

# Smooth Rotation Enhanced As-Rigid-As-Possible Mesh Animation

Zohar Levi and Craig Gotsman

**Abstract**—In recent years, the As-Rigid-As-Possible (ARAP) shape deformation and shape interpolation techniques gained popularity, and the ARAP energy was successfully used in other applications as well. We improve the ARAP animation technique in two aspects. First, we introduce a new ARAP-type energy, named SR-ARAP, which has a consistent discretization for surfaces (triangle meshes). The quality of our new surface deformation scheme competes with the quality of the volumetric ARAP deformation (for tetrahedral meshes). Second, we propose a new ARAP shape interpolation method that is superior to prior art also based on the ARAP energy. This method is compatible with our new SR-ARAP energy, as well as with the ARAP volume energy.

**Index Terms**—As-Rigid-As-Possible (ARAP), shape deformation, shape interpolation

## 1 INTRODUCTION

To animate a 3D mesh, a modeler typically uses a mesh deformation tool to create new poses and a shape interpolation tool to generate the frames between the poses. A shape interpolation tool is also useful in other applications such as example-based deformation [4], [5], or morphing sequence generation by interpolation of two different meshes having a common topology [6]. While Linear Blending Skin (LBS) is the most popular animation technique, it suffers from a number of drawbacks. The main ones are the low quality of the deformation in terms of local shape preservation near joints (e.g. the well known “candy-wrapper” effect); the tedious task of painting the skin weights; and the need to bind a skeleton, which limits the deformation to articulated motion, thus preventing its use for intuitive freeform deformation. Many alternatives to LBS have been proposed, and in terms of user interface, the most convenient ones let the user manipulate handle points, and the rest of the mesh is automatically deformed in a natural way. Afterwards, a shape interpolation method is applied to create a natural animation between poses. The methods in [7], [2], [3] introduced an animation technique based on minimizing the so-called As-Rigid-As-Possible (ARAP) energy. The ARAP energy measures the local deviation of the differential of a mapping between two shapes from rigidity, shown to be advantageous for detail preservation and intuitive elastic behavior. The attractiveness of ARAP methods is in the simplicity of their formulation (built upon classical Laplacian mesh editing techniques), and not requiring any additional accessories or constructions. These lead to an easy implementation, while producing compelling results (at least for the volumetric case, as discussed

later on). Another feature of the ARAP energy is its ability to be easily converted to As-Similar-As-Possible energy, which produces, for example, a quasi-conformal mapping [8], [9]. Thus, it is not surprising that the ARAP energy became popular and has been used successfully in other applications such as parametrization [9], image registration [10], [11], shape decomposition [12], cage-based deformation [13], image warping [14], [15], and video stabilization [16]. Like many surface variational deformation methods, one drawback of the ARAP deformation method is performance; only coarse meshes can be deformed at interactive rates. One popular solution is using a hierarchical-based approach [17].

Given two meshes  $P$  and  $Q$  consisting of vertices  $p_i$  and  $q_i$  respectively, and directed edges  $p_{ij}$  ( $= p_j - p_i$ ) and  $q_{ij}$ , the discrete ARAP energy is defined as:

$$E(P, Q) = \sum_{k=1}^m \min_{R_k \in SO(3)} \sum_{(i,j) \in \mathcal{E}_k} c_{ijk} \|q_{ij} - R_k p_{ij}\|^2 , \quad (1)$$

where  $R_1, \dots, R_m \in SO(3)$  are optimal local rotations;  $\mathcal{E}_1, \dots, \mathcal{E}_m$  are their corresponding edge sets, typically the edges of a triangle, a tetrahedron, or a vertex 1-ring (see Fig. 2); and  $c_{ijk}$  are weighting coefficients, typically uniform or the familiar cotan weights. In the shape deformation setup, deforming a mesh  $P$  involves fixing handle points and solving for the rest of the  $q_i$  by minimizing (1). The intuition for minimizing the ARAP energy is to find a mesh  $Q$  that is locally a rigid transformation of the source mesh  $P$ . More specifically, the differential of a mapping from an edge set in  $P$  to a corresponding edge set in  $Q$  should be optimally a rotation, thus synchronizing corresponding edge vectors of the two edge sets. For example, if the positional constraints allow for  $Q$  to be a global rigid transformation of  $P$ , then all the  $R_k$  would be equal, and the energy would be zero.

While previous applications in 3D used the discrete

• Zohar Levi and Craig Gotsman are with the Technion, Israel.

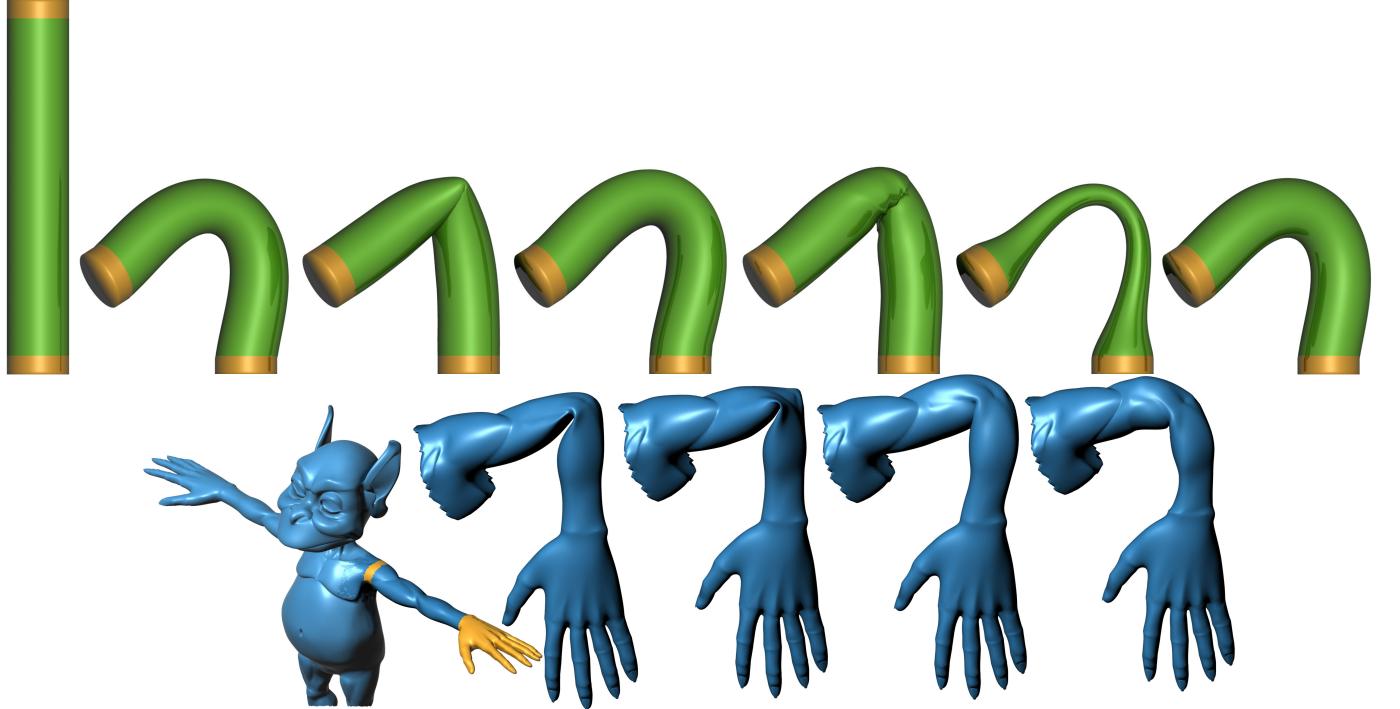


Figure 1: Survey cylinder deformation; from left to right: source, PriMo [1], ARAP surface [2], ARAP volume [3], ARAP volume applied to a tetrahedral stratum, ARAP surface with additional term for a smooth map differential, our SR-ARAP. Goblin’s arm deformation; from left to right: Source, LBS with default Autodesk Maya weights, ARAP surface, ARAP volume, SR-ARAP. The yellow regions are handle constraints.

ARAP energy with, somewhat, arbitrary weights  $c_{ijk}$ , Chao et al. [3] derive the discrete ARAP energy in (1) from the continuous energy of a smooth map between two manifolds  $f: P \rightarrow Q$ :

$$E(f) = \int_P \min_{R \in SO(3)} \|df - R\|_F^2 , \quad (2)$$

which results in cotan weights (that include an edge length factor for tetrahedra [3], [18]). This discretization is consistent (under appropriate refinement conditions and in appropriate norms), which is an essential requirement for a well-behaved method [19]. It exhibits a parametrization invariance behavior (e.g. reasonable consistency of shape deformation results for different discretizations, which is particularly useful in the case of a poor triangulation that has large variation in the size of the triangles), and convergence to the continuous case as the mesh is refined. Unfortunately, while the discretization of [3] is applicable for the volumetric case, where tetrahedral edge sets are used, it is not applicable for the surface case with 1-ring edge sets as in [2]. An effort has been made to address the surface case, and it has been argued in [3] that if 1-ring edge sets are used in the discrete surface scheme in [2], then the discretization can be derived from a new proposed continuous energy. No details were provided for how the derivation is done, and defining the radius of integration ( $r$  in their formula) is still an open problem. Thus, despite the popularity of the ARAP energy, so far there has been no consistent

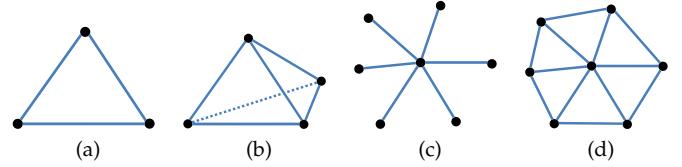


Figure 2: Typical edge sets: (a) triangle, (b) tetrahedron, (c) spokes, (d) spokes and rim (1-ring).

Typical neighboring edge sets of an edge set  $\mathcal{E}_k$ : (a) three triangles sharing an edge with  $\mathcal{E}_k$ , (b) four tetrahedra sharing a face with  $\mathcal{E}_k$ , (c-d) edge sets of the 1-ring vertex neighbors of the central vertex in  $\mathcal{E}_k$  (where an edge set is associated with its central vertex).

discretization for surfaces. We introduce a new ARAP-type energy, named SR-ARAP (ARAP with smooth rotations), which provides a consistent discretization for surfaces. The new energy is used for surface deformation that produces results with quality that competes with the volume deformation. We also propose a new ARAP shape interpolation scheme that is compatible with the SR-ARAP energy and with the ARAP volume energy, and improves the capabilities of previous ARAP shape interpolation methods.

