A 2D Topology-Adaptive Mesh Deformation Framework for Mesh Warping

Jibum Kim and David McLaurin and Suzanne M. Shontz

Abstract We propose a framework for performing anisotropic mesh deformations. Our goal is to produce high quality meshes with no inverted elements on domains which undergo large deformations. To the greatest extent possible, the meshes should have similar element shape; however, topological changes are performed as necessary in order to improve mesh quality. Our framework is based upon the previous work of two of the authors and their collaborators [1, 2] and consists of four steps. The first step is to perform anisotropic finite element-based mesh warping to estimate the interior vertex positions based upon an appropriate choice of the PDE coefficients. The second step is to perform multiobjective mesh optimization in order to eliminate inverted elements and improve element shape. Edge swaps are then performed to further improve the mesh quality. A final mesh smoothing pass is then performed. Our numerical results show that our framework can be used to generate high quality meshes with no inverted elements for very large deformations. In particular, the addition of topological changes to our hybrid mesh deformation algorithm [1] proved to be an extremely efficient way of improving the mesh quality.

1 Introduction

There are numerous scientific applications for which the geometric domains deform as a function of time. Such applications include simulations of Arbitrary-Lagrangian-Eulerian (ALE) fluid flow [3, 4], fluid-structure interaction [5, 6], ALE

Jibum Kim
Incheon National University, Incheon, South Korea, e-mail: jibumkim@incheon.ac.kr

David McLaurin
CD-adapco, Austin, TX 78750 U.S.A. e-mail: david.mclaurin@cd-adapco.com

Suzanne M. Shontz
Mississippi State University, Mississippi State, MS 39762 U.S.A. e-mail: sshontz@math.msstate.edu

plasticity [7], crack propagation [8], biomedical applications [9, 10], patient and medical devices [11, 12, 13], and computer graphics [14]. Whenever such deformations occur, the meshes must be updated with respect to time in order to remain valid approximations of the geometry. There are two main types of strategies for updating the mesh in response to a deforming geometric domain. The first strategy is to remesh the domain whenever necessary in response to the deforming domain. This typically creates a different mesh, and the associated numerical partial differential equation (PDE) solution must be interpolated from the initial mesh to the new mesh since their topology is different. Another issue with this approach is that frequent remeshing can lead to loss of data resolution and accumulation of round-off errors leading to inaccuracies [15]. Alternatively, a mesh deformation (i.e., mesh warping or mesh morphing) strategy can be used to move the mesh from the source domain onto the target domain; such techniques recompute interior mesh vertex positions after the boundary has been deformed. Mesh warping is preferred over remeshing based on accuracy (as described above) as well as efficiency.

There are numerous mesh warping techniques in the literature. We give a summary of relevant mesh warping techniques here; however, this list is not comprehensive. Mesh warping techniques are typically based on the solution of a PDE which describes the motion of the interior mesh vertices or on the solution of an optimization problem which guides the mesh motion based on desired properties. Researchers have proposed various mesh warping techniques [16] based on Laplace's equation, e.g., finite element-based mesh warping (FEMWARP) [17, 18], weighted Laplacian smoothing [19], biharmonic PDEs [3], elasticity [20, 21, 22], and an inverse distance function [23]. Researchers have also proposed techniques which combine the solution of PDEs with techniques for altering the mesh topology [24, 25, 26, 27] in order to yield high quality deformed meshes in simulations with large deformations.

Researchers have also proposed several optimization-based mesh deformation techniques. For example, techniques have been developed based on a log-barrier techniques [19] and the target matrix paradigm [28]. Many of the mesh deformation techniques which are guided by optimization also involve PDE solutions. For example, the nonlinear elasticity-based Untangling before Newton (UBN) method [22] mentioned above can also be thought of as an optimization-based mesh warping technique in that it solves a variational problem (which is equivalent to a minimization problem) to achieve static equilibrium. An adjoint-based optimization procedure for mesh warping was developed in [29] in order to improve the robustness and extend the range of linear elasticity-based mesh warping techniques. FEMWARP has also been combined with PDE-based level set techniques [11, 12], the purpose of which is to first predict the mesh deformation using an evolving level set and then FEMWARP to deform the mesh to the location computed by the level set method. Another example of a hybrid mesh deformation algorithm which employs both optimization and PDEs is found in [1]. This particular hybrid algorithm employs an anisotropic version of FEMWARP for the mesh deformation followed by multiobjective mesh optimization [2] for smoothing and untangling of the deformed mesh.

All of the methods described thus far, with the exception of the hybrid mesh deformation algorithm in [1], suffer from one or more of the following problems: a tendency to produce inverted elements for large boundary deformations, or an inability to preserve features of the initial mesh in the deformed mesh.

In the current work, we consider larger boundary deformations than those in [1]. In particular, the hybrid mesh deformation method in [1] is unable to produce high quality deformed meshes for these test cases.

In this paper, we propose a topology-adaptive mesh deformation framework for mesh warping. In particular, we combine the hybrid mesh deformation algorithm of Kim, Miller, and Shontz in [1] with topological changes in order to generate high quality meshes for extremely large boundary deformations. We describe our topology-adaptive mesh deformation framework in Section 2. In Section 3, we describe several numerical experiments in which we test the ability of our algorithm to generate high quality deformed meshes on several extremely large boundary deformations. The numerical results from the first two experiments can be compared against those from the same experiments in [1] in order to obtain a comparison against the method by Kim, Miller, and Shontz. In Section 4, we summarize our work and describe several possibilities for future work.

2 Algorithmic Framework

In this section, we describe our topology-adaptive mesh deformation framework for use in mesh warping applications. Our mesh deformation framework is composed of four steps: 1) anisotropic FEMWARP, 2) multiobjective mesh optimization, 3) topological changes, and 4) mesh smoothing. Our algorithmic framework builds upon the hybrid mesh deformation algorithm [1] of Kim, Miller, and Shontz, which performs only step 1 (anisotropic FEMWARP) and step 2 (multiobjective mesh optimization), and the multiobjective mesh optimization framework [2] of Kim, Panitanarak, and Shontz.

Initially, the deformation that the user provides is applied to the boundary vertices. This prescribes the final positions of the boundary vertices in the deformed mesh. We then perform the four steps in our topology-adaptive mesh deformation framework in order to compute the positions of the interior vertices.

The first step in our framework is to compute initial, approximate locations for the interior vertices in the deformed mesh by performing one step of anisotropic finite element-based mesh warping (i.e., anisotropic FEMWARP) with an appropriate choice of PDE coefficients as proposed in [1]. In particular, anisotropic FEMWARP solves the following anisotropic version of Laplace's equation

$$-\alpha\frac{\partial^2 u}{\partial x^2}-\beta\frac{\partial^2 u}{\partial y^2}=0 \text{ on } \Omega, \tag{1}$$

where $u=u_0$ on $\partial\Omega$. We assume that $\alpha>0$ and $\beta>0$ and solve the PDE using an elliptic PDE solver. The mesh topology is held fixed during anisotropic FEMWARP.

