Journal of Zhejiang University-SCIENCE C (Computers & Electronics)
ISSN 1869-1951 (Print); ISSN 1869-196X (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

Scale-Aware Shape Manipulation*

Zheng Liu[†1], Wei-ming Wang[2], Xiu-ping Liu[2], Li-gang Liu[1]

[1]*School of Mathematical Sciences, University of Science and Technology of China, 230026, China*

[2]*School of Mathematical Sciences, Dalian University of Technology, 116023, China*

[†]E-mail: Liu.Zheng.jojo@gmail.com

Received Jan. 6, 2010; Revision accepted Apr. 10, 2010; Crosschecked May 8, 2010

Abstract: We propose a novel representation of triangular mesh surface using a set of scale-invariant measures. The measures consist of angles of the triangles (triangle angles) and dihedral angles along the edges (edge angles) which are scale and rigidity independent. We can reconstruct the vertex coordinates for a mesh given its scale-invariant measures, unique up to scaling, rotation and translation. Based on the representation of mesh using scale-invariant measures, we propose a two-step iterative deformation algorithm which can arbitrarily editing the mesh through simple handles interaction. The algorithm can explicitly preserve the local geometric details as much as possible in different scales even under severe editing operations including rotation, scaling and shearing. We demonstrate efficiency and robustness of our algorithm by examples.

Key words: Differential coordinates, Scale-invariant measures, Surface deformation

doi:10.1631/jzus.C1000000 **Document code:** A **CLC number:**

1 Introduction

Shape manipulation has wide applications in computer graphics and animation. A variety of techniques have been developed to transform an original shape into a new one under a certain number of constraints. These techniques can be used to develop efficient deformation tool to provide physically plausible and aesthetically pleasing surface deformation results, which in particular requires its local geometric details to be preserved as much as possible. Any smooth deformation can be decomposed into three modes, namely, rotation, scaling and shearing. We want to find an efficient local encoding of geometric details which can facilitate intuitive surface manipulation and deformation in the three modes above.

In this paper, we introduce a novel differential representation of the 3D triangular mesh surface using a set of angle measures (triangle angles and edge angles) which are invariant to rigid and isotropic scale transformations. And the representation determines a unique surface up to global similarity if the measurements are from existing mesh. The representation is designed especially to be shape preserving in different scales for arbitrarily deformation.

As we have known, the differential representations used in existing works are often absolute length measures, such as edge lengths (Sorkine and Alexa, 2007), local frames (Kircher and Garland, 2008; Lipman *et al.*, 2005; Paries *et al.*, 2007; Wang *et al.*, 2012), differential coordinates (Sheffer and Kraevoy, 2004; Sorkine *et al.*, 2004; Yu *et al.*, 2004), which cannot efficiently handle the scaling deformations furthermore some cannot preserve geometric details in shearing deformations. The reason is that, the absolute length measures can hardly be preserved under such deformation operations. In contrast our scale-invariant measures only encode the relative angles, due to the scale invariance our method aims to find a discrete approximation for a conformal immersion of the source mesh with additional edge angle constraints. We believe that scale variation and

[†] Corresponding author

* Project supported by the National Natural Science Foundation of China (No. 61222206) and the One Hundred Talent Project of the Chinese Academy of Sciences.

conformal deformation are desirable in digital geometry processing, because they do not exhibit shear and therefore preserve texture fidelity as well as the quality of the mesh itself (Crane *et al.*, 2011; Paries *et al.*, 2007).

The angle measurements (triangle angles and edge angles) are then used for a two-step iterative deformation algorithm which can efficiently preserve geometric details in arbitrarily deformation operations through some point and orientation constraints. The basic idea is to minimize the deviation of each vertex 1-ring neighborhoods in the ℓ_2-sense, ignoring rotation, translation, and scaling. The vertex and its 1-ring neighborhoods are encoded using triangle angles and edge angles between consecutive edges. The energy is minimized using an alternating scheme, fixing either face normals or vertex coordinates during each iteration.

Our work endeavors to preserve the local geometric details under the large-scale editing operations using our scale-invariant measures. We can easily note the characteristics of our algorithm in Fig. 9 and Fig. 11, and find that our algorithm is more effective than the other algorithms (Lipman *et al.*, 2005; Paries *et al.*, 2007; Sorkine and Alexa, 2007; Sorkine *et al.*, 2004) to preserve the local features. We can get an optimal rigidity approximation of the deformed shape by rescaling the scale factors to the same one, which provide an option to rescale the local shapes. We can also use the scale factors to magnify the interesting regions to different degrees without losing the detail features.

2 Related work

Deformation is an important part in shape manipulation. The deformation approaches can be classified as surface based methods, space based methods, volumetric deformation, point cloud deformation, etc. It is beyond our scope to review all existing categories of methods, and we will restrict ourselves to surface based and space based methods which are mainly two types of the methods. As our contribution fits squarely in the surface based category, here we concentrate on surface based methods, although we mention some relevant space based approaches as well.

Early work of shape deformation approaches focus on space based methods, some surveys are well summarized in the literatures (Gain and Bechmann, 2008; Milliron *et al.*, 2002), some recent space methods are Jacobson *et al.* (2011) and Levi and Levin (2014), etc. Recently the high-resolution meshes which produced by 3D scanning devices of real-world objects become a prominent representation for 3D models. Having abundant local geometric details at various scales is one distinct property of these mesh. The surface based methods are particularly suited for this type representation of models and attract widespread attention of researchers. A common interactive mode is to specify a number of original and target vertices and compute the remaining vertex positions by a variational approach. The details preserving is the central goal of these approaches.

Multiresolution methods. Multiresolution methods preserve the geometric details by decomposing the mesh into a series of hierarchical simplified levels with different detail coefficients. The details are defined by differences successive levels and encoded to local frames of the lower level (Botsch and Kobbelt, 2004; Kobbelt *et al.*, 1998; Zorin *et al.*, 1997). These methods often apply deformation on low levels of the model then reconstruct the high level details.

Physics-based methods. An alternative approach is to develop algorithms based on some physics-based energies, e.g., by using continuum mechanics and elasticity theory (Bao *et al.*, 2005; Chao *et al.*, 2010; Hu *et al.*, 2004). For example, Chao *et al.* (2010) defined an elastic energy based on the distance between the deformation gradient and the rotation group, including an additional thin-shell bending term if necessary, and gave a nice analysis of the relationship between the as-rigid-as-possible methods and standard elastic energy minimization.

Differential representation methods. Using the differential representation based methods to preserve the details have gained significant popularity in recent years. Commonly used differential representations often extract different local geometric properties, such as Laplacian coordinates (Sorkine *et al.*, 2004), pyramid coordinates (Sheffer and Kraevoy, 2004), gradient fields (Yu *et al.*, 2004), local frames (Kircher and Garland, 2008; Lipman *et al.*, 2005; Paries *et al.*, 2007; Wang *et al.*, 2012), dual Laplacian coordinates (Au *et al.*, 2006), etc, see (Botsch and Sorkine, 2008) for a survey on these methods. The deformation technique that we derive from our discrete surface representation can be seen as differential coordinates methods. These methods are characterized by applying the change of local geometric properties to reconstruct the whole mesh by inputting some handle constraints.