### 1.1 Further Motivation

As further motivation for our new energy, we will note more drawbacks of previous ARAP deformation meth-

ods. As explained in the previous section, the ARAP surface deformation [2] is not a result of a consistent discretization. Moreover, Fig. 1 demonstrates additional problems with this scheme. The same problems that occur in a bending of a cylinder could also occur, e.g., at a joint in a more complex shape. The drawback of the ARAP surface deformation is that it aims at preserving the surface and ignores the volume, and consequently may introduce a squash in the deformed result. The ARAP volume deformation, on the other hand, results in a nice round shape, although not as rounded as PriMo due to a sensitivity to our particular tetrahedralization (see implementation notes in Section 4). The main disadvantage of the ARAP volume deformation is that it requires a tetrahedral mesh. Unlike the ubiquitous triangle mesh, tetrahedral mesh structures are not supported in most of the popular modeling software systems. Another restriction is that algorithms to construct a tetrahedral volume mesh from the interior of a triangle surface mesh require that the triangle mesh be a clean, water-tight, and non-self-intersecting surface. Furthermore, the tetrahedralization process generally introduces additional vertices, but even without them, the number of tetrahedral edge sets is larger than the number of 1-ring edge sets in the surface case, which directly influences the algorithm complexity. To emphasize the difficulty of the problem at hand (improving the quality of the ARAP surface deformation), we will iterate a few simple solutions that might come to mind, and explain why they do not work:

- Use a construction simpler than a complete tetrahedralization of the volume, e.g. a stratum of tetrahedra: Given a triangle mesh, for each pair of triangles sharing an edge, create a tetrahedron from the union of the triangle vertices. However, applying the ARAP volume deformation to the stratum results in a cramped surface similar to the squashed results of the surface case; see Fig. 1.
- A smooth map differential. Add to the ARAP surface energy a term that penalizes the difference between the map differential of an edge set and the map differential of its neighboring edge sets:

$$\begin{aligned} E_{smooth}(P, Q) = & \\ \sum_{k=1}^m \min_{R_k \in SO(3)} \sum_{(i,j) \in \mathcal{E}_k} c_{ijk} \|q_{ij} - R_k p_{ij}\|^2 & \\ + \sum_{\mathcal{E}_l \in N(\mathcal{E}_k)} w_{kl} \|df_k - df_l\|_F^2 & \end{aligned} \quad (3)$$

where  $N(\mathcal{E}_k)$  are the neighboring edge sets of  $\mathcal{E}_k$ ,  $w_{kl}$  are scalar weights, and  $df_i$  is the differential of the mapping from the edge set  $\mathcal{E}_i$  in  $P$  to the corresponding edge set in  $Q$ ; a detailed definition of the differential of a mapping is given in Section 3. Fig. 1 shows the resulting surface of the smooth map differential energy (minimized w.r.t.  $Q$ ) that exhibits extreme volume loss.

- Add a smoothing term for the surface itself, as done in surface fairing methods, e.g. adding a term based on the bi-Laplacian. The problem with this solution is that such a term would smooth out surface details.
- Use a larger edge set, such as a 2-ring. This does not have much impact on the resulting mesh. Taking a very large neighborhood may help the situation, but at impractical cost of memory and performance.

The last image in Fig. 1 shows a surface deformation based on our SR-ARAP energy.

## 1.2 Related Work

Since our application is a pure 3D mesh-based animation, we will review related methods in shape deformation and interpolation.

**Deformation** See [20] for a survey on surface deformation methods. The leading method described in this survey, in terms of quality, is PriMo [1]. PriMo emulates physically plausible surface behavior inspired by thin shells and plates. Nevertheless, the scheme, like many other direct deformation methods, is not a consistent discretization of a continuous energy. The implementation involves a construction of a layer of volumetric prisms, which are coupled through non-linear elastic forces. Most surface deformation methods are based on non-linear energy and rely on an iterative solver. Thus they are typically too expensive for interactive manipulation of high resolution models, and usually multiresolution is employed. Discrete Shells [21] model describes the behavior of thin flexible structures, where the energy is based on edge lengths and dihedral angles. Two versions of discretization are given: One that depends on the geometry only, and is triangulation invariant. The other is a discretization that converges to its continuous equivalent under refinement. It has been noted that there is no discretization that satisfies both properties. Pyramid coordinates [22] are a natural non-linear local representation of vertex positions, which is used for deformation. Linear rotation invariant (LRI) coordinates [23] offer a similar idea based on local frames, and the method is one of the few Laplacian-based deformation methods that can handle large rotations and run at interactive rates. However it does not cope well with situations where the handles undergo translation only. The reconstruction of the deformed mesh requires solving only two sparse linear systems that arise from discrete forms, but the method possibly requires more iterations to compete with the quality of state-of-the-art methods. Another method that is motivated by Cartan's moving frame is suggested in [24]. The method encodes the discrete first fundamental form at the local frame of a vertex, which is preserved through enforcing frames to be orthogonal. Also, it encodes the discrete second fundamental form as differences between the local frames in terms of quaternions. [25] performs constrained mesh deformation tasks with gradient domain techniques.

Sorkine et al. [2] propose a surface deformation based on the ARAP energy. Chao et al. [3] derive a similar ARAP formulation from the continuous case for volume deformation. [26], [8] base their ARAP deformation scheme on the Moving Least Squares (MLS) approach. Borosan et al. [13] introduce a hybrid approach that couples a surface deformation with a cage-based deformation. The user can perform edits on an automatically-generated simplified version of an input shape using ARAP surface modeling. The edit is propagated to the original shape by a precomputed space deformation based on Mean Value Coordinates. Manson et al. [17] build a low-resolution representation of a mesh by using edge collapses, and perform an ARAP deformation on the simplified mesh. Then details are added back by reversing edge-collapses, so that the shape of the mesh is locally preserved. While adding details, the mesh is deformed to match the predicted positions of constraints, so that constraints on the full-resolution mesh are met. Zollhofer et al. [27] present a novel lattice-based mesh editing that decouples the runtime complexity from the mesh geometric complexity. Its non-linear optimization minimizes an ARAP energy and is implemented as a data-parallel multi-resolution on the GPU, which allows to pose meshes consisting of millions of triangles in real-time. Jacobson et al. [28] offer a variant of Linear Blend Skinning (LBS) controlled by disconnected skeletons. The clustering of vertices is based on their distance in *weight space*, and the cluster transformations are optimized using the ARAP energy.

**Shape Interpolation** ARAP shape interpolation was introduced in [7]. The mapping differential for a tetrahedron is factorized using polar decomposition into rotation and stretching. The rotation is interpolated using Slerp (Spherical Linear Interpolation), and the stretching component is linearly interpolated. Baxter et al. [29] pointed out a drawback in using Slerp for interpolating rotations larger than 180 degrees, due to the Slerp selection of the shortest path. Their solution in 2D was to propagate the rotation phases using FFT. Winkler et al. [30] observed that it would be difficult to extend this method to 3D, where more than one rotation axis is involved, and instead proposed a hierarchical approach that interpolates edge lengths and dihedral angles. Frohlich et al. [5] note that the method in [30] is based on an energy similar to Discrete Shells [21], and using a similar approach, they add a volume interpolation term to the energy, and describe an efficient way to optimize it. Liu et al. [31] present a surface morphing method based on ARAP; the method cannot handle large rotations. Heeren et al. [32] offer a computational model for geodesics in the space of thin shells. They incorporate bending contributions into the deformation energy on top of membrane distortion terms in order to obtain a physically sound notion of distance between shells. Huang et al. [33] introduce an interactive approach to

generate physically-based shape interpolation between poses, by extending linear modal analysis. Pyramid coordinates [22], besides giving a shape deformation scheme, also offer a shape interpolation scheme, which was the first to handle large rotations. [34] formulates the trajectory problem of shape interpolation as solving Poisson equations. LRI [23] also can be used for shape interpolation. Kircher et al.'s relative blending [35] is similar to LRI, but the local frames are not orthonormal, and are defined on mesh faces instead of vertices. Baran et al. [36] extend LRI to contiguous disjoint patches. Chao et al. [3] propose a shape interpolation technique based on minimizing the ARAP energy, and compare their approach to interpolation in shape space [37]. The latter defines a Riemannian metric that penalizes non-isometric deformations, and search for geodesic paths in the resulting shape space. Gao et al. [38] propose a data-driven approach for shape morphing, based on clustering models from the same category in shape space. Huang et al. [39] offer a non-rigid shape registration method. Its underlying mechanism is based on a deformation similar to MLS, where the edge sets are clusters of vertices that go through a similar rigid transformation. A shape interpolation based on this mechanism is offered as well. The advantage of more recent methods is that they solve for the best rotations instead of interpolating or propagating given rotations.

### 1.3 Contribution

In this work, we introduce a new ARAP-type energy, which results in consistent discretization for surfaces, and further improves the quality of the surface deformation. Additionally, we propose a new ARAP shape interpolation scheme, which has better performance and extrapolation capabilities, and is compatible with both SR-ARAP energy and ARAP volume energy. Our contribution is magnified by the popularity of the ARAP energy in various applications.

### 1.4 Problem Summary

Before diving into the technical details, we summarize the problems to be addressed in this paper. In our surface deformation and interpolation scheme, we would like to achieve a volumetric deformation effect. This effect resolves or ameliorates shrinkage or collapse artifacts that previous methods are prone to; for example, see Fig. 1. To achieve that, our scheme includes a bending term, which we formulate in a way that can be consistently discretized as detailed in Section 2.2. The importance of a consistent discretization can be seen in our results in Figures 14, 15, and 16, which are compared to state-of-the-art methods that lack it, and thus show artifacts on non-uniform meshes.

