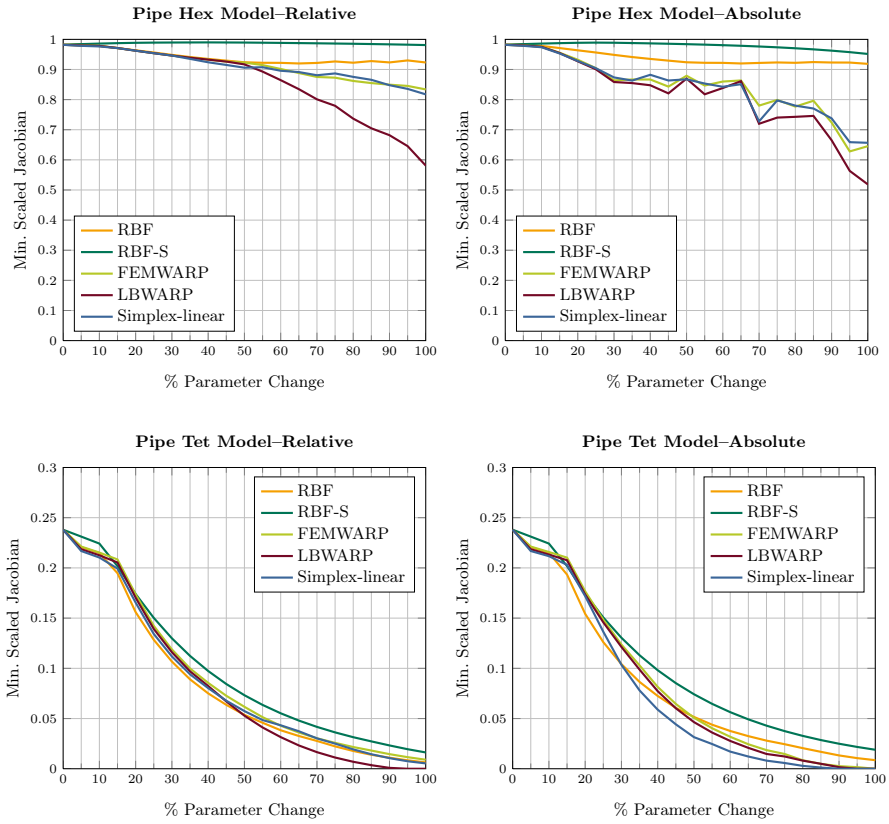

Fig. 5. Morphing results of the Pipe model.

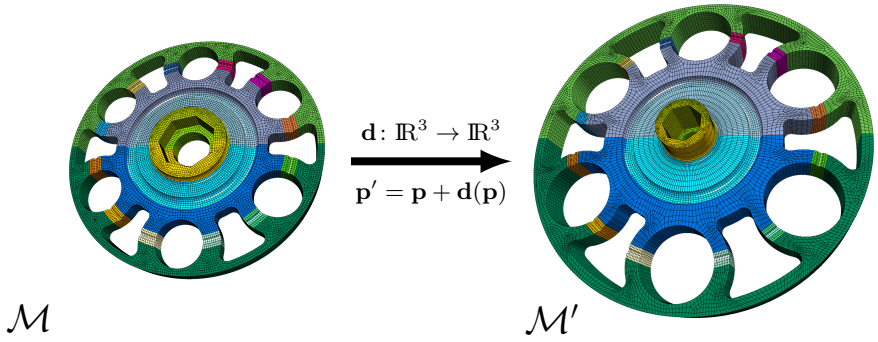

Fig. 6. Volume mesh morphing of the hexahedral Courier model using RBFs.