Comment on metric. Our approach uses iteration to solve the main optimization problem, which is similar to local frames (Paries *et al.*, 2007), pyramid coordinates (Sheffer and Kraevoy, 2004) and dual Laplacian coordinates (Au *et al.*, 2006), but with a completely different differential representation which only consist of the angular metrics.

The edge lengths and edge angles have been used as the local shape descriptor in some literatures. Grinspun *et al.* (2003) used edge angles on some cloth energies in physically based simulation. Frohlich and Botsch (2011) and Winkler *et al.* (2010) proposed methods based on interpolating the edge lengths and edge angles, in some physics-based energies coupled with a non-linear reconstruction method. Recently, Wang *et al.* (2012) used edge lengths and edge angles to reconstruct the mesh only through a sparse linear solve, Zhang *et al.* (2012) used metric design through Ricci flow to deform surface, Cao *et al.* (2012) used L_p norms to shape deformation. We note that the absolute quantities such as edge lengths can hardly be preserved under the large-scale editing operations, in contrast we encode the relative values of angular measures and strive to preserve the local details in different scales.

Comment on deformation constraints. Lipman *et al.* (2005) and Wang *et al.* (2012) proposed linear and rotation-invariant method to handle rotational deformations, which based on fundamental theorem of surfaces in the continuous setting. However, linear method cannot avoid artifacts when rotational constrains are required, due to the fact that deformation energies should be invariant to rotation is a non-linear function of the shape geometry (Botsch and Sorkine, 2008). Attempts to linearize the rotational constraints show some problems, for example, the method is insensitive to translational constraints, the local frames may degenerate and become inconsistent with the reconstructed geometry (Paries *et al.*, 2007). The non-linear iterative method (Au *et al.*, 2006; Paries *et al.*, 2007; Sheffer and Kraevoy, 2004) can efficient cope of above problems, comparing to these methods our method generates less edge angles distortion.

Deformations with large shearing constraints may result in severe artifacts, here our method can efficient reduce angular distortion and preserve the local details as much as possible.

Conformal deformations permit efficient manipulate of surface with scaling constraints. The goal of our method is allowing scale invariance of each vertex 1-ring in the ℓ_2-sense. In 2D, our idea equals to the as-similar-as-possible method (Igarashi *et al.*, 2005) and relates to the conformal shape editing methods (Chen *et al.*, 2013; Lipman *et al.*, 2008; Weber and Gotsman, 2010). In 3D, the quasi-conformal shape editing methods (Crane *et al.*, 2011; Lipman, 2012; Paries *et al.*, 2007) can permit efficient manipulation of surface geometry up to scale. Our method strives to achieve the same goal, but in *sharp contrast*, we provide additional edge angles constraints. (Crane *et al.*, 2011) produced conformal deformed results by manipulating mean curvature and boundary data which can introduce less conformal distortion than our method, however changing the mean curvature will introduce significant edge angles distortion. (Paries *et al.*, 2007) is similar to our method, it seeks conformal deformations and encode local frames as quaternions at vertices to preserve geometric details, it restricts local frames to be orthogonal not orthonormal which can isotropic scale the shape, however they have not considered the edge angles constraints which can preserve the mean curvature of the surface and produces larger angular distortion than our method.

Essentially, our method preserves edge angles between adjacent triangles which corresponds to an implicit preservation of mean curvature (Grinspun *et al.*, 2003), preserves corner angles which corresponds to an implicit preservation of gauss curvature. And our differential representation based method is conceptually simple, which leads to an efficient algorithm.

3 Scale-invariant measures

Scale-invariant measures are designed to capture the shape of the mesh around each vertex and measure the set of angles uniquely relating a vertex to its 1-ring neighborhoods.

3.1 Notations and definitions

Denote a 2-manifold triangular mesh as $M = \{V, E, F\}$, where V denotes the set of its vertices, E

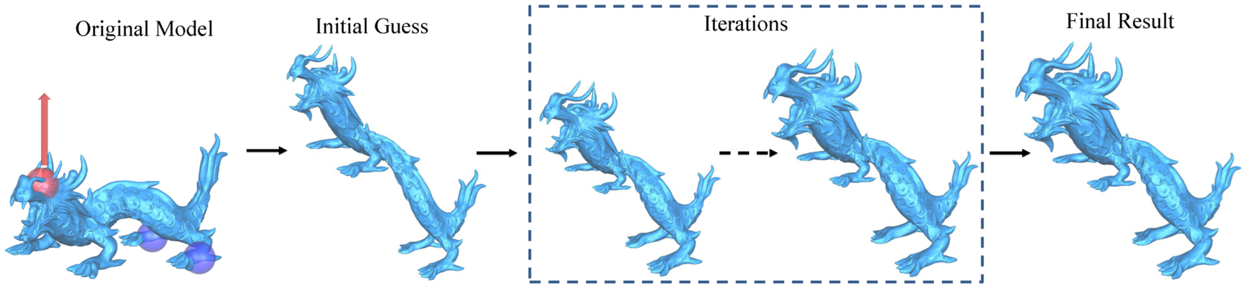


Fig. 2 Overview of our iterative algorithm. Given a Dragon model, the user fixes two points (in blue) on its legs and drags one point (in red) upwards. Starting from a severe distorted initial guess, our algorithm runs a two-step processing to update the mesh in an iterative manner. The final result is explicitly preserve the local geometric details as much as possible in different scales shown in the right.

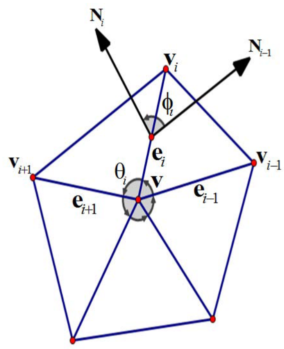


Fig. 1 The scale-invariant measures defined at the 1-ring cell of vertex v in a 3D triangular mesh.

denotes the set of its edges, and F denotes the set of its triangles. Let $v \in V$ (with $\mathbf{v}$ denoting its coordinates throughout the paper) be a vertex of M with valence k and $\mathbf{v_1}, \mathbf{v_2}, \cdots, \mathbf{v_k}$ be its 1-ring neighborhood vertices (Fig. 1). $C_v = \{\mathbf{v}, \mathbf{v_1}, \mathbf{v_2}, \cdots, \mathbf{v_k}\}$ is called the 1-ring cell of $\mathbf{v}$. Denote the edges $\mathbf{e}_i = \mathbf{v}\mathbf{v}_i$ and triangles $f_i = \Delta\mathbf{v}\mathbf{v}_i\mathbf{v}_{i+1}$, $i = 1, 2, \cdots, k$ (here $\mathbf{v}_{k+1} = \mathbf{v}_1$). We define the triangle angles $\theta_i = \angle(\mathbf{e}_i, \mathbf{e}_{i+1})$, and the edge angles $\phi_i = \angle(f_{i-1}, f_i)$.