We adaptively choose α and β with respect to the direction of the boundary deformation. In [1], we demonstrated that α and β control the strength of the x- and y-axis couplings between adjacent vertices, respectively. Therefore, α should be larger than β if more deformation occurs along the x-axis. Similarly, β should be larger than α if more deformation occurs along the y-axis. We compute the cumulative boundary vertex displacements in the x and y directions to decide in which direction more deformation occurs. Let x_k and y_k be the k^{th} vertex coordinates on the initial mesh and $\hat{x_k}$ and $\hat{y_k}$ be the k^{th} vertex coordinates on the deformed mesh. A particular choice of the anisotropic PDE coefficients that worked well for our experiments in [1] was

$$\begin{cases} \alpha = \sum_{k=1}^{N_B} (\hat{x_k} - x_k), \ k \in B \\ \beta = \sum_{k=1}^{N_B} (\hat{y_k} - y_k), \ k \in B, \end{cases} \tag{2}$$

where B is the set of boundary vertices and N_B is the number of boundary vertices. Here, the relative ratio between α and β can be understood as the angle of the direction of the deformation [1].

Our second step is the multiobjective mesh optimization. The deformed mesh using anisotropic FEMWARP could include inverted elements with poor element qualities on the deformed domain when a huge deformation occurs. Similar to the hybrid deformation algorithm [1], we employ the target matrix paradigm (TMP) shape metric to improve the element quality on the deformed domain. The TMP shape metric is useful when our goal is to preserve good element qualities (shapes) on the deformed domain. Let $(A_{\text{def}})_i$ and $(A_{\text{init}})_i$ be the Jacobians of the mappings from the reference element to the actual elements in the deformed and initial domains, respectively. The TMP shape metric in 2D is defined as

$$q_i = \left| T_i - (\text{adj}(T_i)^T) \right|_F^2,$$

where $T_i = (A_{\text{def}})_i (A_{\text{init}})_i^{-1}$. The TMP shape metric ($q_i$) is zero when the quality of the element on the deformed mesh is same as the one on the initial mesh. In order to eliminate inverted elements on the deformed domain, we employ the untangling beta quality metric [2]. The untangling beta quality metric is defined as

$$q_j = \left| V_j - \beta \right| - (V_j - \beta),$$

where V_j is the signed area of the j^{th} element and β is a user defined small constant value. The untangling beta quality metric is zero when the deformed mesh does not have any inverted elements.

Let the overall mesh quality computed by the TMP shape metric and the overall mesh quality computed by the untangling beta quality metric be F_1 and F_2, respectively. Then, $F_1 = \sum_{i=1}^{|E|} q_i^2$ and $F_2 = \sum_{j=1}^{|E|} q_j^2$, where $|E|$ is the number of elements on the mesh.

In order to simultaneously untangle inverted elements and smooth the deformed mesh, we employ the exponential sum multiobjective mesh optimization method in [2]. The exponential sum multiobjective mesh optimization method utilizes the

min-max property, which minimizes the worst (maximum) objective function using the exponential penalty function. The exponential sum multiobjective mesh optimization function is denoted as

$$F = c\ ln\left[e^{F_1/c} + e^{F_2/c}\right].$$

Our goal is to minimize F to find the optimal vertex positions on the deformed domain. Similar to [2], we employ the Fletcher-Reeves nonlinear conjugate gradient method to find a locally optimal point. The combination of the first two steps described above is the hybrid mesh deformation algorithm in [1]. Similar to other mesh untangling algorithms [2, 28], there is no guarantee that our hybrid mesh deformation algorithm in [1] is able to always untangle the deformed mesh.

Once the mesh is untangled, we perform edge swaps (step 3) as indicated in [30] in order to further improve the quality of the mesh. Even if we eliminate all inverted elements after step 2, the output after step 2 still suffer from poor element qualities when a huge deformation occurs. From [30],

> "For an edge uv with opposite vertices p and q, we flip uv if the Delaunay flipping criterion (i.e., $\angle upv + \angle uqv > \pi$) is satisfied... We used a greedy strategy to flip edges in decreasing order of maximum opposite angle...".

This work differs from [30] in the manner in which infinite loops cause by repeated edge flipping are avoided. There, a constraint was added to the algorithm to only allow a given edge to be flipped once. Here, a suitably small tolerance, tol (a scalar multiple of the run-time-calculated round-off error), is used as a "buffer" for angle comparison. This changes the edge-flip criteria to: $(\angle upv + \angle uqv > (\pi + tol))$ This buffer lessens the likelihood of numerical errors causing infinite loops. Since an infinite is loop is still possible (but improbable), a limit of 100 flips per edge was enforced—but never encountered in practice. This developed strategy of limiting edge-flips produced results that were more favorable for this application than those in [30]. Additionally, any edge flips that would create invalid topology (cannot flip boundary edges) or geometry (inverted/tangled triangles) were not allowed. GRX5 [31], a topology repair and feature removal toolkit and library, was used to make the topological changes. The data structures and algorithms are optimized for performing topological operations on triangular surface grids. Currently it is available as a stand-alone tool and is also incorporated into SolidMesh [32].

The final step in our mesh deformation framework is to perform a final pass of mesh smoothing according to the inverse mean ratio (IMR) metric in order to obtain further improvement in the mesh quality. Let the coordinates of the three vertices of a triangle denoted by a, b and c. Then, an incidence matrix, C is given by $[b-a, c-a]$. For an equilateral triangle, the incidence matrix for an ideal element, W, is denoted by

$$W = \begin{bmatrix} 1 & \frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} \end{bmatrix},$$

The IMR quality metric measures how similar the current element to the ideal element (equilateral element). The IMR quality metric is defined as

$$q_{IMR} = \frac{\left\|CW^{-1}\right\|_F^2}{2\left|\det(CW^{-1})\right|}.$$

The IMR quality metric has the value of 1 for the ideal element (equilateral triangle) and a smaller value indicates a better element quality. Similar to the TMP shape metric, we compute the overall mesh quality by computing, $F_{IMR} = \sum_{i=1}^{|E|} q_{IMR}^2$. We minimize F_{IMR} using the the Fletcher-Reeves nonlinear conjugate gradient method to find a locally optimal point. For our test meshes, roughly 10 to 30 nonlinear conjugate gradient iterations are needed to reach the locally optimal points. Numerical results show that our final mesh smoothing step is able to improve the average and worst mesh quality up to 52.2% and 96.2%, respectively for the test meshes.

3 Numerical Experiments

Table 1 summarizes a description of each step and language/software we used to perform each step. We use Mesquite [33] to perform step 2 and 4. We stop to perform step 2 and move to step 3 when we eliminates all inverted elements on the deformed mesh. We use GRX5 [31] to perform step 3. For step 3, GRX5 was used to swap interior edges that did not meet Delaunay criteria. The resultant mesh is a constrained Delaunay triangulation.

In our framework, we first perform multiobjective mesh optimization (step 2) and perform topological changes (step 3) as a next step. This is because preliminary numerical experiments show that the reverse order (first perform step 3 and second perform step 2) results in much slower multiobjective mesh optimization time compared with the proposed order. When we perform multiobjective mesh optimization, we use the TMP shape metric such that the initial element and the deformed element have the similar element shape. However, we noticed that initially performing edge swaps often result in output meshes which are less similar to the initial mesh and therefore time to perform multiobjective mesh optimization could be very slow.