## 2 DEFORMATION

### 2.1 SR-ARAP Energy

Our SR-ARAP energy for a smooth map between two 2-manifolds  $f : P \rightarrow Q$  is

$$E_{SR}(f) = \int_P \min_{R \in SO(3)} (\|df - R\|_F^2 + \alpha \hat{A} \|dR\|_F^2) . \quad (4)$$

The first term in the integral is a membrane energy as in (2), and the second term is a bending energy that penalizes the difference between rotations.  $\alpha \hat{A}$  is a weighting scalar, where  $\hat{A}$  is the area of the whole surface.  $\hat{A}$  was added to make the energy invariant to global scaling of  $P$  (the differential of the mapping does not change. But  $dR$  measure the difference between the same values over a larger area, so if  $P$  is scaled by  $s$ , the gradient is scaled by  $s^{-2}$ ). Normally, the differential of a mapping of a 2-manifold is a  $2 \times 2$  matrix, which maps tangent vectors from the parametric domain to a tangent plane at a surface point. Here,  $df$  is a  $3 \times 3$  matrix, which maps the 3D embedding of the tangent vectors from one surface to another (which is simply the Jacobian matrix of  $f : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ ). Our discretization is

$$\begin{aligned} E_{SR}(P, Q) = & \min_{R_1, \dots, R_m \in SO(3)} \sum_{k=1}^m \left( \sum_{(i,j) \in \mathcal{E}_k} c_{ijk} \|q_{ij} - R_k p_{ij}\|^2 \right. \\ & \left. + \alpha \hat{A} \sum_{\mathcal{E}_l \in \mathcal{N}(\mathcal{E}_k)} w_{kl} \|R_k - R_l\|_F^2 \right) \end{aligned} \quad (5)$$

where  $R_1, \dots, R_m \in SO(3)$  are optimal local rotations associated with the edge sets;  $\mathcal{N}(\mathcal{E}_k)$  are the neighboring edge sets of  $\mathcal{E}_k$ ;  $w_{kl}$  are scalar weights; and  $\hat{A}$  is the triangle mesh area, which is used to make the energy scale invariant (scaling the edges by  $s$ , would scale the first term by  $s^2$ , which is the scale of  $\hat{A}$  in the second term). The first term, the membrane energy, is similar to the discretization in (1), and the second term, the bending energy, penalizes the difference between an edge set rotation and the rotations of neighboring edge sets. The objective of the membrane term is to lower the distortion of an edge set (resist stretching and shearing), by keeping the map differential close to rigid. The objective of the bending term is to keep the variation in the rotations in an edge set neighborhood low, such that the neighborhood would transform as a unit, as much as possible.

In the ARAP surface deformation, an overlap of the cells is required in order to avoid surface stretching or shearing at the boundary of the cells, and a 1-ring edge set (Fig. 2c, 2d) is typically used [2]. Therefore, the ARAP surface deformation cannot cope with triangle edge sets (Fig. 2a), since there is nothing that accounts for the preservation of dihedral angles. For example, a mesh  $P$  that consists of two triangles with some dihedral angle between them, and a mesh  $Q$  that consists of the same two triangles, but with a different dihedral angle, would have a zero ARAP surface energy (optimal deformation) when using triangle edge sets. On the other hand, our

new SR-ARAP energy works well with triangle edge sets. This is because our energy prefers to rotate a triangle face and its neighbors with the same rotation (due to the low variation in rotations term), thus preserving the neighborhood shape. The bending term in (5) consists of the rotational part of the differential of the maps in the second term in (3). Comparing the second terms in (3) and (5), we conclude that (5) allows for independent stretching of the edge sets, thus putting more emphasis on the preservation of the dihedral angles, which leads to a better preservation of the local shape. We should note that the bending term is not pure bending due to geometric rigidity [40] (which intuitively states that two neighboring triangles cannot have purely different rotations without tearing the common edge apart).

### 2.2 Discretization Consistency

When discussing consistency, we do not refer to a pointwise convergence, but rather to the consistency defined by the respective approaches that are used to discretize the Laplace-Beltrami operator (and thus, the consistency is subjugated to the appropriate refinement conditions in the appropriate norms). For example, using the Finite Element Method (FEM), the convergence is in the weak sense (in the  $L^2$ -norm). We will now show that for the right choice of weights and triangle neighborhood, (5) is a consistent discretization of (4). While it is possible to use the SR-ARAP energy with 1-ring edge sets, as discussed in Section 4, for a consistent discretization we will consider triangle edge sets. Following [3], a consistent discretization of the membrane term is achieved using cotan weights for  $c_{ijk}$ . In fact, Pinkall et al. [41] address the discretization of a similar energy (Dirichlet energy) for the surface case. Minimizing the energy of the first term (with  $R$  fixed) results in a Poisson equation, which involves discretization of the Laplace-Beltrami operator. FEM and Discrete Exterior Calculus (DEC) are two popular techniques that are used to consistently discretize the Laplace-Beltrami operator, and they both result in the same discretization [42], [43], [44].

While the discrete variables of the membrane term are associated with the mesh vertices, the discrete variables of the bending term are associated with the mesh faces. A consistent discretization of the bending term, which is a Dirichlet energy w.r.t. the rotations, is then derived by applying a discrete Laplace operator to the faces (using typical triangle neighborhood for connectivity; see Fig. 2), where the values on the faces are the rotations. This discrete Laplace operator results in the weighted sum of the neighborhood rotations such as in (6), and the weights  $w_{kl}$  may be chosen to satisfy some properties, such as the linearity property or the convergence property (the convergence property of a differential operator implies consistency and stability) [44]. We will suggest three ways to discretize the Laplace-Beltrami operator for values associated with the mesh faces.

The first approach is to consider the barycentric dual

mesh, whose vertices are at the barycenters of the triangle faces of the primal mesh (the values associated with the primal mesh faces are associated with the dual mesh vertices), and it is triangulated, such that we get a simplicial complex [45]. Then, the standard consistent discrete cotan-Laplace operator (for vertices) [18] is employed (treating the dual mesh as a primal mesh). In terms of  $w_{kl}$ , the triangulation of the dual mesh induces a neighborhood for the triangle faces in the primal mesh, and  $w_{kl}$  are the cotan weights (of the dual mesh triangulation). We assume that the refinement of the primal mesh is well behaved, in the sense that the dual mesh has normal convergence to the continuous surface.

The second approach is also based on the barycentric dual mesh, but instead of triangulating it, we use the discrete Laplacian for general meshes [46] to derive the weights.

The third approach is to consider the circumcentric dual mesh, whose vertices are in the circumcenters of the triangle faces of the primal mesh. If the dual mesh is not well-centered (a vertex of the circumcentric dual mesh is not necessarily inside the associated primal face), then the dual mesh is not necessarily a simplicial complex, and its embedding in 3D may result in edges that are not straight. This, though, is not an obstacle for deriving a discrete Laplace-Beltrami operator. In the context of the DEC approach, when deriving the quantities for the discrete Hodge star, we consider the intrinsic dual mesh (mapping each pair of triangles isometrically to a 2D plane), which need not be well-centered [45]. We know that the weights for the discrete Laplace-Beltrami operator for the primal mesh are the ratio of the intrinsic dual edge length to the primal edge length. For the dual mesh, this ratio simply inverts. An explicit derivation of the *inverse cotan weights* is given in the Appendix.

In our experiments, the third approach did not perform well, and the first and second approaches performed without noticeable difference.

### 2.2.1 Rotations and the Frobenius Norm

Using the Frobenius norm to measure the difference between two rotation matrices in the bending term enabled us to easily derive a proper discretization. But further motivation should be given for this choice, since  $SO(3)$  is not closed under addition. Consider the rotation difference in the bending term

$$\|R_k - R_l\|_F^2 = 2\text{tr}(I - R_l^T R_k) = 6 - 2\text{tr}(R_l^T R_k) .$$

Define the rotation matrix  $R = R_l^T R_k$ .  $R$  can be represented by a rotation axis and an angle  $\theta$ . It is known that

$$\text{tr}(R) = 1 + 2\cos\theta .$$

Thus, the Frobenius norm measures the rotation angle of the composition of a rotation matrix and a second rotation matrix inverse, and its minimization favors a zero rotation angle (identity matrix).

## 2.3 Optimization

To minimize our energy in (5) w.r.t.  $q_i$  we employ the alternating local/global method from [2]. This is an efficient iterative method, where each iteration consists of a local step followed by a global step.

**Local Step** In the local step, the  $q_i$  are fixed, and the local rotations are optimized. Following [2], we solve for an optimal rotation  $R_k$  independently (fixing the other rotations). Dropping the terms in (5) that do not contain  $R_k$ , we get:

$$\begin{aligned} & \underset{R_k \in SO(3)}{\operatorname{argmin}} \left( \sum_{(i,j) \in \mathcal{E}_k} c_{ijk} \|q_{ij} - R_k p_{ij}\|^2 \right. \\ & \quad \left. + 2\alpha \hat{A} \sum_{\varepsilon_l \in \mathcal{N}(\mathcal{E}_k)} w_{kl} \|R_k - R_l\|_F^2 \right) \\ &= \underset{R_k \in SO(3)}{\operatorname{argmin}} \text{tr}(-R_k (2 \sum_{(i,j) \in \mathcal{E}_k} c_{ijk} p_{ij} q_{ij}^T \\ & \quad + 4\alpha \hat{A} \sum_{\varepsilon_l \in \mathcal{N}(\mathcal{E}_k)} w_{kl} R_l^T)) \\ &= \underset{R_k \in SO(3)}{\operatorname{argmax}} \text{tr}(R_k S_k) \end{aligned} \quad (6)$$

(in the bending term, the contribution is once from  $\mathcal{E}_k$ , and once from its neighbor) where  $S_k$  is defined as

$$S_k = 2 \sum_{(i,j) \in \mathcal{E}_k} c_{ijk} p_{ij} q_{ij}^T + 4\alpha \hat{A} \sum_{\varepsilon_l \in \mathcal{N}(\mathcal{E}_k)} w_{kl} R_l^T ,$$

and defining its SVD as  $S_k = U_k \Sigma_k V_k^T$  results in the optimal  $R_k = V_k U_k^T$ , since

$$\text{tr}(R_k S_k) = \text{tr}(V_k U_k^T U_k \Sigma_k V_k^T) = \text{tr}(I \Sigma_k)$$

is the maximum value, due to the identity matrix  $I$  being the orthonormal matrix with the maximal values on the diagonal. The difference from [2] is the added weighted sum of the rotations of the edge set neighbors to the covariance matrix in  $S_k$ .