Then $\Omega_v = \{\Theta_i, \Phi_i\}$, where $\Theta_i = \{\theta_i, i = 1, \cdots, k\}$, and $\Phi_i = \{\phi_i, i = 1, \cdots, k\}$, are defined as the scale-invariant measure of $\mathbf{v}$. We can see that the measures are invariant under rigid and isotropic scaling transformation.

We define *the scale-invariant measures* of the mesh M as $\Omega = \{\Omega_v, v \in V\}$. On one hand, Ω is uniquely defined for a given mesh M. On the other hand, given the scale-invariant measures Ω, if we fix one of the edges and one of the orientation of the triangle containing this edge, all the edges can be obtained from Ω. Then a mesh can be uniquely determined by edge lengths and edge angles as shown in (Wang *et al.*, 2012). It is thus seen that a triangular mesh M has its shape (and size) completely determined by its scale-invariant measures Ω up to a rigid transformation (and a specific scale). If the user inputs more handle constraints, the system is overconstrainted we solve it in least squares sense to reconstruct a deformed mesh according to the prescribed handle constraints.

3.2 Scale-aware equations

Let us discuss about how we can reconstruct the mesh M from Ω. Consider the triangle f_i of the cell C_v. Denote $\mathbf{N}_i$ as the normal of f_i. Then edge $\mathbf{e}_{i+1}$ can be obtained by rotating edge $\mathbf{e}_i$ by angle θ_i around the axis $\mathbf{N}_i$. That is, we have

$$\mathbf{e}_{i+1} = l_i \mathbf{R}_i \mathbf{e}_i, \tag{1}$$

where $\mathbf{R}_i$ represents the rotation matrix of rotating around $\mathbf{N}_i$ by angle θ_i which encodes rotation and l_i encodes the scaling from $\mathbf{e}_i$ to $\mathbf{e}_{i+1}$. In order to preserve the connectivity of the mesh, we define the the length ratios $l_i = \|\mathbf{e}_{i+1}\|/\|\mathbf{e}_i\| = \sin\alpha/\sin\beta$, where α is the opposite triangle angle of edge $\mathbf{e}_{i+1}$ and β is the opposite triangle angle of edge $\mathbf{e}_i$ due to the law of sines.

If $\mathbf{N}_i$ is known, we can have a linear equation on $\mathbf{v}, \mathbf{v}_i, \mathbf{v}_{i+1}$ from (1) as

$$(l_i\mathbf{R}_i - \mathbf{I})\mathbf{v} - l_i\mathbf{R}_i\mathbf{v}_i + \mathbf{v}_{i+1} = 0, \tag{2}$$

where $\mathbf{I}$ is the identity matrix. Apply this to all vertex angles $\theta \in \Theta$ yields a large sparse linear system

$$\mathbf{G}(\Theta, \mathcal{N}(\mathbf{V}))\mathbf{V} = 0, \tag{3}$$

where $\mathcal{N}$ denotes all the triangle normals of M. We call Equation (3) the *scale-aware equations* of M.

Moreover, as $\angle(\mathbf{N}_{i-1}, \mathbf{N}_i) = \phi_i$, apply this to all edges yields the following constraint equations on $\mathcal{N}$:

$$H(\Phi, \mathcal{N}(\mathbf{V})) = \{\angle(\mathbf{N}_e^l, \mathbf{N}_e^r) = \phi_e, e \in E\}, \qquad (4)$$

where $\mathbf{N}_e^l$ and $\mathbf{N}_e^r$ denote the normals of the two triangles sharing e and ϕ_e is the edge angle between two triangles. We call Equation (4) the *scale-aware constraints* of M.

Given Ω, the vertices $\mathbf{V}$ of M can be reconstructed by solving the scale-aware equations (3) under the scale-aware constraints (4). Note that $\mathbf{V}$ also appears in the coefficient matrix $\mathbf{G}$ in (3). Thus it is nontrivial to solve such a non-linear system with nonlinear constraints. We will propose a two-step method to reconstruct M from Ω in the next section.

4 Scale-Aware Shape Manipulation

Solving $\mathbf{V}$ from the scale-aware equations (3) is essentially a chicken-and-egg problem: on one hand, the reconstruction of the mesh requires the triangle normals; on the other hand, the computation of triangle normals depends on the mesh vertices. In order to solve this problem, we present a two-step iterative solution by separating the vertices $\mathbf{V}$ and the triangle normals $\mathcal{N}$ to compute $\mathbf{V}$ and $\mathcal{N}$ alternatively.

Vertex constraints. To accurately reconstruct the mesh we need to fix some vertices (at least three) so as to eliminate the extra degrees of freedom of the intrinsic equations. In the application of shape deformation, the user fixes some vertices of the mesh and manipulates some other vertices (called handles). The deformed mesh has to be reconstructed according to the user's inputs. We denote the input constraints of the user as

$$\mathbf{v}_j = \mathbf{v}_j{}^*, j \in C, \qquad (5)$$

where C is the set of fixed vertices and handles specified by the user.

4.1 Energy function

We solve the following minimization problem

$$\min_{\mathbf{V}} E(\mathbf{V}) = \|\mathbf{G}(\Theta, \mathcal{N}(\mathbf{V}))\mathbf{V}\|^2$$
$$\text{s.t.}\{(4),\ (5)\} \qquad (6)$$

Starting from an initial guess of the vertices, our algorithm alternatively updates the triangle normals $\mathcal{N}$ and the vertex positions $\mathbf{V}$ in two steps in an iterative manner (see Fig. 2).

Step one. In this step, we are given an intermediate $\mathbf{V}$. The normals $\mathcal{N}$ are computed accordingly. We then update $\mathcal{N}$ such that it satisfied the scale-aware constraints (4).

Step two. In this step, we fix the normals $\mathcal{N}$. Then the vertices $\mathbf{V}$ is solved by solving the sparse linear system (3) under constraints (5).

4.2 Step one: Updating normals

Suppose we have an initial guess which is generated by any available algorithm. Then we can compute its normal $\bar{\mathcal{N}}$ explicitly. However, $\bar{\mathcal{N}}$ may not satisfy the constraints (4). Thus we need to update $\bar{\mathcal{N}}$ to satisfy (4).

Consider the Gaussian map $\tilde{G}$, a graph on the unit sphere with each node representing the normal of a triangle, of M. Actually, $\tilde{G}$ is a dual mesh on the unit sphere of M (Fig. 3(a), (b)).