We compare our mesh deformation framework with Knupp's mesh deformation algorithm [28]. Knupp's mesh deformation algorithm sets the initial mesh to be a reference mesh and deforms the given mesh to be similar to the reference mesh based on a target matrix paradigm framework. We use Mesquite [33] to implement Knupp's mesh deformation algorithm.

The machine employed for this study is equipped with an AMD Opteron processor 6174 (2.2 GHz) and 6.5GB of RAM.

Table 1 Description of each step.

	Description of each step	Language/Software
Step 1	Perform anisotropic FEMWARP	C/C++
Step 2	Perform multiobjective mesh optimization	Mesquite [33]
Step 3	Perform topological changes	C/C++
Step 4	Perform mesh smoothing	Mesquite [33] (C++)

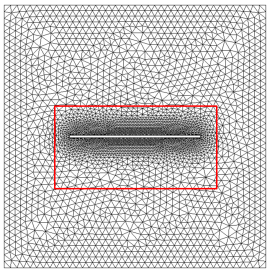
3.1 Moving bar domain for anisotropic boundary deformation along the y-axis

We consider a moving bar domain where the deformation occurs along the y-axis. Since the deformation only occurs along the y-axis, we chose the PDE coefficients, α=0 and β=1. Figure 1 shows the initial mesh and output meshes after each step. The initial mesh (Figure 1(a) and (b)) has no inverted elements and the output mesh after performing anisotropic FEMWARP (Figure 1(c) and (d)) has 84 inverted elements. Most of the inverted elements occur around the bar due to the large deformation. The output mesh after performing multiobjective mesh optimization is shown in Figure 1(e) and (f). After this step, all inverted elements are eliminated but many elements are still distorted due to the large deformation. The output mesh after performing topological changes is shown in Figure 2(a) and (b). Here we observe that mesh quality is improved—especially the poorest quality elements. The output mesh after performing final mesh smoothing is shown in Figure 2(c) and (d). Here we observe that the final output mesh has a good element quality and similar isotropy relative to the initial mesh.

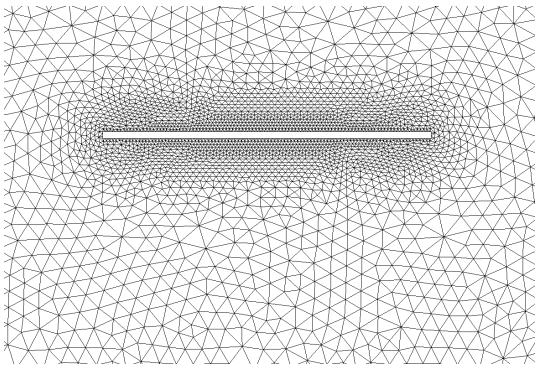
Table 2 shows mesh quality statistics and the number of inverted elements of the initial and output meshes after each step. The IMR quality metric was used to measure the element quality. Here, a smaller value indicates a better element quality. The final output mesh is exhibits similar element quality to the initial mesh (near isotropy). Knupp's mesh deformation algorithm [28] results in an output mesh with 27 inverted elements and poor element qualities. Knupp's mesh deformation algorithm does not include the mesh untangling step and only tries to keep similar element qualities. Therefore, we observe that Knupp's mesh deformation algorithm fails to produce an output mesh with no inverted elements and good element qualities.

3.2 Moving cylinder domain for anisotropic boundary deformation along the x-axis

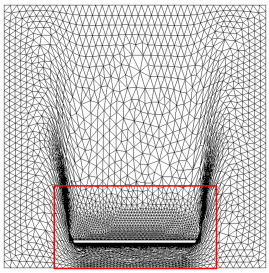
We consider a moving cylinder domain where the deformation occurs along the x-axis. We choose the PDE coefficients α=1 and β=0, since the deformation only



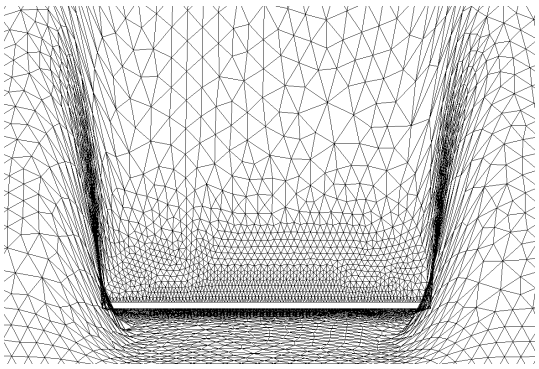
(a) Mesh on initial domain



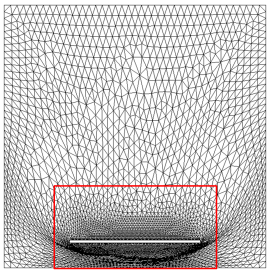
(b) Zoomed-in mesh on the initial domain