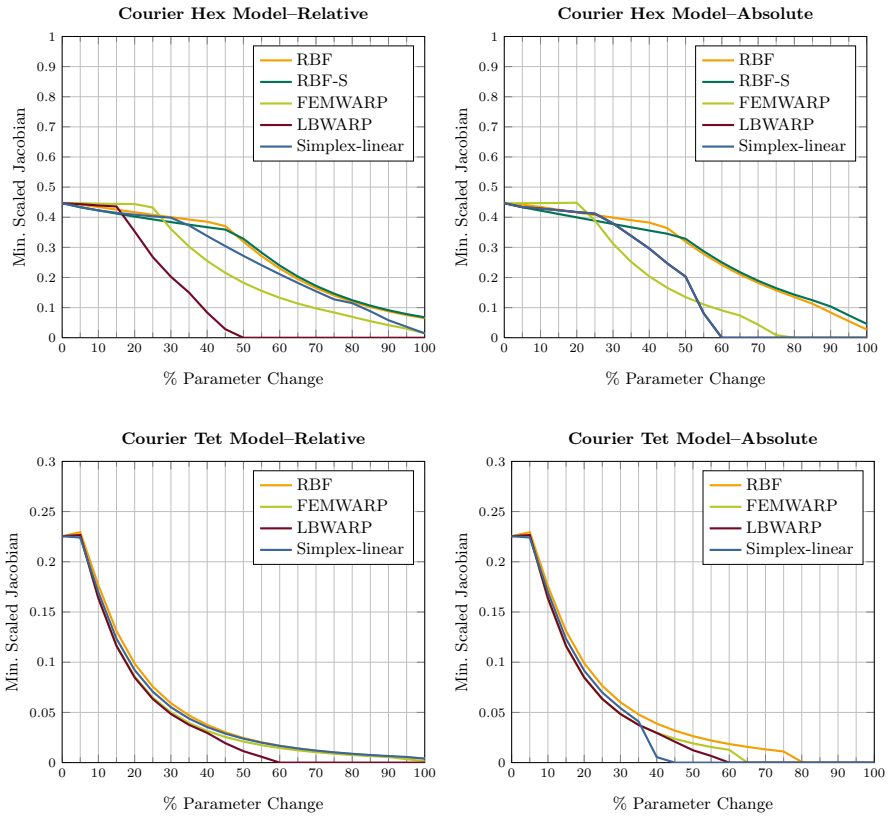

Fig. 7. Morphing results of the Courier model.

Bore Hex	Bore Tet	Pipe Hex	Pipe Tet	Courier Hex	Courier Tet
5.483	8.078	3.094	5.690	9.3397	32.7698

Table 1. Single step performance in seconds.

6 Conclusion and Future Work

In this paper we presented a simple and versatile method for high-quality mesh morphing of both surface and volume meshes using RBFs. The fairness of our triharmonic RBF morphs leads to similar or superior element quality compared to all techniques evaluated in [26]. The implementation of our method is straightforward and essentially requires setting up the linear system (3) and solving it using a standard solver. Therefore, it can be considered significantly easier to implement than, e.g., the LBWARP and FEMWARP approaches employed in [26]. Furthermore, our RBF morphs are more flexible, since they can be applied to tetrahedral, hexahedral, or even arbitrary polyhedral meshes.

As has already been observed in [26], using relative morphing can be used to prevent inverted elements in the resulting mesh. Similar approaches have been used in the graphics community to perform deformations that are free of self-intersections [9, 17]. These approaches typically rely on the integration of a smooth space-time vector field, which by construction guarantees absence of self-intersections and element inversions. However, for most practical applications a full integration is not necessary and successive splitting of the deformation can be used to prevent inversions. An interesting challenge and direction for future work is the question when and how to split the deformation, e.g., uniformly across the whole deformation or only when an actual inversion occurs.

Acknowledgments

The authors kindly thank Matthew Staten from Sandia National Laboratories for providing us with the Bore, Pipe, and Courier models from his last year's paper. Daniel Sieger is supported by Honda Research Institute Europe (HRI-EU). Mario Botsch is supported by the German National Research Foundation (DFG CoE 277: CITEC).