Now the problem is how to adjust the normals under the constraints (4). That is, given a set of nodes on the unit sphere, how to adjust the positions of the nodes such that the angles (distances) between each node and its adjacent nodes are equal to the given edge angles Φ. We develop an iterative scheme to update the normals in an as-rigid-as-possible manner (Sorkine and Alexa, 2007).

Denote $N_v = \{\mathbf{N}_1, \mathbf{N}_2, \cdots, \mathbf{N}_k\}$ as the triangle normals of the 1-ring neighbors of vertex $\mathbf{v}$. The pyramid consists of N_v and the sphere origin $\mathbf{O}$ (with edges $\tilde{\mathbf{e}}_i = \mathbf{ON}_i$, $i = 1, 2, \cdots, k$) is denoted by $\tilde{P}_v$. Denote M' as the deformed mesh of M. The counterpart X of M in M' are denoted by X' in this section.

Local phase. For each pyramid $\tilde{P}_v$, the optimal rotation $\tilde{\mathbf{R}}_v$ from $\tilde{P}_v$ to $\tilde{P}'_v$ is computed by minimizing the following energy function

$$E_{\tilde{P}_v} = \sum_{i=1}^{k} w_i \|\tilde{\mathbf{e}}'_i - \tilde{\mathbf{R}}_v \tilde{\mathbf{e}}_i\|^2, \qquad (7)$$

where w_i are the cotangent weights for spatially consistent on non-uniform meshes. This is solved as $\tilde{\mathbf{R}}_v = \mathbf{V}_v \mathbf{U}_v^T$ by the signed SVD factorization of the

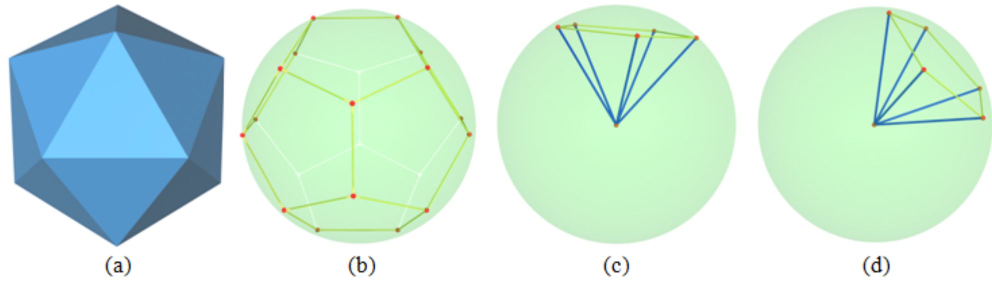


Fig. 3 Illustration of the triangular mesh (a) and its dual mesh on the unit sphere (b). (c) shows a pyramid consisting of nodes of a face in (b) and the origin of the sphere. The pyramid is deformed into another pyramid (d) with a rotation transformation which can be computed in the local phase of updating the normals in our algorithm.

cross-covariance matrix (Sorkine and Alexa, 2007).

$$\mathbf{S}_v = \sum_{i=1}^{k} w_i \tilde{\mathbf{e}}_i \tilde{\mathbf{e}}_i'^T = \mathbf{U}_v \mathbf{D}_v \mathbf{V}_v^T. \tag{8}$$

Global phase. After we apply the rotation $\tilde{\mathbf{R}}_v$ to each pyramid $\tilde{P}_v$ individually, the corresponding nodes of $\tilde{G}$ may not coincide as shown in Fig. 4 (left). We then stitch them by minimizing the following energy function

$$E_{\tilde{G}} = \sum_{v \in V} E_{\tilde{P}_v}. \tag{9}$$

Its derivative is

$$\frac{\partial E_{\tilde{G}}}{\partial \tilde{\mathbf{e}}_i'} = \|\tilde{\mathbf{e}}_i' - \tilde{\mathbf{R}}_{v_1}\tilde{\mathbf{e}}_i\|^2 + \|\tilde{\mathbf{e}}_i' - \tilde{\mathbf{R}}_{v_2}\tilde{\mathbf{e}}_i\|^2 + \|\tilde{\mathbf{e}}_i' - \tilde{\mathbf{R}}_{v_3}\tilde{\mathbf{e}}_i\|^2, \tag{10}$$

where v_1, v_2, v_3 are the adjacent nodes of node $\mathbf{N}_i$ (i.e., $\mathbf{e}_i$) in $\tilde{G}$. By setting $\frac{\partial E_{\tilde{G}}}{\partial \tilde{\mathbf{e}}_i'} = 0$ we have

$$\mathbf{N}_i' = \frac{1}{3}(\tilde{\mathbf{R}}_{v_1}\mathbf{N}_i + \tilde{\mathbf{R}}_{v_2}\mathbf{N}_i + \tilde{\mathbf{R}}_{v_3}\mathbf{N}_i). \tag{11}$$

$\mathbf{N}_i'$ is then normalized on the unit sphere.

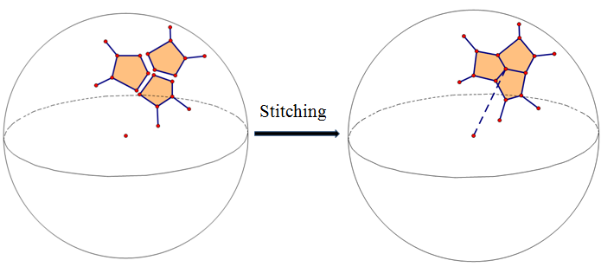


Fig. 4 Stitching the separate pyramids into a coherent mesh on the unit sphere.

4.3 Step two: Recovering vertices from normals

After $\mathcal{N}$ is updated, we fix $\mathcal{N}$ and recover $\mathbf{V}$ by solving the linear system $\mathbf{G}(\mathcal{L}, \Theta, \mathcal{N})\mathbf{V} = 0$ with constraints (5). The constraints are regarded as soft constraints and we solve $\mathbf{V}$ by the following minimization problem

$$\min_{\mathbf{V}} E(\mathbf{V}) = \|\mathbf{G}\mathbf{V}\|^2 + \lambda \sum_{j \in C} \|\mathbf{v}_j - \mathbf{v}_j{}^*\|^2, \tag{12}$$

where λ is a weight (we set $\lambda = 100.0$ in our implementation).

4.4 Algorithm

Our algorithm is summarized in Algorithm 1. As the deformed mesh is constructed by preserving the scale-invariant measures. That is, the result preserves the local similarity with the input mesh as much as possible. The optimal local scales are automatically obtained in different parts of the mesh.