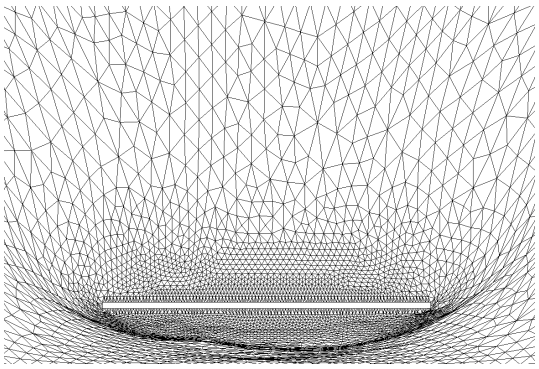
(c) Output mesh after step 1



(d) Zoomed-in output mesh after step 1

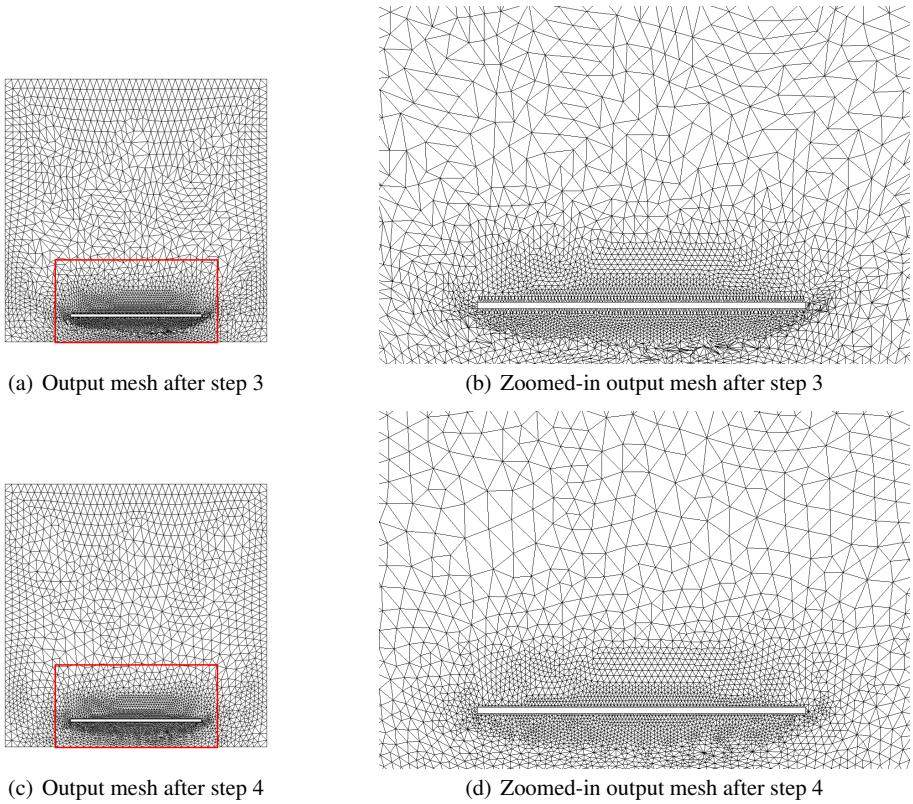


(e) Output mesh after step 2



(f) Zoomed-in output mesh after step 2

Fig. 1 Moving bar domain for anisotropic boundary deformation:(a) Initial mesh and (b) zoomed-in mesh on the bar domain. (c) Deformed mesh with anisotropic FEMWARP and (d) zoomed-in mesh with anisotropic FEMWARP. (e) Output mesh after performing multiobjective mesh optimization (f) zoomed-in mesh after performing multiobjective mesh optimization.



(a) Output mesh after step 3

(b) Zoomed-in output mesh after step 3

(c) Output mesh after step 4

(d) Zoomed-in output mesh after step 4

Fig. 2 Moving bar domain for anisotropic boundary deformation: (a) Output mesh after performing topological changes. (b) zoomed-in mesh after performing topological changes. (c) Output mesh after mesh smoothing. (d) zoomed-in mesh after mesh smoothing.

Table 2 Mesh quality statistics of moving bar domain measured by the inverse mean ratio quality metric and the number of inverted elements after each step. Here, 'each step' indicates the output mesh after each step.

Mesh quality	minimum	avg	rms	maximum	std.dev.	# of inverted elements
Initial	1	1.076	1.081	1.649	0.110	0
Step 1	1	11628.3	107803	1e+06	107174	84
Step 2	1	2.135	3.669	71.127	2.970	0
Step 3	1	1.268	1.348	10.682	0.456	0
Step 4	1	1.081	1.086	2.019	0.0969	0
Knupp [28]	-0.000471	2.478	33.826	2822.830	33.736	27

occurs along the x-axis. Figure 3 shows the initial mesh (Figure 3(a) and (b))and output meshes after performing each step. Overall results are similar to the previous moving bar example. Many poor quality and inverted elements (19 inverted elements) occur after the large deformation (Figure 3(c) and (d)). After performing multiobjective mesh optimization, we are able to eliminate inverted elements; but poor quality elements persist around the cylinder (Figure 3(e) and (f)). We observe that topological changes are effective in that they adjust the edges so that an increased amount of vertex movement can be performed as compared to before. For this example, topological changes reconnected the edges in the crowded area around the cylinder (Figure 4(a) and (b)). The final output mesh has a good element quality after performing mesh smoothing (Figure 4(c) and (d)).

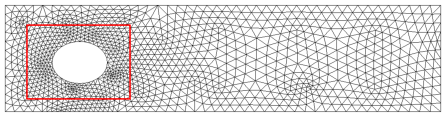
Table 3 shows mesh quality statistics and the number of inverted elements of the initial mesh and output meshes after performing each step. The output mesh, after performing multiobjective mesh optimization, still suffers from poor element quality. However, performing topological changes significantly improves the element quality. The worst element quality improves approximately 99% after performing topological changes. Note that the step 3 is significantly faster than the step 2 as we will discuss later. Similar to the previous moving bar example, the final output mesh has a similar element quality to the initial mesh. Similar to the previous example, Knupp's mesh deformation algorithm [28] fails to eliminate inverted elements and results in an output mesh with poor element qualities.

Table 3 Mesh quality statistics of moving cylinder domain measured by the inverse mean ratio quality metric and the number of inverted elements after each step. Here, 'each step' indicates the output mesh after each step.

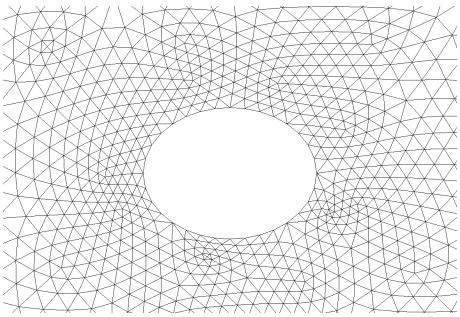
Mesh quality	minimum	avg	rms	maximum	std.dev.	# of inverted elements
Initial	1	1.040	1.040	1.940	0.0594	0
Step 1	1	8490.440	92098.6	1e+06	91706.4	19
Step 2	1	9.437	148.19	6979.570	147.889	0
Step 3	1	1.761	2.358	17.131	1.568	0
Step 4	1	1.159	1.171	1.948	0.170	0
Knupp [28]	-0.000441	15.843	7296.871	345349	7295.152	248

3.3 Moving particles domain for anisotropic boundary deformation along the x-axis

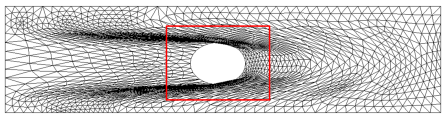
We consider a geometry that simulates moving particles—where several particles (cylinders) are moving in anisotropic ways. This example is more challenging than the previous moving cylinder example, since particles with different size are moving in different directions along the x-axis. The PDE coefficients are chosen as



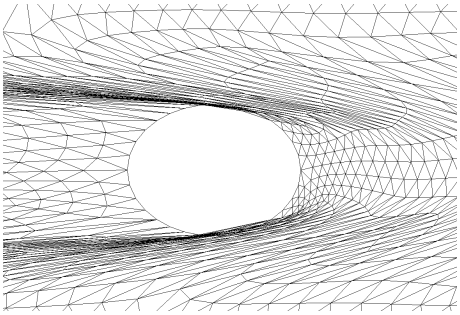
(a) Mesh on initial domain



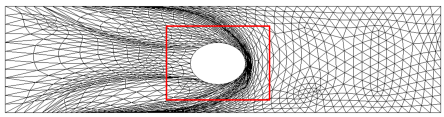
(b) Zoomed-in mesh on the initial domain