**Global Step** In the global step the rotations are fixed, and we differentiate the energy in (5) w.r.t. the vertices. The derivative w.r.t.  $q_l$ :

$$\frac{\partial E}{\partial q_l} = 2 \sum_{k=1}^m \sum_{(i,j) \in \mathcal{E}_k} c_{ijk} \left( \frac{\partial q_{ij}}{\partial q_l} \right)^T (q_{ij} - R_k p_{ij})$$

where

$$\frac{\partial q_{ij}}{\partial q_l} = \begin{cases} -I_{3 \times 3} & i = l \\ I_{3 \times 3} & j = l \\ 0_{3 \times 3} & \text{else} \end{cases}$$

which results in a linear Poisson system  $LQ = b$  ( $L$  is the discrete Laplace-Beltrami operator). The positional constraints are incorporated into the system as usual (adding rows and columns of Lagrange multipliers, or eliminating rows and columns to get a positive-definite matrix). In a precomputation step,  $L$  is factorized, and solving the linear Poisson system amounts to a back substitution.

### 3 SHAPE INTERPOLATION

We start by introducing a new shape interpolation technique for tetrahedral meshes, and then we introduce a new shape interpolation technique for triangle meshes based on our SR-ARAP energy.

#### 3.1 Tetrahedral Mesh

Eq. (1) can also be written in terms of the differential of the mapping  $f : P \rightarrow Q$ , where  $P, Q$  are tetrahedral meshes [3]. For example, using cotan weights (that include the edge length factor) [3], [18]) for  $c_{ijk}$ :

$$E(P, Q) = \sum_{k=1}^m \min_{R_k \in SO(3)} A_k \|df_k - R_k\|_F^2 , \quad (7)$$

where  $df_k = Q_k P_k^{-1}$  is the differential of the mapping from the  $k$ th tetrahedron in  $P$  to the  $k$ th tetrahedron in  $Q$ ;  $P_k, Q_k$  are  $3 \times 3$  matrices whose columns are the vectors of three edges emanating from an arbitrary vertex in the  $k$ th tetrahedron in each of the meshes respectively (e.g.  $P_0 = [p_{01}|p_{02}|p_{03}]$ ); and  $A_k$  is the volume of the  $k$ th tetrahedron (for the general case, the vectors of the edges, which comprise the columns of the matrices, can be multiplied by weights to implement any arbitrary weighting in (1)). Chao et al. [3] propose a simple interpolation scheme that solves for a mesh  $M_t$  at time  $t$  by minimizing a convex combination of two energy terms

$$E_{\text{spring}}(M_0, M_1, t) = (1-t)E(M_0, M_t) + tE(M_1, M_t) ,$$

where  $M_0, M_1$  are the given meshes at times  $t = 0, 1$  to interpolate between. The minimization is performed using an optimization scheme similar to that in the deformation method. While this interpolation scheme solves the problem of sensitivity to large rotations in a straightforward manner, it has two drawbacks. The first drawback is that the shapes cannot be extrapolated in at least one of the time segments  $t < 0$  or  $t > 1$ . This is because in one of the time segments, the coefficient of the larger energy term is negative, and the total optimal energy would be negative infinity. The second drawback is that the cost of interpolating between two shapes is approximately twice the cost of a shape deformation process based on the same energy in (7). This is due to the optimization complexity being dominated by the SVD (or a cheaper polar decomposition [28]) calculation in the local step, which is performed twice in the shape interpolation compared to the deformation. In the general case, when interpolating between  $s$  meshes (e.g. as done in example-based deformation), we need to solve in the local step  $s \cdot m$  SVDs, compared to only  $m$  in the deformation.

We propose a new method that resolves both problems. Our method takes a more direct approach: We explicitly prescribe the stretching component, and then solve for the best rotations. Let  $f_0 : M_0 \rightarrow M_0$  (identity),  $f_1 : M_0 \rightarrow M_1$  be the mappings between corresponding

edge sets in the meshes, and  $df_{0k} = R_{0k}Y_{0k}$ ,  $df_{1k} = R_{1k}Y_{1k}$  be the respective polar decompositions of their mapping differentials for the  $k$ th tetrahedron. We define the interpolation at time  $t$  of the stretching part as

$$Y_k = (1-t)Y_{0k} + tY_{1k} . \quad (8)$$

Using the local/global scheme we solve for the vertices of an intermediate shape  $M_t$  by minimizing

$$\begin{aligned} E(M_0, M_1, t) &= \sum_{k=1}^m \min_{R_k \in SO(3)} A_k \|df_k - R_k Y_k\|_F^2 \\ &= \sum_{k=1}^m \min_{R_k \in SO(3)} \sum_{(i,j) \in \mathcal{E}_k} c_{ijk} \|q_{ij} - R_k Y_k p_{ij}^0\|^2 \\ &= \sum_{k=1}^m \min_{R_k \in SO(3)} \sum_{(i,j) \in \mathcal{E}_k} c_{ijk} \|q_{ij} - R_k \hat{p}_{ij}^0\|^2 , \end{aligned} \quad (9)$$

where  $df_k$  is the differential of the mapping  $f : M_0 \rightarrow M_t$  on the  $k$ th tetrahedron as in (7);  $A_k$  is the volume of the  $k$ th tetrahedron in  $M_0$ ;  $p_{ij}^0, q_{ij}$  are the edges of  $M_0, M_t$  respectively; and  $c_{ijk}$  are the cotan weights (with the edge length factor). Eq. (9) is the result of incorporating the interpolation of stretching part in the ARAP energy, and it provides further intuition: Solving for an intermediate shape is tantamount to deforming a stretched source mesh  $\hat{M}_0$  (with edges  $\hat{p}_{ij}^0 = Y_k p_{ij}^0$ ); compare Eq. (9) with Eq. (1). It can be proven for a mesh consisting of a single tetrahedron that the resulting mesh  $M_t$  would be identical (up to a rigid transformation) for our method and the method in [3], and in our experiments with meshes of several tetrahedra the difference between the results was imperceptible. The complexity of generating  $M_t$  with our method is the same as the ARAP volume deformation complexity, and since our energy is always non-negative, it can be safely extrapolated to any arbitrary time.

#### 3.2 Triangle Mesh

In Section 1.1 we discussed why working with a triangle mesh is more practical than working with a tetrahedral mesh. When attempting shape interpolation in the context of morphing between two different shapes, the advantage of triangle meshes is amplified, due to the difficulty off constructing a compatible tetrahedralization between two meshes. One of the obstacles is that the bijectivity of the mapping between the constructed meshes cannot be guaranteed. While the method in [3] can be used with the ARAP surface energy (with the limitations discussed before), it cannot be applied to our SR-ARAP energy. Since the strength of a shape interpolation method is based on its underlying energy, especially when leaving the example space (e.g. example-based deformation), we design a new shape interpolation method based on the SR-ARAP energy, using concepts similar to those used in our new interpolation method in the previous section.



Figure 3: Elephant extrapolation using SR-ARAP:  $t = -0.25, 0, 0.25, 0.5, 0.75, 1, 1.25$  (Source shapes are in yellow, interpolated shapes are in blue, and extrapolated shapes are in pink).

We now consider the maps' differentials  $df_{0k} = Q_k^0(P_k^0)^\dagger$ ,  $df_{1k} = Q_k^1(P_k^1)^\dagger$  for the surface edge sets (where  $\dagger$  stands for pseudoinverse), with  $P_k^i$ ,  $Q_k^i$  defined as before (but now of dimension  $3 \times |\mathcal{E}_k|$ , where  $|\mathcal{E}_k|$  is the number of edges in an edge set  $\mathcal{E}_k$ ), comprising of columns of weighted vector edges (e.g. uniform weights). The method in the previous section cannot be used here immediately, since the maps' differentials are not unique. When using a 1-ring edge set, the linear system that determines the map differential,  $df_{ik}P_k^i = Q_k^i$ , is overdetermined, and the surface cannot be uniquely determined. Thus, we use a triangle edge set ( $|\mathcal{E}_k| = 3$ ). The linear system that determines its map differential is underdetermined (three coplanar vectors), and thus the energy in (9) does not determine a unique shape. Therefore, we regularize the energy by prescribing rotations difference (enforcing them in the least-squares sense), such that the surface could be uniquely determined, and deviation from the interpolated bending is minimized. Note that penalizing the rotation smoothness instead, as in the deformation, is not enough. This is because it would only force the rotations to be smooth, while we need specific information (specific rotation difference) to determine a desired shape, which does not necessarily optimize the rotation smoothness energy. Given the specific rotation difference, the change in the dihedral angles can be inferred uniquely. For a map  $f_i$ , we define the rotation difference between two edge sets  $\mathcal{E}_l$ ,  $\mathcal{E}_k$  as  $dR_{ilk} = R_{ik}R_{il}^T$ , where  $R_{ik}$ ,  $R_{il}$  are the rotational components of the respective maps' differentials of the edge sets ( $dR_{0lk}$  is again the identity). We define the interpolation of the rotation difference as  $dR_{lk} = Slerp(dR_{0lk}, dR_{1lk}; t)$ , and the linear interpolation of the stretching part  $Y_k$  as in (8). We solve for an intermediate shape  $M_t$  by minimizing (w.r.t.  $q_i$ , the vertices of  $M_t$ )

$$\begin{aligned} E_{SR}(M_0, M_1, t) = & \\ \min_{R_1, \dots, R_m \in SO(3)} & \sum_{k=1}^m \sum_{(i,j) \in \mathcal{E}_k} c_{ijk} \|q_{ij} - R_k Y_k p_{ij}^0\|^2 \\ + \alpha \hat{A} \sum_{\ell \in \mathcal{N}(\mathcal{E}_k)} & w_{kl} \|R_k - dR_{lk} R_l\|_F^2 , \end{aligned} \quad (10)$$

where  $p_{ij}^0$  are the edges of the source mesh  $M_0$ . To reiterate, the first term in (10) prescribes new interpolated edges in an edge set. The second term prescribes the

interpolated rotation difference between the edge set and its neighbors, which varies between the identity matrix and the difference between the edge sets and its neighbors in  $M_1$ , and in essence prescribes an interpolated dihedral angle. Another perspective is instead of treating the reference mesh as  $M_0$ , we interpolate the reference mesh (between  $M_0$  and  $M_1$ ), and then the second term is just smoothing the rotations of the map differentials from the interpolated reference mesh (where smoothing can be viewed as prescribing zero rotation difference). Note that unlike the interpolation of edge set rotations with Slerp, interpolating the rotation difference with Slerp can be regarded as safe, since we expect the difference to be small for a map between similar shapes. The energy is minimized using the same local/global method described in Section 2.3.