Advantages over previous methods. Our scale-aware approach can preserve the shape of local details efficiently and automatically determine the scale factors of the local details everywhere. When users stretch or squash the object in *large scales*, previous methods *cannot* preserve the local shape because those methods try to preserve the *absolute* measure of the length (rigidity) whereas our method try to preserve the *relative* length similarity. Comparing the previous surface based conformal deformation method (Paries *et al.*, 2007) our method has a smaller angular distortion and better details preserved results (see Section 6), comparing the frame based methods (Kircher and Garland, 2008; Lipman *et al.*, 2005; Paries *et al.*, 2007; Wang *et al.*, 2012)

Algorithm 1 Mesh reconstruction from the scale-aware variables

Require: The scale-aware variables Ω
Ensure: The mesh vertices $\mathbf{V}$
Initialization: Generate an initial guess $\mathbf{V}$.
Repeat until convergence {
Step one:
Repeat until convergence {
$\mathcal{N} \longleftarrow \mathbf{V}$;
Update $\mathcal{N}$ under constraints (4) (Section 4.2) }
Step two:
$\mathbf{V} \longleftarrow$ Solve (3) under constraints (5) (Section 4.3) }

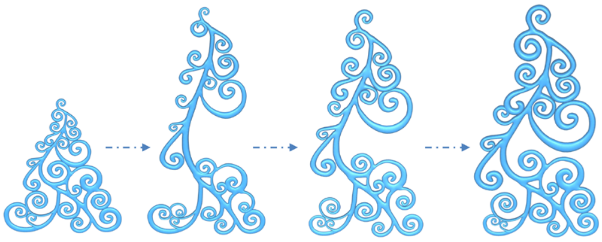


Fig. 5 The results of the iterations using our algorithm. From left to right: the input model; the intermediate result; the intermediate result; the final result. Carefully observe the difference in the branches, the latter ones better keep the similarity of the local shapes in the original model.

which require manual scaling some discrete frames our scale factors are obtained automatically and our method has a smaller distortion.

5 Discussion

We discuss about some issues of our algorithm in this section.

5.1 Convergence of the iterative algorithm

Fig. 5 shows the results of the iterations using our algorithm. We can see that the local details are better preserved as the algorithm is applied in more iterations. Furthermore, we also quantitatively evaluate the convergence of our algorithm.

Error metric. Our algorithm tries to preserve the scale-invariant measures Ω. To quantitatively evaluate our algorithm we respectively analyze the error metric on the angles as follows.

$$E_\theta = \sqrt{\sum_{\theta\in\Theta}(\theta_{org}-\theta_{res})^2/|\Theta|},$$

$$E_\phi = \sqrt{\sum_{\phi\in\Phi}(\phi_{org}-\phi_{res})^2/|\Phi|},$$

where θ_{org} and ϕ_{org} denote the original angles respectively, and θ_{res} and ϕ_{res} denote the result angles respectively.

Quantitative evaluation. We show the error curves of the example in Fig. 11(f). Fig. 6(a) shows the error curve of the triangle angles E_θ. It is seen that the curve of E_θ decreases quickly and thus converges. Fig. 6(b) shows the error curve of the edge angles E_ϕ. In Fig. 6(b), we can see that error curve of E_ϕ always decreasing in step one (normal updating step), and it might lift up in step two (vertex updating step), finally the curve converges to a stable status.

The convergence of the algorithm seriously impact the final deformed result. In Fig. 5, we find that the final convergence result better preserve the local shapes in the branches of the tree model than the intermediate results. Although it is difficult to formally proof convergence of the above proposed algorithm, we observed in our experiments (Fig. 6) that the reconstruction converges after only a few iterations even for large handle transformations. Thus we prove the convergence of the algorithm via numerical experiment.

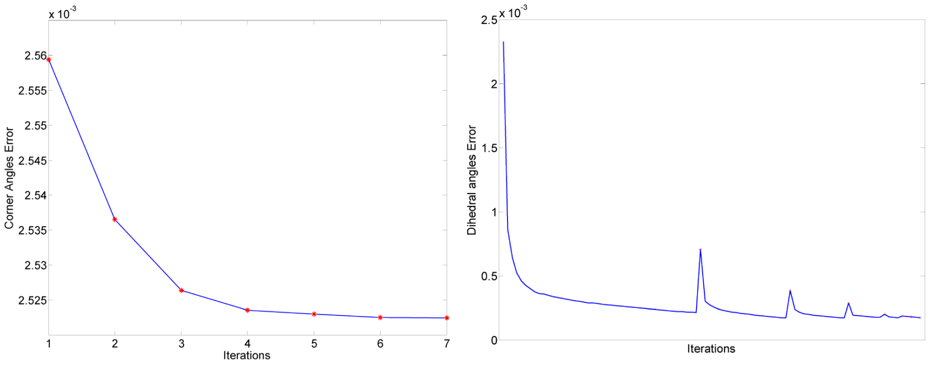
5.2 Initialization

Our algorithm is insensitive to the choose of initialization. In our implementation, we all adopt the original Laplacian editing method Sorkine *et al.* (2004) as the initialization which should produce severe distortion in rotation, shearing and scaling transformations. However, our algorithm can produce visually pleasing results from the highly distorted initialization even for large handle transformations.

5.3 Optimal rigidity approximation

Our scale-aware approach can well preserve the local shape by scaling the local details in an optimal manner. However in some applications, the user want to keep the rigidity of the local detail. Starting from the result obtained by Algorithm 1, where the shapes of the triangles are well preserved, the rigidity of the triangles as well as its 1-ring neighbors can be easily estimated.

We extend our algorithm, called optimal rigidity



(a) (b)

Fig. 6 The curves of angle errors of the example in Fig. 11(f). (a) shows the error curve of triangle angles (corner angles of triangles); (b) shows the error curve of edge angles (dihedral angles of edges).

approximation (ORA), to satisfy the requirements of preserving the rigidity in the results. Specifically, we scale each individual triangle to make it have the same size with the original one. Then we stitch all the scaled triangles into one coherent mesh (Igarashi *et al.*, 2005).

Note that the ORA results will increase some distortion of the edge angles and triangle angles, but producing a global rigidity results.

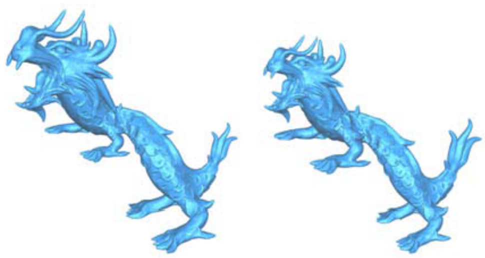


Fig. 7 The results of shape manipulation using our scale-aware manipulation method (left) and our optimal rigidity approximation method (right).