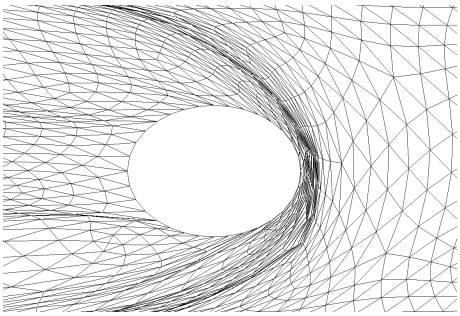
(c) Output mesh after step 1



(d) Zoomed-in output mesh on after step 1

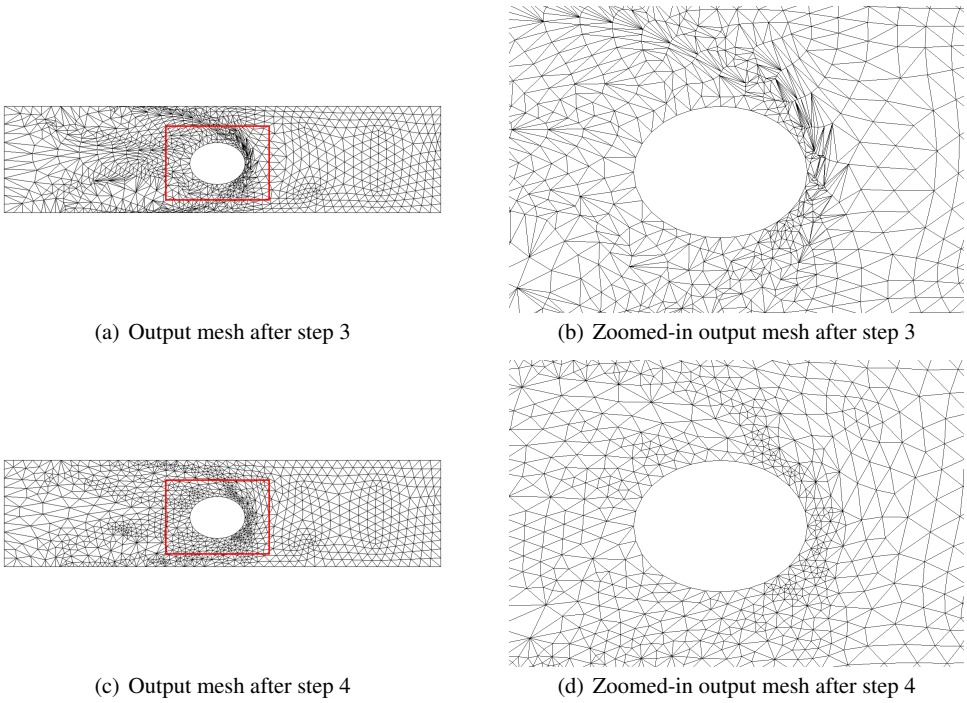


(e) Output mesh after step 2



(f) Zoomed-in output mesh after step 2

Fig. 3 Moving cylinder domain for anisotropic boundary deformation:(a) Initial mesh and (b) zoomed-in mesh on the cylinder domain. (c) Deformed mesh with anisotropic FEMWARP and (d) zoomed-in mesh with anisotropic FEMWARP. (e) Output mesh after performing multiobjective mesh optimization (f) zoomed-in mesh after performing multiobjective mesh optimization.



(a) Output mesh after step 3

(b) Zoomed-in output mesh after step 3

(c) Output mesh after step 4

(d) Zoomed-in output mesh after step 4

Fig. 4 Moving cylinder domain for anisotropic boundary deformation: (a) Output mesh after performing topological changes. (b) zoomed-in mesh after performing topological changes. (c) Output mesh after mesh smoothing. (d) zoomed-in mesh after mesh smoothing.

α=1 and β=0 because the deformation occurs along the x-axis. Figure 5 shows the initial mesh (Figure 5(a) and (b))and the output meshes after performing each step. The bottom two particles are initially close each other and move toward the boundary. Therefore, movement of the vertices in elements around each particle is highly constrained after the deformation occurs (Figure 5(c) and (d)). After deformation, 218 inverted and (or) anisotropic elements are generated—mostly around each particle. After performing multiobjective mesh optimization, all inverted elements are removed, but many of the elements are still anisotropic, and the movement of the corresponding vertices is highly constrained. (Figure 5(e) and (f)). However, we are able to more uniformly distribute the positions of these elements after performing topological changes. After performing topological changes, the worst element quality is improved approximately 93% (Figure 6(a) and (b)). After the final mesh smoothing pass, we are able to restore element quality similar to the input mesh(Figure 6(i) and (j)).

Table 4 shows mesh quality statistics and the number of inverted elements for the initial mesh and the output meshes after each step. We observe that the overall mesh quality of the final output mesh is similar to the initial mesh. Similar to previous examples, Knupp's mesh deformation algorithm is not able to remove in-

verted elements in the deformed mesh and instead increases the number of inverted elements.

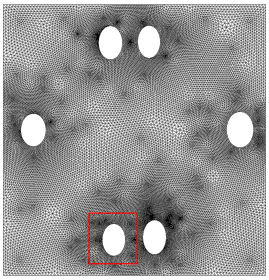
Table 4 Mesh quality statistics of moving particles domain measured by the inverse mean ratio quality metric and the number of inverted elements after each step. Here, 'each step' indicates the output mesh after each step.

Mesh quality	minimum	avg	rms	maximum	std.dev.	# of inverted elements
Initial	1	1.017	1.018	2.548	0.0430	0
Step 1	1	4893.13	69907.3	1e+06	69735.9	218
Step 2	1	2.889	17.421	1346.45	17.180	0
Step 3	1	1.348	1.774	93.742	1.154	0
Step 4	1	1.087	1.095	3.609	0.132	0
Knupp [28]	-0.00520	2.850	28.434	4631.670	28.290	647

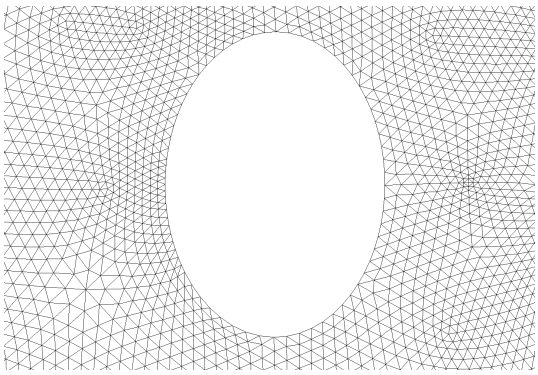
3.4 Summary of numerical results

Choice of PDE coefficients. We compare FEMWARP [17, 18] with anisotropic FEMWARP in terms of both the number of inverted elements and efficiency. FEMWARP always fixes the PDE coefficient as α=1 and β=1; however, anisotropic FEMWARP adaptively chooses the appropriate PDE coefficients with respect to the direction of deformation. Figure 7 shows the number of inverted elements after performing FEMWARP and anisotropic FEMWARP. The output mesh with anisotropic FEMWARP has up to 95.9% fewer number of inverted elements than the output mesh with isotropic FEMWARP. Since mesh untangling step is a relatively time-consuming step, these results indicate the importance of choosing appropriate PDE coefficients. Figure 8 shows the timing results for using FEMWARP and anisotropic FEMWARP to eliminate inverted elements using multiobjective mesh optimization (step 2). The FEMWARP algorithm by itself does not include the mesh untangling step. Therefore, for the purposes of comparison, we apply our step 2 to untangle the output meshes after applying FEMWARP. We observe that the use of anisotropic FEMWARP results in a decrease in the untangling time of up to 67.3% compared with using FEMWARP. Note that that the multiobjective optimization step, which performs untangling, is the slowest individual step of the developed algorithm. Therefore, it is desirable to have as fewer inverted elements as possible by choosing appropriate PDE coefficients.