## 4 RESULTS AND DISCUSSION

**Implementation** CGAL [47] was used for triangle mesh operations, TetGen [48] for tetrahedral mesh operations, and Eigen [49] for the linear algebra. In our experiments we did not use interior points when generating the tetrahedral meshes, and we used the surface of the tetrahedral mesh as an input for the surface case. The models were scaled to tightly fit inside a unit cube scaled by ten. The source mesh was used as an initial guess for the optimization process. For the SR-ARAP weights in Eq. (5), we experimented with cotan weights and uniform weights 1 for  $c_{ijk}$ . For the models with reasonable triangulation quality (where triangles have good aspect ratio, and their size does not vary much), using 1 for  $w_{kl}$  was usually enough to get compelling results. Otherwise, we used the weights from the first and second approaches in Section 2.2, which performed similarly. In both cases we normalized the weights to sum to one on a neighborhood. The effect of a proper discretization and the choice of weights is shown in Figures 14, 15, and 16. We used 0.01 for  $\alpha$ , which did not prove to be a sensitive parameter. With the ARAP surface energy we experimented with the 1-ring edge set, with and without rim edges, and uniform and cotan weights. With SR-ARAP we experimented with the three types of surface edge sets. On our models, the changes in the resulting mesh were imperceptible when using different edge sets for the surface energies. For

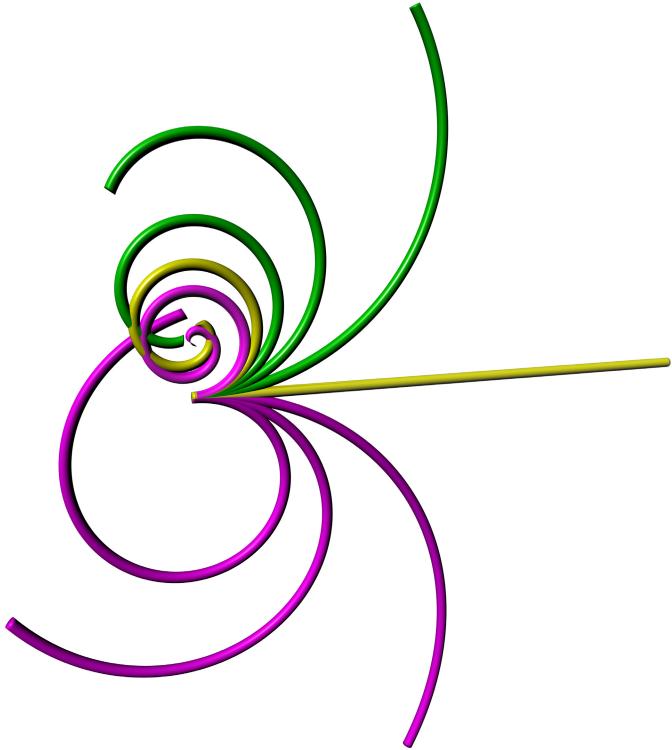


Figure 4: Spiral extrapolation using our formulation for ARAP volume:  $t = -0.25, 0, 0.25, 0.5, 0.75, 1, 1.25, 1.5, 1.75$  (Source shapes are in yellow, interpolated shapes are in green, and extrapolated shapes are in pink).

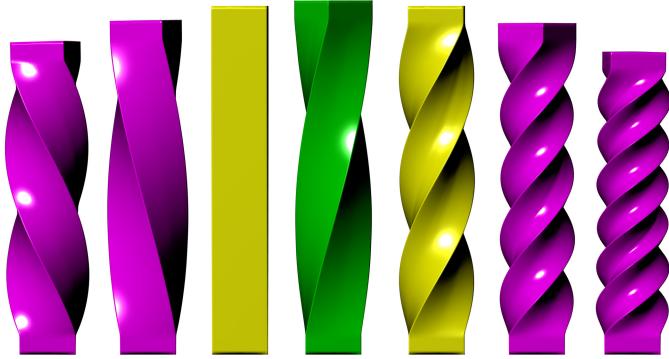


Figure 5: Bar twist extrapolation using our formulation for ARAP volume:  $t = -1, -0.5, 0, 0.5, 1, 1.5, 2$  (Source shapes are in yellow, interpolated shapes are in green, and extrapolated shapes are in pink).

the ARAP volume (tetrahedral edge sets) we used cotan weights. We stopped the iterative process when the maximum change in the mesh coordinates was below  $10^{-3}$ .

**Performance** It has been noted in [2] that the number of iterations required to get reasonably close to a minimum depends on the condition number of the anchored Laplacian matrix, which is generally proportional to the number of vertices. Moreover, it has been discussed in [1] that the main limitation of the local matching

Model	#Vertices	#Triangles	#Tetrahedra	Energy	#Iterations	Total (sec)
Cylinder	4.8	9.6	13	Tri	709	6.9
				Tet	111	1.7
				SR	195	2.5
Goblin arm	24	49	88	Tri	166	16
				Tet	79	7.6
				SR	285	32
Octopus	149	299	509	Tri	817	368
				Tet	338	160
				SR	480	267
Bar twist	6	12	18	Tri	124	3.8
				Tet	556	10
				SR	713	16
Elephant	47	95	167	Tri	684	203
				Tet	741	144
				SR	988	206

Table 1: Performance. We ran the experiments on a laptop with a 2.2Ghz Intel Core i7-2720QM CPU. #Vertices, #Triangles, and #Tetrahedra are the number of the model vertices, triangles, and tetrahedra in thousands; Energy is the type of energy that was used, where Tri stands for ARAP surface, Tet stands for ARAP volume, and SR stands for SR-ARAP; #Iterations is the number of iterations using the local/global method; and Total is the total deformation time in seconds. The first three models are the result of a shape deformation, and the last two models are the result of a shape interpolation at time  $t = 0.5$ .

is that it corresponds to an error diffusion process, and hence exhibits the typical behavior of an iterative smoother, which takes impractically long to converge. Thus, for an application that requires interactive rates, multi-resolution technique should be employed [17], [1], [30], [5]. Another approach that has been taken in previous work is to terminate the optimization after an arbitrary number of iterations [2], [28], compromising on the results quality. Our timings in Table 1 illustrate the need for such approaches. Since our contribution does not extend to optimization of the elements in a single iteration nor to lowering the number of needed iterations, we will not discuss the timings any further. We will focus instead on the theoretical runtime complexity of our new methods, which is independent of employing the previous approaches that improve the runtime.

We start by comparing the runtime complexity of the SR-ARAP and ARAP surface animation methods. Unlike the ARAP surface, the optimized rotations in the local step for SR-ARAP are codependent. Since we optimize each rotation independently, while fixing the others, the local step can be considered as a relaxation, thus more than one local iteration can be executed before performing the global step. In our experiments we found that using two local relaxations for each global iteration is a good tradeoff (depends on relaxation cost vs. global step cost, which are implementation dependent); see Fig. 10.



Figure 6: Armadillo interpolation using SR-ARAP:  $t = 0, 0.25, 0.5, 0.75, 1$  (source shapes are in yellow).

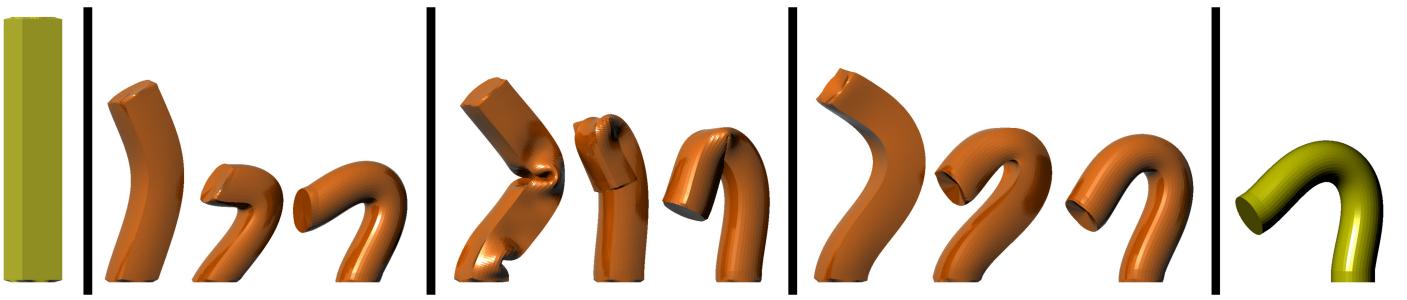


Figure 7: Morphing a bar into the survey cylinder:  $t = 0, 0.25, 0.5, 0.75, 1$  (source shapes are in yellow). There is a group of three intermediate shapes for each method; from left to right: Slerp, ARAP surface, SR-ARAP. A few vertices at the bottom of the bar are statically anchored, and the center vertex at the top of the bar is constrained and animated.



Figure 8: Lion interpolation using SR-ARAP:  $t = 0, 0.25, 0.5, 0.75, 1$  (source shapes are in yellow).

Another issue that needs to be addressed, due to the dependency, is parallelism. Similar to the red-black relaxation method, we partition the edge sets into five independent groups, allowing for the same level of parallelism as the ARAP surface. More specifically, we construct a graph, where each edge set is a vertex, and there is an edge between neighboring edge sets. We color the graph vertices, such that two vertices sharing an edge would have different colors [50]. We partition the edge sets according to the colors. Now, optimizing (6) for group can be done in parallel (we use OpenMP), since the neighbors of each edge set in the group are fixed (belong to another group).

Note that the number of 1-ring edge sets is half of the number of triangle edge sets (Euler's polyhedron formula), and the SR-ARAP shape interpolation can use only triangle edge sets. In our experiments we found that an SR-ARAP with two relaxations per iteration requires on the average the same number of iterations to converge as the conventional ARAP surface technique, making it twice as slow as the ARAP surface. Still, one iteration of SR-ARAP outperforms one iteration of the ARAP volume technique. For example, when considering deformation, SR-ARAP can use 1-ring edge sets, and the ARAP volume would need on the average more than three times tetrahedron edge sets, this without considering additional vertices (Steiner points). As explained in Section 3.1, our shape interpolation method for ARAP volume outperforms the method in [3] by a factor of the number of interpolated shapes. The complexity analysis may not be reflected in Table 1, due to implementation difference.