Scale factor. We denote the area of triangle $t \in F$ as $Area_t$. The scale factor of t is defined as

$$\sigma_t = \frac{Area'_t}{Area_t}. \tag{13}$$

Scaling each triangle. We scale each triangle $t' = \{\mathbf{v}'^t_0, \mathbf{v}'^t_1, \mathbf{v}'^t_2\}$ according to its center $\mathbf{c}'_t = \frac{1}{3}(\mathbf{v}'^t_0 + \mathbf{v}'^t_1 + \mathbf{v}'^t_2)$ to a new triangle $\hat{t}_t = \{\hat{\mathbf{v}}^t_0, \hat{\mathbf{v}}^t_1, \hat{\mathbf{v}}^t_2\}$ as (see Fig. 8)

$$\hat{\mathbf{v}}^t_j = \mathbf{c}'_t + \tau_t \mathbf{c}'_t \mathbf{v}'^t_j, j = 0, 1, 2, \tag{14}$$

where $\tau_t = \sqrt{1/\sigma_t} = \sqrt{Area_t/Area'_t}$.

Stitching the triangles. We stitch all the triangles into a coherent mesh by minimizing the following

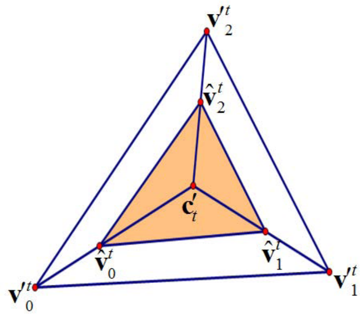


Fig. 8 Construction of an adjusted triangle from a deformed triangle by a scale factor to make the area of adjusted triangle equal to the original triangle.

energy

$$E_{stitch} = \sum_{t \in F} \sum_{j=0}^{2} \|\check{\mathbf{v}}^t_j \check{\mathbf{v}}^t_{j+1} - \hat{\mathbf{v}}^t_j \hat{\mathbf{v}}^t_{j+1}\|^2. \tag{15}$$

The above stitching scheme is actually an extension of that presented in (Igarashi *et al.*, 2005) from 2D mesh to 3D mesh. As Algorithm 1 obtains similar shapes of the triangles with the input mesh, the above scaling and stitching scheme reveals the rigidities of all triangles very well, as shown in Fig. 7.

6 Experimental results

Illustration of 2D curve editing. Fig. 9 compares the editing results of a 2D Koch curve using various deformation approaches, including the original Laplacian editing approach (Sorkine *et al.*, 2004), the optimized Laplacian editing approach (Sorkine *et al.*, 2004), the LRI approach (Lipman *et al.*, 2005), the ARAP approach (Sorkine and Alexa, 2007) and our approach. Large distortions in the local details ap-

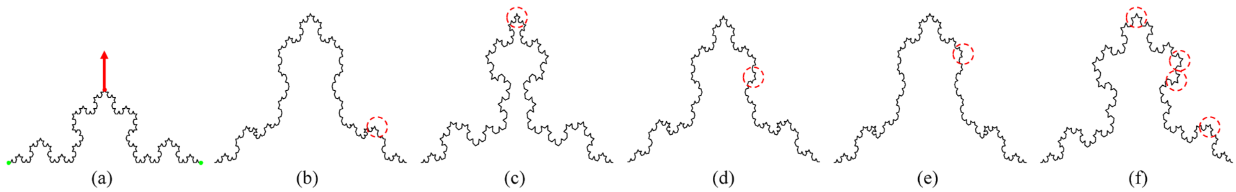


Fig. 9 Results of editing the 2D Koch curve (a) (the two ends in green are fixed and the middle point in red is pulled upwards) using different approaches: (b) the original Laplacian-based approach (Sorkine *et al.*, 2004); (c) the implicit optimization Laplacian-based approach (Sorkine *et al.*, 2004); (d) the linear rotation-invariant (LRI) approach (Lipman *et al.*, 2005); (e) the as-rigid-as-possible (ARAP) approach (Sorkine and Alexa, 2007); (f) our scale-aware approach.

pear in the results as shown in Fig. 9(b-e). On the contrary, our method faithfully preserves the details as much as possible to the original ones with different scales even including the sharp features (Fig. 9(f)), which showing the characteristic of our algorithm.

Mesh deformation. We compare our algorithm with other deformation methods in 3D deformation. To demonstrate the quality of the deformed results by our approach, we applied the test examples from (Botsch and Sorkine, 2008). The extreme examples were chosen to reveal limitations of linear approaches, i.e. existing linear methods show gross artifacts on at least one of these examples.

From the Fig. 10, we can see that our approach produces plausible results for all these examples and does not suffer linearization artifacts. The quality of our results can compare to any non-linear approaches, such as Primo (Botsch *et al.*, 2006), ARAP (Sorkine and Alexa, 2007) and Consistent Local Frames (Paries *et al.*, 2007), etc.

Fig. 11 shows the deformation results when we stretch the Xmas tree model using different approaches. From the results, we find that our algorithm appropriately scales the local features to satisfy the input constraints of user without obvious distortion. There are abundant of methods focus on rotation invariance only, but there are couple methods that address the similar problem as ours. In order to demonstrate the advantage of our representation, we compare with the conformal deformation method (Paries *et al.*, 2007) which is quaternion based. Comparing Fig. 11(e) and Fig. 11(f) we can find that our method produced a better shape preservation and a less distortion of edge angles result than the conformal deformation method (Paries *et al.*, 2007).

As is evident in Fig. 10 and Fig. 11, our scale-aware method consistently gives the best value of edge angle distortion (E_ϕ), at a very small even insignificant penalty in triangle angle distortion (E_θ).

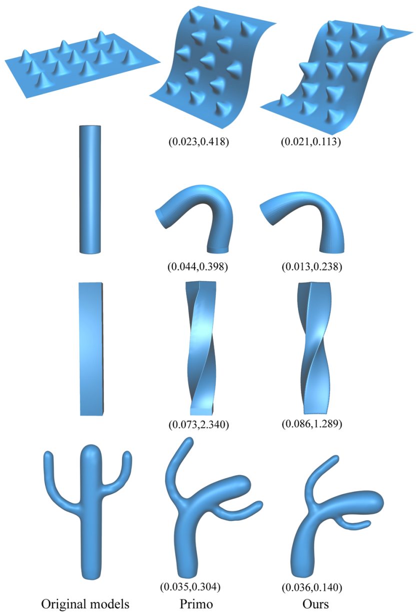


Fig. 10 Our results (right) compare to the results of Primo (Botsch *et al.*, 2006) (middle) for a pure translation, a $120°$ bend, a $135°$ twist and a $70°$ bend of different objects, which are some extreme examples shown in (Botsch and Sorkine, 2008). Numbers in brackets denote triangle angle distortion and edge angle distortion.

Our algorithm can be used to expand the focal region and reconstruct the deformed result. Fig. 12 shows the results of magnifying the local region by different scales. Fig. 13 shows the results of magnifying different local regions of the same scale.

The results show that the details of the local shapes can be well preserved in different scales.

Fig. 14 shows more results of deforming a Dinosaur model using our method.