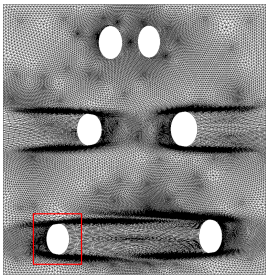
Mesh quality. Figure 9 and 10 shows the average and the worst element quality of the initial mesh and the final output mesh (after step 4), respectively, for each geometric domain. We observe that our framework is able to maintain good element quality even when large deformations were performed. For a moving bar and cylin-



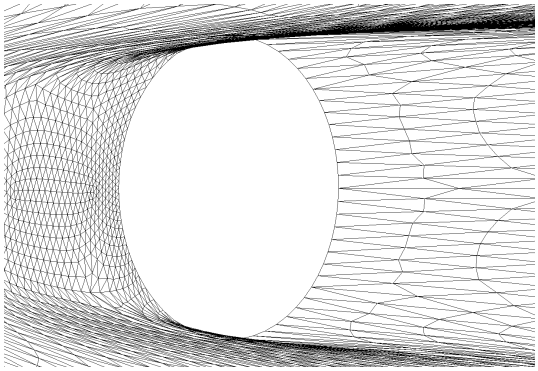
(a) Mesh on initial domain



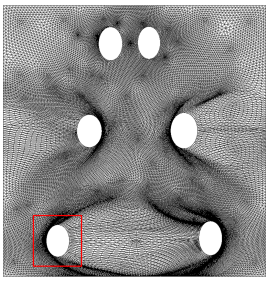
(b) Zoomed-in mesh on the initial domain



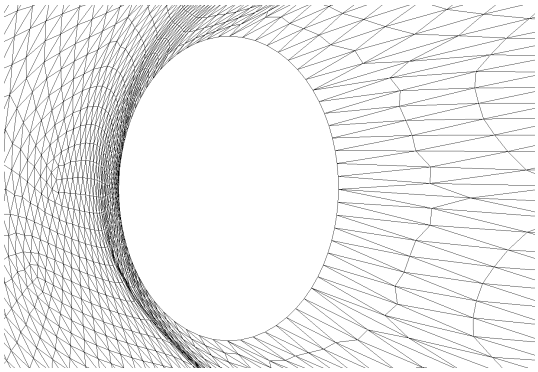
(c) Output mesh after step 1



(d) Zoomed-in output mesh after step 1

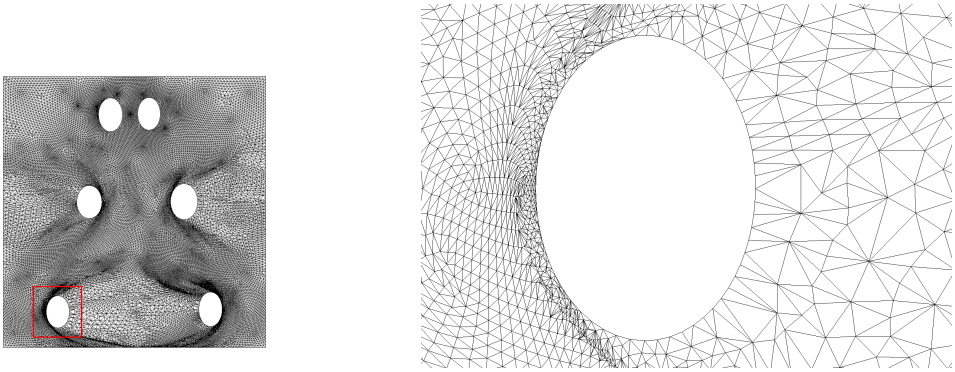


(e) Output mesh after step 2



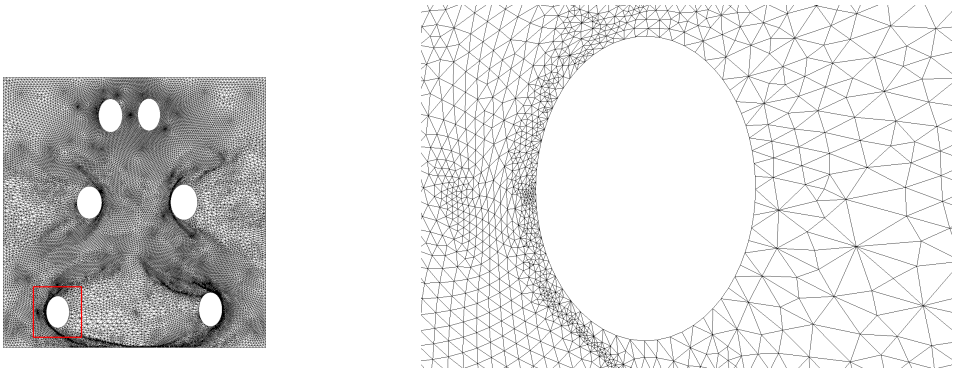
(f) Zoomed-in output mesh after step 2

Fig. 5 Moving particles domain for anisotropic boundary deformation:(a) Initial mesh and (b) zoomed-in mesh on the particles domain. (c) Deformed mesh with anisotropic FEMWARP and (d) zoomed-in mesh with anisotropic FEMWARP. (e) Output mesh after performing multiobjective mesh optimization (f) zoomed-in mesh after performing multiobjective mesh optimization.



(a) Output mesh after step 3

(b) Zoomed-in output mesh after step 3



(c) Output mesh after step 4

(d) Zoomed-in output mesh after step 4

Fig. 6 Moving particles domain for anisotropic boundary deformation: (a) Output mesh after performing topological changes. (b) zoomed-in mesh after performing topological changes. (c) Output mesh after mesh smoothing. (d) zoomed-in mesh after mesh smoothing.

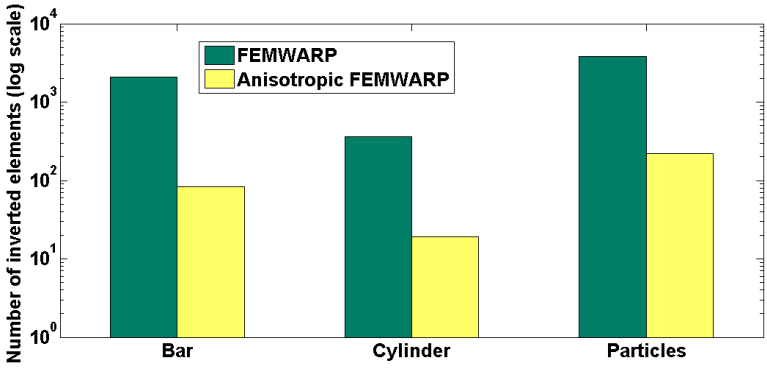


Fig. 7 Number of inverted elements after performing FEMWARP and anisotropic FEMWARP.