**Results** We demonstrate the SR-ARAP deformation on the benchmark models from the survey [20]; see Fig. 1, 9. Note that a comparison to the survey models implies a comparison to the surveyed methods, and subsequent works that performed this benchmark as well. Since the survey did not supply the results for the ARAP surface and volume methods, we provide them as well. Sorkine et al. [2] provide results on models that are

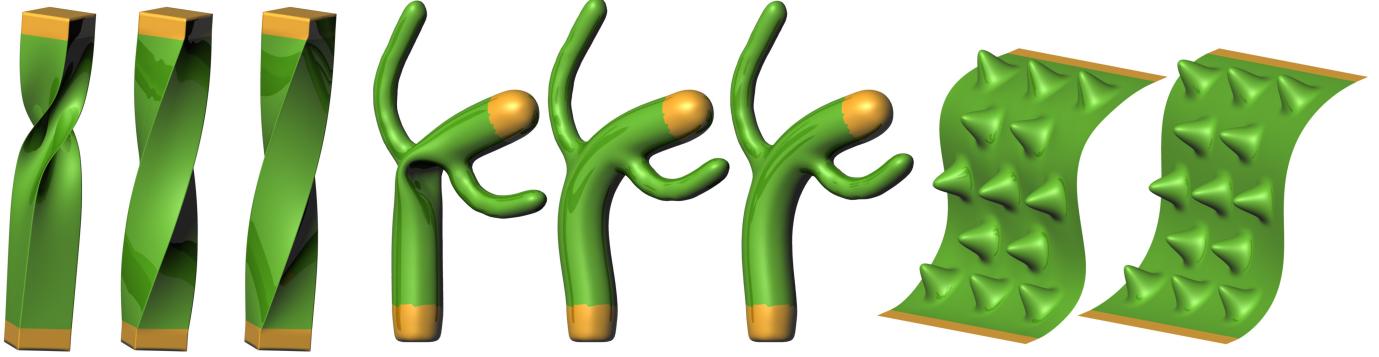


Figure 9: Survey models deformation; from left to right in each group: ARAP surface, ARAP volume, SR-ARAP (the plane has no volume).

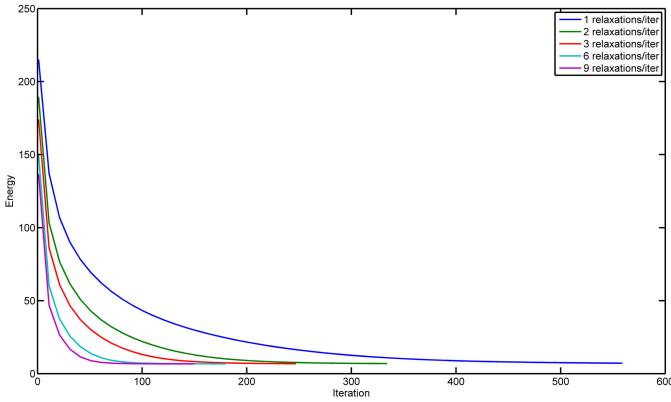


Figure 10: Number of relaxations per iteration and convergence rate tradeoff. Demonstrates the rate of convergence for different choices of number of relaxations per iteration.

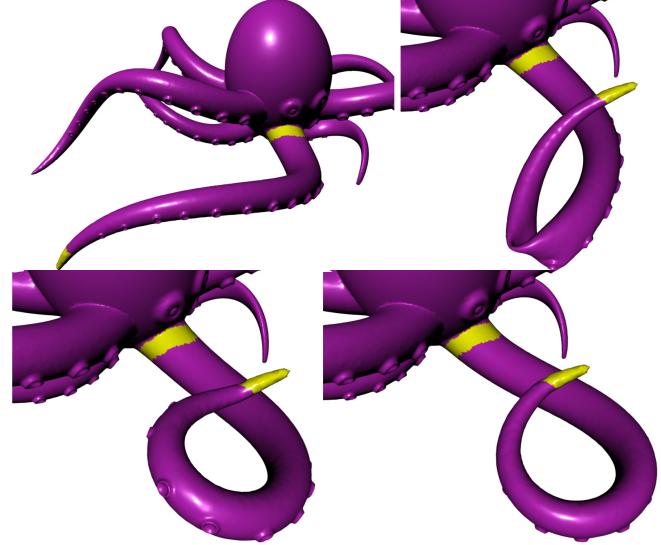


Figure 12: Octopus deformation, (Top left) source, (Top right) ARAP surface, (Bottom left) ARAP volume, (Bottom right) SR-ARAP.

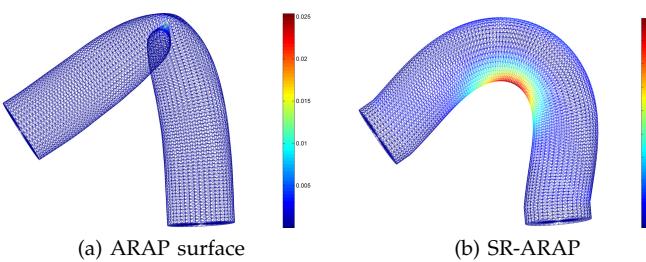


Figure 11: Distortion of local elements. We calculated the distortion of each 1-ring edge set, according to the ARAP energy in Eq. (1), i.e. its contribution for each edge set, and we colored the center vertex of each edge set. Warm colors are associated with larger values. The models are the corresponding results of Fig. 1. (a) distortion sum 0.937, max distortion 0.025, (b) distortion sum 6.23, max distortion 0.008.

similar to the survey models in shape only, and are far more coarse. For example, the bar in Fig. 3 in [2] consists of 500 vertices, while our cylinder in Fig. 1 consists of 4802 vertices - almost ten times as many. This gives us further intuition. When the models are coarse, each edge distortion weighs significantly in the total energy in (1), and thus the minimization distributes the error evenly. On the other hand, in models of finer resolution each triangle weighs significantly less, and the minimization prefers to incur a large distortion on a few triangles, such as those near the corner of the crease in Fig. 11a, for the price of a lower distortion in the rest of the triangles. When adding our rotation smoothness term in (5), this tradeoff does not pay off anymore. Thus, as can be seen in Fig. 11, SR-ARAP distributes the distortion far better, and while in  $l^2$ -norm it is worse than the ARAP surface energy, in  $\infty$ -norm it is better. The problem of using an  $l^2$ -norm energy motivated bounded error methods such as [51]. See Fig. 13, 12 for more deformation results.

A demonstration of our interpolation scheme is shown

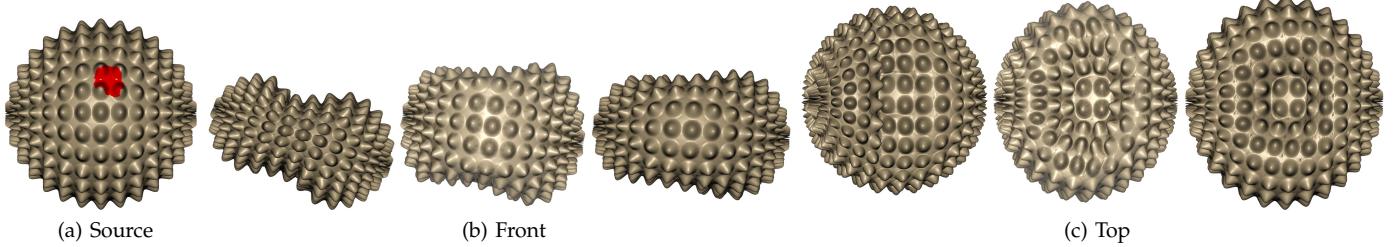


Figure 13: Bumpy sphere. The red region in the source consists of anchored vertices, and there is a similar region on the other side of the sphere. From left to right in each of the views: ARAP volume, ARAP surface, SR-ARAP. The ARAP volume does not preserve well the protrusions on the front. The ARAP surface does not preserve well the protrusions on the top, and some of them are stretched or connected to each other. Also, the surface itself resembles the cylinder in Fig. 1: The front and the top parts of the surface are flattened, and between them a corner is formed.

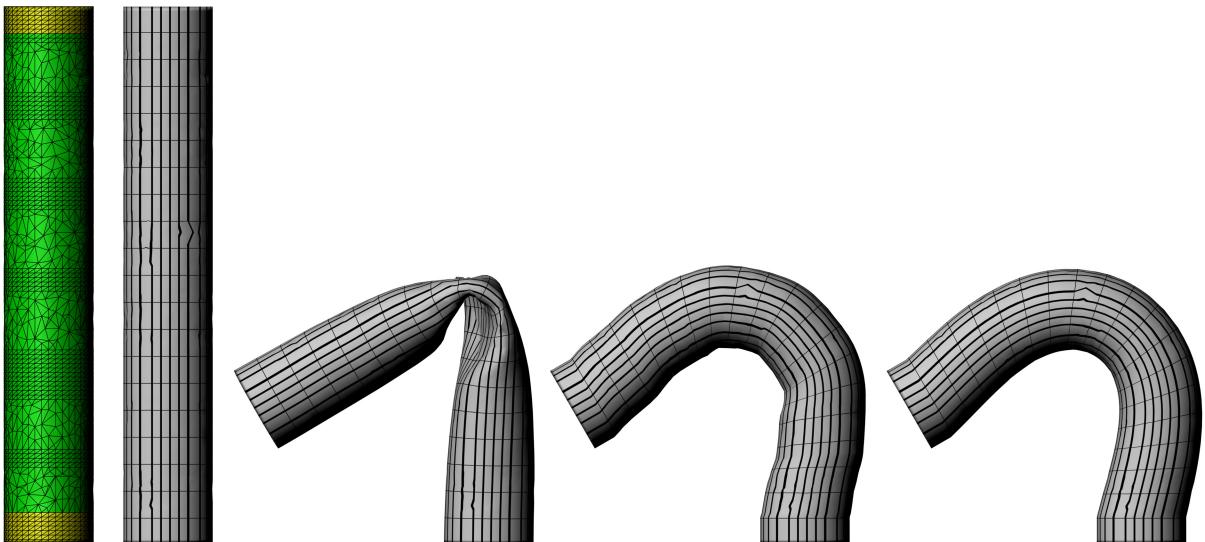


Figure 14: Cylinder deformation. It is the same cylinder from the survey, with some patches simplified. From left to right: source, source with texture, the method in [39], the method in [24], SR-ARAP. The first two methods use uniform weights and are influenced by the variation in triangle size, while SR-ARAP is not.

in Fig. 3-8. Most of the extrapolation is not possible using the method in [3]. The poses of the armadillo and the lion in Fig. 6, 8 are the same as in [30], and the poses of the elephant in Fig. 3 are the same as in [37], [30], [5]. The difference between the results of the different methods is not noticeable. In Fig. 7 we animated one of the constraints during the morphing, as done in example-based deformation. The first group shows a surface rotation Slerp similar to [7]. The difference between the methods is more evident here, outside the example space (the convex hull of the input shapes in shape space), where positional constraints are used. In the accompanying movie, we generated the shape interpolation animation using our proposed method first to interpolate a few intermediate shapes. Then, since the local rotations difference between the intermediate shapes is small (less than 180 degrees), we interpolated the rest of the frames between them using Slerp [7] (both for surface and volume), and one global step of

the local/global method to solve for the coordinates. The latter amounts to back-substitution, which has a performance similar to that of LBS.