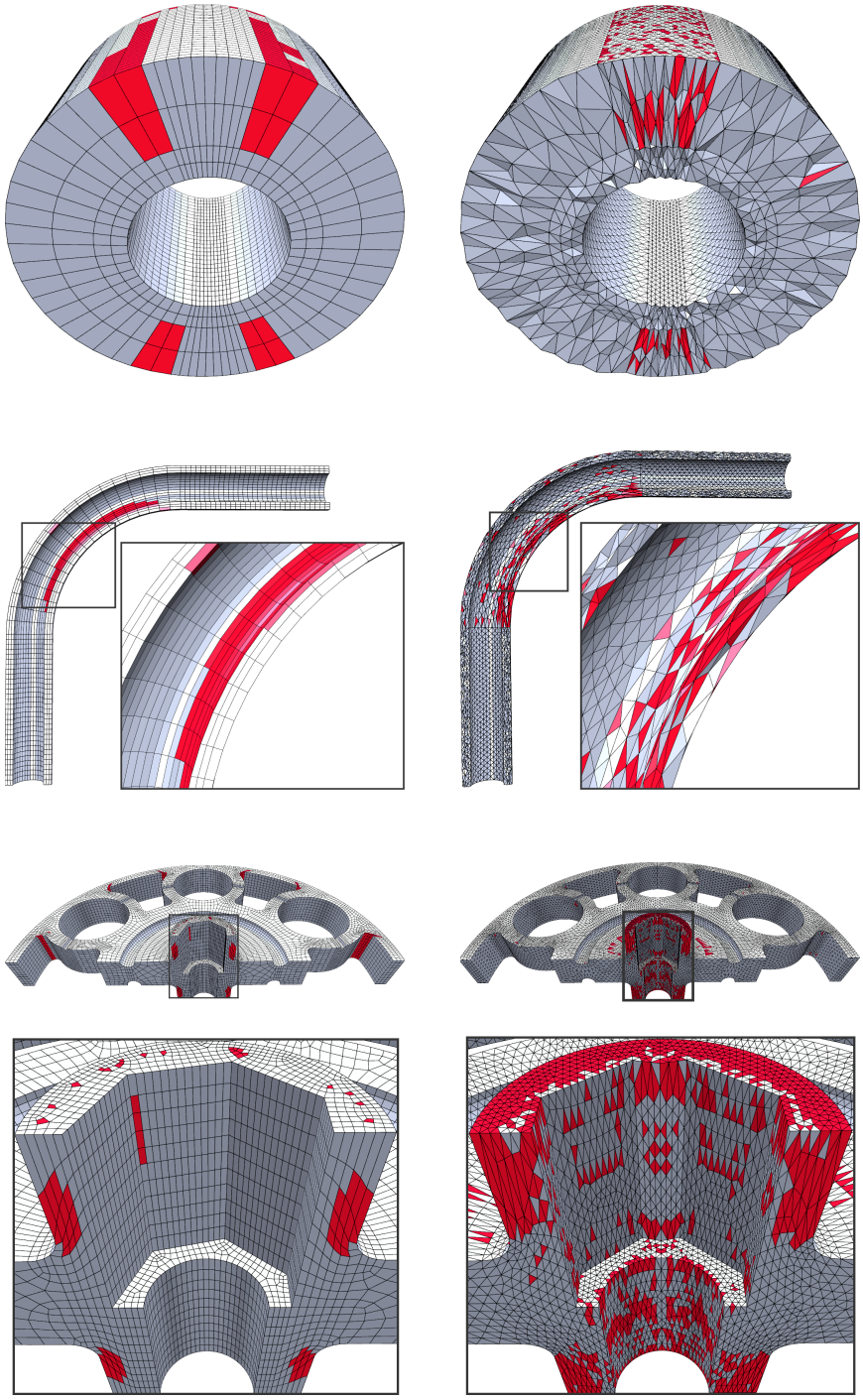

Fig. 8. Cut views for the Bore, Pipe, and Courier models after performing an absolute morph to the full parameter change. The worst 5% of the elements are highlighted in red.