Implementation details. We have implement-

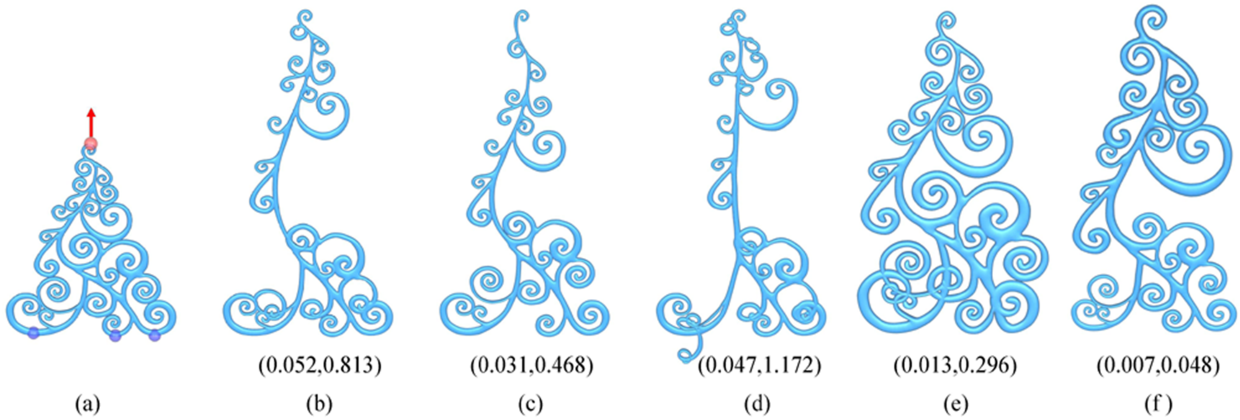


Fig. 11 Results of manipulating the Xmas tree model (a) (the three points in blue are fixed and the point in red is pulled upwards) using different approaches: (b) the Laplacian editing approach (Sorkine *et al.*, 2004); (c) the linear rotation-invariant (LRI) approach (Lipman *et al.*, 2005); (d) the as-rigid-as-possible (ARAP) approach (Sorkine and Alexa, 2007); (e) the consistent local frames approach (Paries *et al.*, 2007); (f) our scale-aware approach. Numbers in brackets denote triangle angle distortion and edge angle distortion. Our algorithm is more effective to preserve the local features in different scales than the other algorithms under large-scale edit operation.

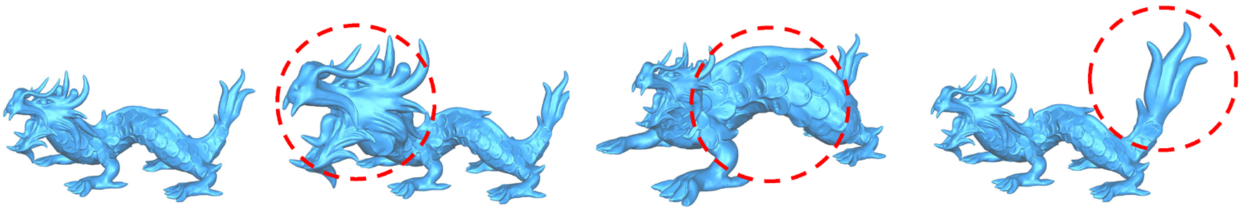


Fig. 13 Results of magnify the different parts of the Dragon model with the same scale. From left to right: original dragon model; scaling(×3) the head; scaling(×3) the body; scaling(×3) the tail.

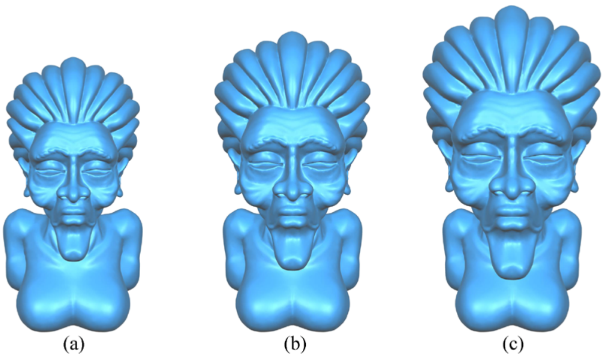


Fig. 12 Results of magnify the same part of the Elder model with different scales. Left: original model; middle: scaling the head by a scale of 1.5; right: scaling the head by a scale of 2.

ed our algorithm using C++ on an Intel dual core 2.10/2.10 GHZ processor notebook with 4GB RAM. In step one, we use a iterative local/global method to update the normals, it usually takes 5-20 iterations to converge, thus it is very fast and the runtime is comparable to the ARAP (Sorkine and Alexa, 2007) method. The most time-consuming part of our algorithm is solving the sparse linear system (3) in step two. The factorization of the normal equation may take a longer time, but it is precomputed only once. At each iteration step, only back-substitutions are performed to solve the system. We used the sparse Cholesky linear solver for matrix computation. In all our experiments, our algorithm usually takes 10-20 iterations to converge. Due to the two-step iterative process, our algorithm need more running time than the linear variational deformation methods such as Laplacian editing method (Sorkine *et al.*, 2004), linear rotation-invariant method (Lipman *et al.*, 2005), etc. However the angular measure errors are very low in all the case of this paper, which demonstrates the efficiency and accuracy of our algorithm. Table 1 gives the model statistic, the time required and the error estimation of the deformed results which measured when iteration converge in this paper.

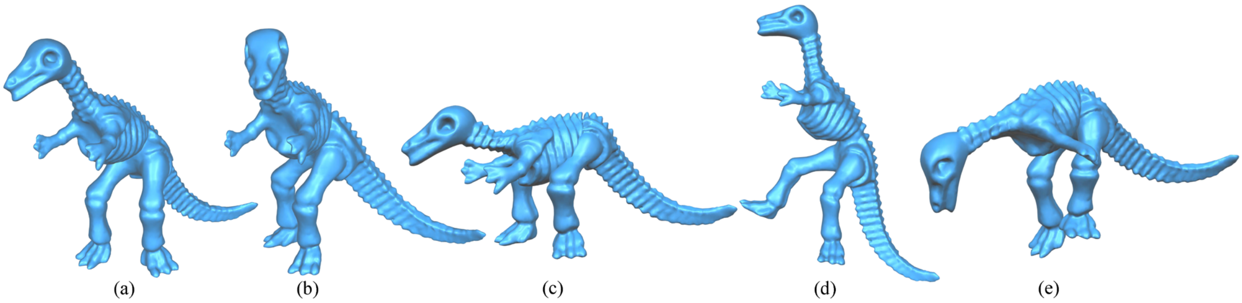


Fig. 14 Results of editing the Dino model (a). (b-e) are results produced by our algorithm.