## 5 CONCLUSION

The power of the ARAP energy lies in its simplicity. The implementation involves construction of the differential of the map from the vectors of the edge sets, polar decomposition for finding the best rotation, and a solution of a Poisson linear system. It does not require additional constructions like PriMo, or derivatives of dihedral angles such in Discrete Shells. We introduced a new ARAP-type animation technique that is based on a new SR-ARAP energy. Our technique fills the missing gaps in the popular ARAP methodology. It offers a consistent discretization, and overcomes weaknesses in the ARAP surface deformation [2], without compromising the ARAP simplicity. The quality of the resulting animation on surfaces competes with the results of the ARAP

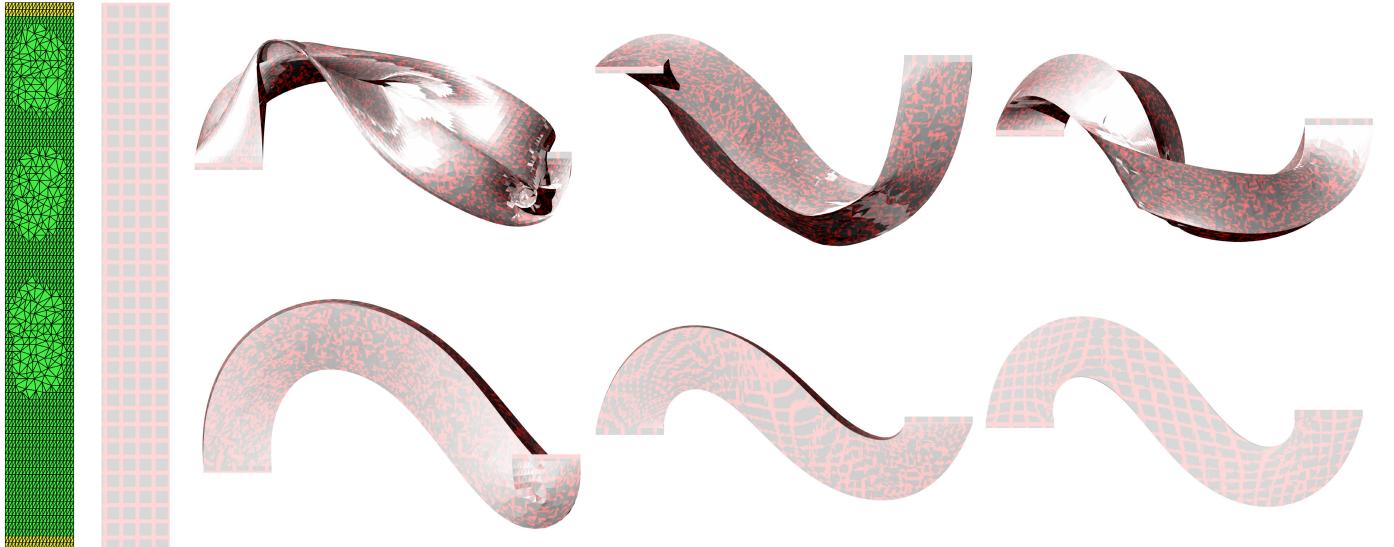


Figure 15: Bar deformation. Some patches on the bar are simplified. On the left are the source mesh and the source mesh with a reflection map [19]. The first row of results (from left to right): ARAP surface, the method in [39], the method in [24]. The second row of results: ARAP volume, SR-ARAP with cotan weights for  $c_{ijk}$  and uniform weights for  $w_{kl}$ , SR-ARAP with cotan weights for  $c_{ijk}$  and the weights from [46] for  $w_{kl}$ . The first row of results consists of methods with no proper discretization, which are influenced by the mesh degradation. ARAP volume was generated without inner points, and thus lacked the freedom to do both bends smoothly. The effect of the degradation on SR-ARAP, which moves the vertices almost only in-plane, is minimal. The difference between the choice of weights in the two SR-ARAP results is emphasized by the amount of preservation of the reflection map pattern.

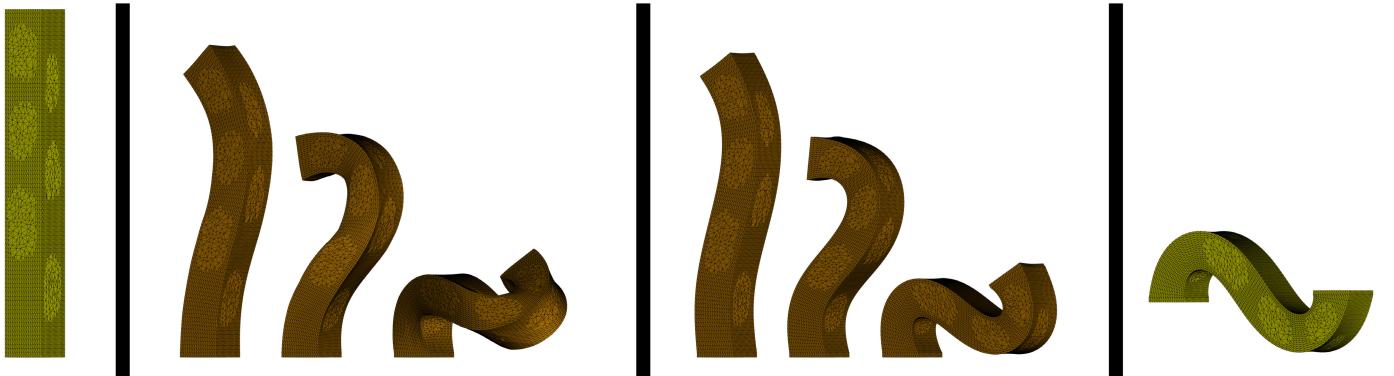


Figure 16: Bar interpolation:  $t = 0, 0.25, 0.5, 0.75, 1$  (source shapes are in yellow). There is a group of three intermediate shapes for the method in [39] (left) and for SR-ARAP (right). A few vertices at the bottom of the bar are statically anchored, and the center vertex at the top of the bar is constrained and animated. Some patches on the bar are simplified. This does not affect SR-ARAP, which moves the vertices only in-plane. On the other hand, the method in [39], which uses uniform weights, is influenced by the imbalance in the triangulation.

volume energy. We demonstrated the effectiveness of our technique in the application of shape deformation and shape interpolation. The latter inspired a new ARAP shape interpolation method that is superior to prior art also based on the ARAP energy. As a future avenue, it would be interesting to test our new SR-ARAP energy in other applications.

## ACKNOWLEDGMENTS

We wish to thank Max Wardetzky, Denis Zorin, and Julian Panetta for insightful discussions. We would also

like to thank Qixing Huang for the results of [39].

## REFERENCES

- [1] M. Botsch, M. Pauly, M. Gross, and L. Kobbelt, “Primo: coupled prisms for intuitive surface modeling,” in *SGP*, 2006.
- [2] O. Sorkine and M. Alexa, “As-Rigid-As-Possible surface modeling,” in *SGP*, 2007, pp. 109–116.
- [3] I. Chao, U. Pinkall, P. Sanan, and P. Schröder, “A simple geometric model for elastic deformations,” in *SIGGRAPH*, 2010, pp. 38:1–38:6.
- [4] R. W. Sumner, M. Zwicker, C. Gotsman, and J. Popović, “Mesh-based inverse kinematics,” in *SIGGRAPH*, 2005.