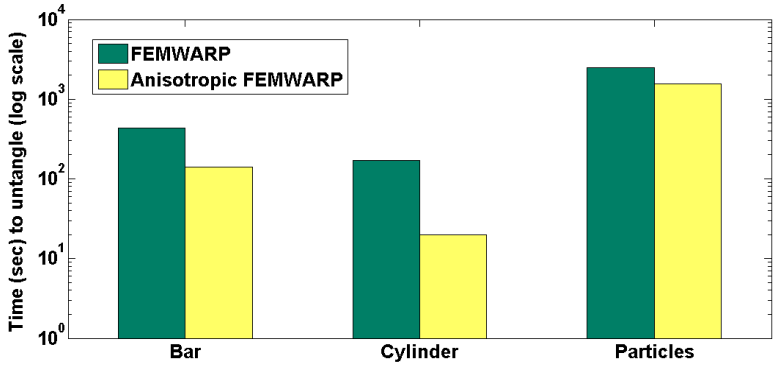


Fig. 8 Time (sec) to untangle inverted elements using multiobjective mesh optimization.

der domain, the initial and the final output meshes exhibit nearly identical mesh quality, which is highly desirable for PDE-based applications. For a challenging moving particles domain, our algorithm is able to maintain similar average mesh quality; although a slight increase in the worst element quality was noticed.

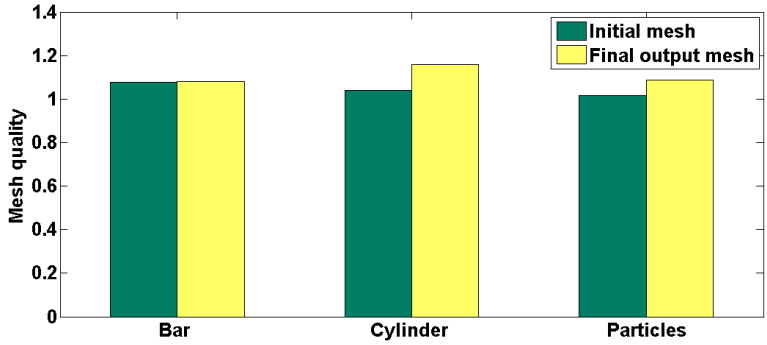


Fig. 9 Average element quality of the initial mesh and the final output mesh for each geometric domain measured by the IMR quality metric.

Timing results. Table 5 shows timing results of each step for three geometric domains. We compare our timing results with Knupp's deformation algorithm [28]. We observe that step 2 and step 3 are the slowest and fastest steps of the entire procedure, respectively. Step 3 significantly improves the overall element quality and distributes the elements whose vertex movement is highly constrained, and it takes less than 1 second to perform. This timing result justifies the motivation of using step 3 to rather than keep performing step 2 to improve the element quality.

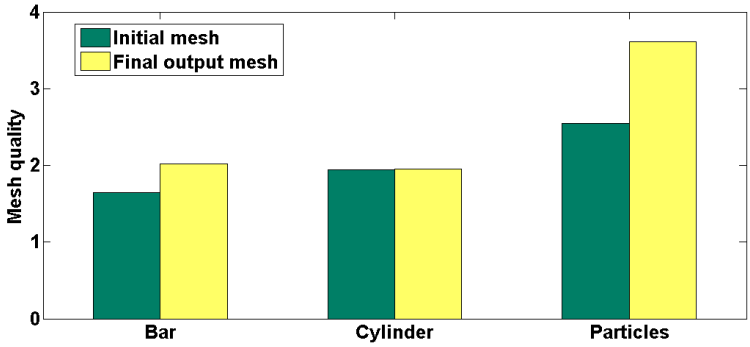


Fig. 10 Worst element quality of the initial mesh and the final output mesh for each geometric domain measured by the IMR quality metric.

Note that our algorithm performs topological changes (step 3) right after all inverted elements are eliminated. Knupp's mesh deformation algorithm is slightly faster to converge than our framework, but it results in an output mesh with inverted elements and poor element qualities.

Table 5 Timing results (sec) of each geometric domain.

Timing (sec)	Step 1	Step 2	Step3	Step 4	Knupp [28]
Bar (7K elements)	1.512	141.714	0.003	16.854	131.902
Cylinder (2K elements)	0.625	20.586	0.003	6.151	20.443
Particles (44K elements)	4.183	540.674	0.062	53.742	534.124

4 Conclusions

We propose a mesh deformation framework for anisotropic mesh deformations. Our framework is composed of four steps: 1) anisotropic FEMWARP, 2) multiobjective mesh optimization, 3) topological changes, 4) mesh smoothing. Numerical results show that our framework successfully eliminates inverted elements and keeps good element qualities on the deformed domain—even when applied to a large deformation. By choosing appropriate PDE coefficients, anisotropic FEMWARP is able to decrease the number of inverted elements up to 95.9%. Our second step, multiobjective mesh optimization successfully eliminates inverted elements while keeping good element quality. We observe that performing topological changes is an extremely efficient and effective way of increasing the amount of vertex movement possible on the final smoothing pass and hence improving the element quality on the deformed domain. Our final mesh smoothing step is able to further improve

the element quality. Numerical results show that our mesh deformation framework significantly outperforms Knupp's mesh deformation algorithm based on a target matrix paradigm framework [28].

Our topology-adaptive mesh deformation framework expands on our earlier hybrid mesh deformation framework by incorporating topological changes for additional mesh quality improvement as is required by large, anisotropic deformations. Because edge swaps yield a large improvement in the mesh quality very efficiently, we developed our initial framework in two dimensions. However, the hybrid mesh deformation algorithm upon which our work builds also works for three-dimensional mesh warping. Thus, our framework can be extended to three dimensions by the consideration of edge swaps and/or face swaps in three dimensions.

It is likely the case that the addition of other operations to alter the topology (e.g., edge splits, edge contractions, and multi-face removal) will lead to even further improvements in the mesh quality. In fact, such a strategy was used to evolve surface meshes in [34]. We plan to investigate the addition of such topological operations for our future research. Discrete optimization algorithms [35] can then be developed which improve the quality of the mesh by altering the mesh topology.

We also plan to compare the performance of our topology-adaptive mesh deformation framework with meshless techniques for mesh deformation (e.g., [36]). Meshless techniques do not use the mesh topology and hence are much faster than either PDE or optimization-based techniques for mesh deformation. However, more insight is needed as to how to appropriately choose a kernel function for use with meshless techniques for various types of mesh deformation.

Acknowledgments

The work of the third author is supported in part by NSF grant CAREER Award ACI-1330054 (formerly OCI-1054459). The authors wish to thank the anonymous referee for his/her careful reading of the paper and for the helpful suggestions which strengthened it.

References

1. J. Kim, B.J. Miller, and S.M. Shontz, *A hybrid mesh deformation algorithm using anisotropy and multiobjective mesh optimization*, Engineering with Computers, Submitted October 2013.
2. J. Kim, T. Panitanarak, and S.M. Shontz, *A multiobjective framework for mesh optimization*, International Journal for Numerical Methods in Engineering, 94(1):20-42, April 2013.
3. B.T. Helenbrook, *Mesh deformation using the biharmonic operator*, Internat. J. Numer. Methods Engrg., 56: 1007-1021, 2003.
4. M.M. Villone, M.A. Hulsen, P.D. Anderson, and P.L. Maffettone, *Simulations of deformable systems in fluids under shear flow using an arbitrary Lagrangian Eulerian technique*, Comput. Fluids, 90:88-100, 2014.