References

1. Anderson, E., Bai, Z., Bischof, C., Blackford, S., Demmel, J., Dongarra, J., Croz, J.D., Greenbaum, A., Hammarling, S., McKenney, A., Sorensen, D.: LAPACK Users' Guide, third edn. Society for Industrial and Applied Mathematics, Philadelphia, PA (1999)
2. Baker, T.J.: Mesh movement and metamorphosis. In: Proceedings of the 10th International Meshing Roundtable, pp. 387–396 (2001)
3. Bechmann, D.: Space deformation models survey. Computers & Graphics **18**(4), 571 – 586 (1994)
4. de Boer, A., van der Schoot, M., Bijl, H.: Mesh deformation based on radial basis function interpolation. Computers & Structures **85**, 784–795 (2007)
5. Botsch, M., Kobbelt, L.: Real-time shape editing using radial basis functions. Computer Graphics Forum (Proc. Eurographics) **24**(3), 611–621 (2005)
6. Brewer, M., Diachin, L.F., Knupp, P., Leurent, T., Melander, D.: The Mesquite mesh quality improvement toolkit. In: Proceedings of the 12th International Meshing Roundtable, pp. 239–250 (2003)
7. Carr, J.C., Beatson, R.K., Cherrie, J.B., Mitchell, T.J., Fright, W.R., McCallum, B.C., Evans, T.R.: Reconstruction and representation of 3D objects with radial basis functions. In: Proc. of ACM SIGGRAPH, pp. 67–76. ACM, New York (2001)
8. Floater, M.S., Kos, G., Reimers, M.: Mean value coordinates in 3D. Computer Aided Geometric Design **22**, 623–631 (2005)
9. von Funck, W., Theisel, H., Seidel, H.P.: Vector field-based shape deformations. ACM Transactions on Graphics (Proc. SIGGRAPH) **25**(3), 1118–1125 (2006)
10. Gain, J., Bechmann, D.: A survey of spatial deformation from a user-centered perspective. ACM Transaction on Graphics **27**, 107:1–107:21 (2008)
11. Helenbrook, B.T.: Mesh deformation using the biharmonic operator. International Journal for Numerical Methods in Engineering **56**, 1007–1021 (2003)
12. Hormann, K., Sukumar, N.: Maximum entropy coordinates for arbitrary polytopes. Computer Graphics Forum (Proc. Symp. Geometry Processing) **27**(5), 1513–1520 (2008)
13. Jakobsson, S., Amoignon, O.: Mesh deformation using radial basis functions for gradient-based aerodynamic shape optimization. Computers & Fluids **36**(6), 1119–1136 (2007)
14. Joshi, P., Meyer, M., DeRose, T., Green, B., Sanocki, T.: Harmonic coordinates for character articulation. ACM Transactions on Graphics (Proc. SIGGRAPH) **26**(3) (2007)
15. Knupp, P.: Achieving finite element mesh quality via optimization of the Jacobian matrix norm and associated quantities. Part I. International Journal for Numerical Methods in Engineering **48**(3), 401–420 (2000)
16. Knupp, P.: Updating meshes on deforming domains: An application of the target-matrix paradigm. Commun. Num. Meth. Engr. **24**(6), 467–476 (2007)
17. Martinez Esturo, J., Rössl, C., Fröhlich, S., Botsch, M., Theisel, H.: Pose correction by space-time integration. In: Proc. of Vision, Modeling, Visualization, pp. 33–40 (2011)
18. Michler, A.K.: Aircraft control surface deflection using RBF-based mesh deformation. International Journal for Numerical Methods in Engineering **88**(10), 986–1007 (2011)

19. OpenCASCADE: Open CASCADE Technology, 3D modeling & numerical simulation (2012). URL `http://www.opencascade.org/`
20. Samareh, J.A.: A survey of shape parameterization techniques. Tech. Rep. NASA/CP-1999-209136/PT1, NASA Langley Research Center (1999)
21. Samet, H.: The Design and Analysis of Spatial Data Structures. Addison Wesley, Reading, MA (1994)
22. Sederberg, T.W., Parry, S.R.: Free-form deformation of solid geometric models. In: Proc. of ACM SIGGRAPH, pp. 151–159. ACM, New York (1986)
23. Shontz, S.M., Vavasis, S.A.: A mesh warping algorithm based on weighted Laplacian smoothing. In: Proceedings of the 12th International Meshing Roundtable, pp. 147–158 (2003)
24. Sibson, R.: Interpreting Multivariate Data, vol. 21, chap. A brief description of natural neighbor interpolation. John Wiley & Sons (1981)
25. Staten, M.L., Canann, S.A., Owen, S.J.: BMSWEEP: Locating interior nodes during sweeping. Eng. Comput. **15**(3), 212–218 (1999)
26. Staten, M.L., Owen, S.J., Shontz, S.M., Salinger, A.G., Coffey, T.S.: A comparison of mesh morphing methods for 3D shape optimization. In: Proceedings of the 20th International Meshing Roundtable, pp. 293–311 (2011)
27. Sukumar, N.: Construction of polygonal interpolants: A maximum entropy approach. International Journal for Numerical Methods in Engineering **61**(12), 2159–2181 (2004)
28. Sukumar, N., Malsch, E.A.: Recent advances in the construction of polygonal finite element interpolants. Archives of Computational Methods in Engineering **13**(1), 129–163 (2006)
29. Wachspress, E.L.: A Rational Finite Element Basis. Academic Press (1975)
30. Wendland, H.: Scattered Data Approximation. Cambridge University Press, Cambridge, UK (2005)