- [5] S. Frohlich and M. Botsch, "Example-driven deformations based on discrete shells," *CGF*, vol. 30, no. 8, pp. 2246–2257, 2011.
- [6] V. Kraevoy and A. Sheffer, "Cross-parameterization and compatible remeshing of 3D models," in *SIGGRAPH*, 2004.
- [7] M. Alexa, D. Cohen-Or, and D. Levin, "As-Rigid-As-Possible shape interpolation," in *SIGGRAPH*, 2000, pp. 157–164.
- [8] Y. Zhu and S. Gortler, "3D deformation using moving least squares," Harvard Univ.: TR-10-07, Tech. Rep., 2007.
- [9] L. Liu, L. Zhang, Y. Xu, C. Gotsman, and S. J. Gortler, "A local/global approach to mesh parameterization," in *SGP*, 2008, pp. 1495–1504.
- [10] Z. Levi and C. Gotsman, "D-Snake: Image registration by As-Similar-As-Possible template deformation," *TVCG*, 2013.
- [11] D. Sýkora, J. Dingliana, and S. Collins, "As-Rigid-As-Possible image registration for hand-drawn cartoon animations," in *NPAR*, 2009, pp. 25–33.
- [12] Q. Huang, M. Wicke, B. Adams, and L. Guibas, "Shape decomposition using modal analysis," *Computer Graphics Forum*, vol. 28, no. 2, pp. 407–416, 2009.
- [13] P. Borosan, R. Howard, S. Zhang, and A. Nealen, "Hybrid mesh editing," in *Eurographics (short papers)*, 2010, pp. 41–44.
- [14] H. Fadaifard and G. Wolberg, "Image warping for retargeting garments among arbitrary poses," *The Visual Computer*, vol. 29, no. 6–8, pp. 525–534, 2013.
- [15] H. Yu and J. J. Zhang, "Topology preserved shape deformation," *The Visual Computer*, vol. 28, no. 6–8, pp. 849–858, 2012.
- [16] Y.-S. Wang, F. Liu, P.-S. Hsu, and T.-Y. Lee, "Spatially and temporally optimized video stabilization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 8, pp. 1354–1361, 2013.
- [17] J. Manson and S. Schaefer, "Hierarchical deformation of locally rigid meshes," *CGF*, pp. 2387–2396, 2011.
- [18] M. Meyer, M. Desbrun, P. Schröder, and A. H. Barr, "Discrete differential-geometry operators for triangulated 2-manifolds," in *Proc. VisMath*, 2002, pp. 35–57.
- [19] E. Grinspun, Y. Gingold, J. Reisman, and D. Zorin, "Computing discrete shape operators on general meshes," *Eurographics*, vol. 25, no. 3, pp. 547–556, 2006.
- [20] M. Botsch and O. Sorkine, "On linear variational surface deformation methods," *TVCG*, vol. 14, no. 1, pp. 213–230, 2008.
- [21] E. Grinspun, A. N. Hirani, M. Desbrun, and P. Schröder, "Discrete shells," in *SCA*, 2003, pp. 62–67.
- [22] A. Sheffer and V. Kraevoy, "Pyramid coordinates for morphing and deformation," in *3DPVT*, 2004, pp. 68–75.
- [23] Y. Lipman, O. Sorkine, D. Levin, and D. Cohen-Or, "Linear rotation-invariant coordinates for meshes," in *SIGGRAPH*, 2005, pp. 479–487.
- [24] N. Paries, P. Degener, and R. Klein, "Simple and efficient mesh editing with consistent local frames," in *Pacific Conference on Computer Graphics and Applications*, 2007, pp. 461–464.
- [25] J. Huang, X. Shi, X. Liu, K. Zhou, L.-Y. Wei, S.-H. Teng, H. Bao, B. Guo, and H.-Y. Shum, "Subspace gradient domain mesh deformation," in *SIGGRAPH*, 2006, pp. 1126–1134.
- [26] A. Cuno, C. Esperanca, A. Oliveira, and C. P. Roma, "3D As-Rigid-As-Possible deformations using MLS," in *Proc. of Computer Graphics International Conference*, 2007.
- [27] M. Zollhöfer, E. Sert, G. Greiner, and J. Süßmuth, "GPU based ARAP deformation using volumetric lattices," in *Eurographics (Short Papers)*, 2012, pp. 85–88.
- [28] A. Jacobson, I. Baran, L. Kavan, J. Popović, and O. Sorkine, "Fast automatic skinning transformations," *SIGGRAPH*, vol. 30, no. 4, 2012.
- [29] W. Baxter, P. Barla, and K. Anjyo, "Rigid shape interpolation using normal equations," in *NPAR*, 2008, pp. 59–64.
- [30] T. Winkler, J. Drieseberg, M. Alexa, and K. Hormann, "Multi-scale geometry interpolation," *CGF*, 2010.
- [31] Y. Liu, H. Yan, and R. R. Martin, "As-rigid-as-possible surface morphing," *J. Comput. Sci. Technol.*, vol. 26, no. 3, pp. 548–557, 2011.
- [32] B. Heeren, M. Rumpf, M. Wardetzky, and B. Wirth, "Time-discrete geodesics in the space of shells," *Comp. Graph. Forum*, vol. 31, no. 5, pp. 1755–1764, 2012.
- [33] J. Huang, Y. Tong, K. Zhou, H. Bao, and M. Desbrun, "Interactive shape interpolation through controllable dynamic deformation," *TVCG*, vol. 17, no. 7, pp. 983–992, 2011.
- [34] D. Xu, H. Zhang, Q. Wang, and H. Bao, "Poisson shape interpolation," in *Symposium on Solid and Physical Modeling*, 2005, pp. 267–274.
- [35] S. Kircher and M. Garland, "Free-form motion processing," *ACM Trans. Graph.*, vol. 27, no. 2, pp. 12:1–12:13, 2008.
- [36] I. Baran, D. Vlasic, E. Grinspun, and J. Popović, "Semantic deformation transfer," *ACM Trans. Graph.*, vol. 28, 2009.
- [37] M. Kilian, N. J. Mitra, and H. Pottmann, "Geometric modeling in shape space," in *SIGGRAPH*, 2007.
- [38] L. Gao, Y.-K. Lai, Q.-X. Huang, and S.-M. Hu, "A data-driven approach to realistic shape morphing," *Comput. Graph. Forum*, vol. 32, no. 2, pp. 449–457, 2013.
- [39] Q. Huang, B. Adams, M. Wicke, and L. J. Guibas, "Non-rigid registration under isometric deformations," in *SGP*, 2008, pp. 1449–1457.
- [40] G. Friesecke, R. D. James, and S. Müller, "A theorem on geometric rigidity and the derivation of nonlinear plate theory from three-dimensional elasticity," *Comm. Pure Appl. Math.*, vol. 55, no. 11, pp. 1461–1506, 2002.
- [41] U. Pinkall and K. Polthier, "Computing discrete minimal surfaces and their conjugates," *Experimental Mathematics*, vol. 2, no. 1, pp. 15–36, 1993.
- [42] K. Crane, F. de Goes, M. Desbrun, and P. Schröder, "Digital geometry processing with discrete exterior calculus," in *SIGGRAPH Courses*, 2013, pp. 7:1–7:126.
- [43] B. Lévy and H. R. Zhang, "Spectral mesh processing," in *SIGGRAPH Courses*, 2010, pp. 8:1–8:312.
- [44] M. Wardetzky, S. Mathur, F. Kälberer, and E. Grinspun, "Discrete Laplace operators: no free lunch," in *SGP*, 2007, pp. 33–37.
- [45] A. N. Hirani, "Discrete exterior calculus," Ph.D. dissertation, California Institute of Technology, 2003.
- [46] M. Alexa and M. Wardetzky, "Discrete laplacians on general polygonal meshes," in *SIGGRAPH*, 2011, pp. 102:1–102:10.
- [47] "CGAL, Computational Geometry Algorithms Library," <http://www.cgal.org>.
- [48] H. Si, "TetGen: a quality tetrahedral mesh generator and three-dimensional delaunay triangulator." URL:<http://tetgen.berlios.de>, 2007.
- [49] G. Guennebaud, B. Jacob et al., "Eigen v3," <http://eigen.tuxfamily.org>, 2010.
- [50] N. Robertson, D. P. Sanders, P. Seymour, and R. Thomas, "Efficiently four-coloring planar graphs," in *Symposium on Theory of Computing*, 1996, pp. 571–575.
- [51] Y. Lipman, "Bounded distortion mapping spaces for triangular meshes," *TOG*, vol. 31, no. 4, pp. 108:1–108:13, 2012.
- [52] M. Desbrun, A. N. Hirani, M. Leok, and J. E. Marsden, "Discrete exterior calculus," *arXiv:math/0508341v2*, 2005.



**Zohar Levi** received his PhD in computer science from the Technion, Israel, in 2013. His research interests include computer graphics and geometry processing.



**Craig Gotsman** received a PhD in computer science from the Hebrew University of Jerusalem. Since 1991, he has been on the Faculty of Computer Science at the Technion at Haifa, Israel, where he co-founded the Center for Graphics and Geometric Computing (CGGC). His main research interests are computer graphics, geometry processing and geometric modeling.

## APPENDIX: INVERSE COTAN WEIGHTS

Following Section 6.4 in [45] (a similar derivation is given in Section 9 in [52], and in Section 6.3 in [42]), we develop the explicit formula for the *inverse cotan-Laplace*. We compute  $\Delta R$  on a dual vertex  $\star\sigma^2$  (corresponding to the primal triangle face  $\sigma^2$ ). We have that

$$\begin{aligned} \langle \Delta R, \star\sigma^2 \rangle &= \langle \delta dR, \star\sigma^2 \rangle \\ &= -\langle \star d \star dR, \star\sigma^2 \rangle . \end{aligned}$$

Using the definition of the discrete Hodge star, followed by the discrete Stokes' theorem (and the definition for the volume of a vertex  $|\star\sigma^2| = 1$ ), we get

$$\begin{aligned} &= -\frac{|\star\sigma^2|}{|\sigma^2|} \langle d \star dR, \sigma^2 \rangle \\ &= -\frac{1}{|\sigma^2|} \langle \star dR, \partial(\sigma^2) \rangle . \end{aligned}$$

By the definition of the boundary operator

$$\begin{aligned} &= -\frac{1}{|\sigma^2|} \langle \star dR, \sum_{\sigma^1 \in \sigma^2} \sigma^1 \rangle \\ &= -\frac{1}{|\sigma^2|} \sum_{\sigma^1 \in \sigma^2} \langle \star dR, \sigma^1 \rangle , \end{aligned}$$

where  $\sigma^1$  is one of the three edges in  $\sigma^2$ . Using the definition of the discrete Hodge star:

$$\begin{aligned} &= -\frac{1}{|\sigma^2|} \sum_{\sigma^1 \in \sigma^2} \frac{|\sigma^1|}{|\star\sigma^1|} \langle dR, \star\sigma^1 \rangle \\ &= -\frac{1}{|\sigma^2|} \sum_{\sigma^1 \in \sigma^2} \frac{|\sigma^1|}{|\star\sigma^1|} (R(v_j) - R(v_i)) , \end{aligned}$$

where  $v_i, v_j$  are two dual vertices, which are the end points of a dual edge  $\star\sigma^1$ . To remind,  $|\sigma^2|$  is the area of a primal triangle face, and

$$\frac{|\star\sigma^1|}{|\sigma^1|} = \frac{1}{2} (\cot\alpha + \cot\beta) ,$$

where  $\alpha, \beta$  are two opposing angles to a primal edge  $\sigma^1$ . One pitfall that should be noted is that a dual edge cannot have zero length.