5. F. Pan, J. Kubby, and J. Chen, *Numerical simulation of fluid-structure interaction in a MEMS diaphragm drop ejector*, J. Micromech. Microeng., 12:70-76, 2002.
6. P. Crosetto, P. Reymond, S. Deparis, D. Kontaxakis, N. Stergiopulos, and A. Quateroni, *Fluid-structure interaction simulation of aortic blood flow*, Comput. Fluids, 43(1): 46-57, 2011.
7. F. Armero and E. Love, *An arbitrary Lagrangian-Eulerian finite element method for finite strain plasticity*, Int. J. Numer. Meth. Engng, 57:471-508, 2003.
8. L. Kaczmarczyk, M.M. Nezhad, and C. Pearce, *Three-dimensional brittle fracture: Configurational force-driven crack propagation*, Int. J. Numer. Meth. Engng., Published online ahead of print, 2013.
9. M.T. Bah, P.B. Nair, and M. Browne, *Mesh morphing for finite element analysis of implant positioning in cementless total hip replacements*, Med. Eng. Phys., 31:1235-1243, 2009.
10. M.A. Baldwin, J.E. Langenderfer, P.J. Rullkoetter, and P.J. Laz, *Development of subject-specific and statistical shape models of the knee using an efficient segmentation and mesh-morphing approach*, Comput. Meth. Prog. Bio, 97:232-240, 2010.
11. J. Park, S.M. Shontz, and C.S. Drapaca, *A combined level set/mesh warping algorithm for tracking brain and cerebrospinal fluid evolution in hydrocephalic patients*, Invited submission to Image-Based Geometric Modeling and Mesh Generation, Springer, Lecture Notes in Computational Vision and Biomechanics, Volume 3, 107-141, 2013.
12. J. Park, S.M. Shontz, and C.S. Drapaca, *Automatic boundary evolution tracking via a combined level set method and mesh warping technique: Application to hydrocephalus*, in Proc. of the MICCAI Workshop on Mesh Processing in Medical Image Analysis, pp. 122-133, 2012.
13. S.P. Sastry, J. Kim, S.M. Shontz, B.A. Craven, F.C. Lynch, K.B. Manning, and T. Panitanarak, *Patient-specific model generation and simulation for pre-operative surgical guidance for pulmonary embolism treatment*, Image-Based Geometric Modeling and Mesh Generation, Lecture Notes in Computational Vision and Biomechanics, 3:223-301, 2013.
14. A.W. F. Lee, D. Dobkin, W. Sweldens, P. Schroder, *Multiresolution mesh morphing*, in Proc. of the 26th SIGGRAPH Conference, pp. 343-350, 1999.
15. B. Klingner, *Tetrahedral Mesh Improvement*, Ph.D. Thesis, University of California at Berkeley, 2009.
16. M.L. Staten, S.J. Owen, S.M. Shontz, A.G. Salinger, and T.S. Coffey, *A comparison of mesh morphing techniques for 3D shape optimization*, Proc. of the 2011 International Meshing Roundtable, p. 293-312, 2011.
17. T.J. Baker, *Mesh movement and metamorphosis*, in Proc. of the 10th International Meshing Roundtable, 387-396, 2001.
18. S.M. Shontz and S.A. Vavasis, *Analysis of and workarounds for element reversal for a finite element-based algorithm for warping triangular and tetrahedral meshes*, BIT, Numerical Mathematics, 50: 863-884, 2010.
19. S.M. Shontz and S.A. Vavasis, *A mesh warping algorithm based on weighted Laplacian smoothing*, in Proc. of the 12th International Meshing Roundtable, pp. 147-158, 2003.
20. K. Stein, T. Tezduyar, and R. Benney, *Mesh moving techniques for fluid-structure interactions with large displacements*, Trans. ASME, 70: 58-63, 2003.
21. K. Stein, T. Tezduyar, and R. Benney, *Automatic mesh update with the solid-extension mesh moving technique*, Comput. Method Appl. M., 193:2019-2031, 2004.
22. S.M. Shontz, S.A. Vavasis, *A robust solution procedure for hyperelastic solids with large boundary deformation*, Eng. with Comp., 28:135-147, 2012.
23. Luke, E., E. Collins, and E. Blades, *A fast mesh deformation method using explicit interpolation*, J. Comput. Phys., vol. 231:586-601, 2012.
24. J. Antaki, G. Blelloch, O. Ghattas, I. Malcevic, G. Miller, and N. Walkington. *A parallel dynamic mesh Lagrangian method for simulation of flows with dynamic interfaces*. in Proc. of the 2000 Supercomputing Conference, pp. 26, 2000.
25. D. Cardoze, A. Cunha, G. Miller, T. Phillips, and N. Walkington. *A Bezier-based approach to unstructured moving meshes*, in Proc. of the 20th ACM Symposium on Computational Geometry, 2004.

26. D. Cardoze, G. Miller, M. Olah, and T. Phillips. *A Bezier-based moving mesh framework for simulation with elastic membranes*, in Proc. of the 13th International Meshing Roundtable, pp. 71-80. Sandia National Laboratories, 2004.
27. F. Alauzet and D. Marcum, *A closed advancing-layer method with changing topology mesh movement for viscous mesh generation*, Proc. of the 22*nd* International Meshing Roundtable, p. 241-262, 2013.
28. P. Knupp, *Updating meshes on deforming domains: An application of the target-matrix paradigm*, Commun. Num. Meth. Engr. 24:467-476, 2007.
29. Z. Yang and D.J. Mavripilis, *Mesh deformation strategy optimized by the adjoint method on unstructured meshes*, AIAA Journal, 45(12): 2885-2896, 2007.
30. X. Jiao, A. Colombi, X. Ni, and J.C. Hart, *Anisotropic mesh adaptation for evolving triangulated surfaces*, Proc. of the 15*th* International Meshing Roundtable, p. 173-190, 2006.
31. D. McLaurin, D. Marcum, M. Remotigue, E. Blades, *Algorithms and Methods for Discrete Surface Repair*, Ph.D. Thesis, Mississippi State University, 2010.
32. D. McLaurin, "Discrete Mesh Intersection Tutorial", http://www.simcenter.msstate.edu/docs/solidmesh/discretegridintersection.html, 2011.
33. M. Brewer, L. Freitag Diachin, P. Knupp, T. Leurent, and D. Melander, *The mesquite mesh quality improvement toolkit*, in Proc. of the 12*th* International Meshing Roundtable, Sandia National Laboratories, p. 239-250, 2003.
34. A. Zaharescu, E. Boyer, and R. Horaud, *Topology-adaptive mesh deformation for surface evolution, morphing, and multiview reconstruction*, IEEE Trans. Pattern Anal. Mach. Intell., 33(4):823-837, 2011.
35. JR. Shewchuk, *Two discrete optimization algorithms for the topological improvement of tetrahedral meshes*, Unpublished, 2002.
36. D. Sieger and S. Menzel and M. Botsch, *High quality mesh morphing using triharmonic radial basis functions*, Proc. of the 21*st* International Meshing Roundtable, pp. 1-15, 2013.