Table 1 Model statistic and performance

Model	n	Runtime(s)	E_θ	E_ϕ
Bumpy plane	40401	76	0.021	0.113
Cylinder	4802	8	0.013	0.238
Bar	2602	6	0.086	1.289
Cactus	5261	11	0.036	0.140
Xmas tree	8582	24	0.007	0.048
Dragon	19974	42(a)	0.022(a)	0.113(a)
Dino	5420	13(a)	0.033(a)	0.311(a)
Elder	12500	37(a)	0.004(a)	0.032(a)

n the number of vertices; s the convergence time in seconds; a the average of all the examples

7 Conclusion

In this paper, we introduce a new set of scale-invariant measures for representing the 3D triangular mesh. The measures are invariant to rigid and isotropic scale transformations. Then we present a robust and efficient approach to edit the shape based the measures. The shape of the local details are well preserved in the deformed results. The experimental results show that our algorithm is an effective tool for manipulating shapes and can generate visually pleasing results even under severe editing operations.

As a future work, we will speed up the algorithm using GPU implementation. It is also worthwhile to apply our scale-invariant measures in other applications such as surface morphing, detail enhancement, coating, etc.

References

Au, K-C, Tai, C-L., Liu, L.G., H.B.Fu, 2006. dual laplacian editing for meshes. *Visualization and Computer Graphics, IEEE Transactions on*, **12**(3):386-395. [doi:10.1109/TVCG.2006.47]

Bao, Y.F., Guo, X.H., Qin, H., 2005. physically based morphing of point-sampled surfaces. *Computer Animation and Virtual Worlds*, **16**:509-518. [doi:10.1002/cav.100]

Botsch, M., Kobbelt, L., 2004. an intuitive framework for real-time freeform modeling. *ACM Trans. Graph.*, **23**(3):630-634. [doi:10.1145/1015706.1015772]

Botsch, M., Pauly, M., Gross, M., Kobbelt, L., 2006. Primo: coupled prisms for intuitive surface modeling. Proceedings of the fourth Eurographics symposium on Geometry processing, Aire-la-Ville, Switzerland, Switzerland, p.11-20.

Botsch, M., Sorkine, O., 2008. on linear variational surface deformation methods. *IEEE Transactions on Visualization and Computer Graphics*, **14**(1):213-230. [doi:10.1109/TVCG.2007.1054]

Cao, L., Zhang, G.X., Lai, Y.K., 2012. l_p shape deformation. *Science China Information Sciences*, **55**(5):983-993. [doi:10.1007/s11432-012-4574-y]

Chao, I., Pinkall, U., Sanan, P., Schröder, P., 2010. a simple geometric model for elastic deformations. *ACM Trans. Graph.*, **29**(4):38:1-38:6. [doi:10.1145/1778765.1778775]

Chen, R.J., Weber, O., Keren, D., Ben-Chen, M., 2013. planar shape interpolation with bounded distortion. *ACM Trans. Graph.*, **32**(4):108:1–108:12. [doi:10.1145/2461912.2461983]

Crane, K., Pinkall, U., Schröder, P., 2011. spin transformations of discrete surfaces. *ACM Trans. Graph.*, **30**(4):104:1-104:10. [doi:10.1145/2010324.1964999]

Frohlich, S., Botsch, M., 2011. example-driven deformations based on discrete shells. *Computer Graphics Forum*, **30**(8):2246–2257. [doi:10.1111/j.1467-8659.2011.01974.x]

Gain, J., Bechmann, D., 2008. a survey of spatial deformation from a user-centered perspective. *ACM Trans. Graph.*, **27**(4):107:1-107:21. [doi:10.1145/1409625.1409629]

Grinspun, E., Hirani, A.N., Desbrun, M., Schröder, P., 2003. Discrete shells. Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation, Aire-la-Ville, Switzerland, Switzerland, p.62-67.

Hu, S.M., Li, C.F, Zhang, H., 2004. Actual morphing: a physics-based approach to blending. Proceedings of the ninth ACM symposium on Solid modeling and applications, Aire-la-Ville, Switzerland, Switzerland, p.309-314.

Igarashi, T., Moscovich, T., Hughes, J.F., 2005. as-rigid-as-possible shape manipulation. *ACM Trans. Graph.*, **24**(3):1134-1141. [doi:10.1145/1073204.1073323]

Jacobson, A., Baran, I., Popovic, J., Sorkine, O., 2011. bounded biharmonic weights for real-time deformation. *ACM Trans. Graph.*, **30**(4):1-8. [doi:10.1145/2010324.1964973]

Kircher, S., Garland, M., 2008. free-form motion processing. *ACM Trans. Graph.*, **27**(2):1-13. [doi:10.1145/1356682.1356685]

Kobbelt, L., Campagna, S., Vorsatz, J., Seidel, H-P., 1998. Interactive multi-resolution modeling on arbitrary meshes. Proceedings of the 25th annual conference on Computer graphics and interactive techniques, New York, NY, USA, p.105-114. [doi:10.1145/280814.280831]

Levi, Z., Levin, D., 2014. shape deformation via interior rbf. *IEEE Transactions on Visualization and Computer Graphics*, **20**(7):1062-1075. [doi:10.1109/TVCG.2013.255]

Lipman, Y., 2012. bounded distortion mapping spaces for triangular meshes. *ACM Trans. Graph.*, **31**(4):108:1–108:13. [doi:10.1145/2185520.2185604]

Lipman, Y., Levin, D., Cohen-Or, D., 2008. green coordinates. *ACM Trans. Graph.*, **27**(3):78:1–78:10. [doi:10.1145/1360612.1360677]

Lipman, Y., Sorkine, O., Levin, D., Cohen-Or, D., 2005. linear rotation-invariant coordinates for meshes. *ACM Trans. Graph.*, **24**(3):479-487. [doi:10.1145/1073204.1073217]

Milliron, T., Jensen, R.J., Barzel, R., Finkelstein, A., 2002. a framework for geometric warps and deformations. *ACM Trans. Graph.*, **21**(1):20-51. [doi:10.1145/504789.504791]

Paries, N., Degener, P., Klein, R., 2007. Simple and efficient mesh editing with consistent local frames. Proceedings of the 15th Pacific Conference on Computer Graphics and Applications, Los Alamitos, CA, USA, **0**:461-464. [doi:http://doi.ieeecomputersociety.org/10.1109/PG.2007.43]

Sheffer, A., Kraevoy, V., 2004. pyramid coordinates for morphing and deformation. :68-75. [doi:10.1109/3DPVT.2004.99]

Sorkine, O., Alexa, M., 2007. As-rigid-as-possible surface modeling. Proceedings of the Fifth Eurographics Symposium on Geometry Processing, Aire-la-Ville, Switzerland, Switzerland, p.109-116.

Sorkine, O., Cohen-Or, D., Lipman, Y., Alexa, M., Rössl, C., Seidel, H.-P., 2004. laplacian surface editing. :175-184. [doi:10.1145/1057432.1057456]

Wang, Y., , Liu, B., Tong, Y., 2012. linear surface reconstruction from discrete fundamental forms on triangle meshes. *Computer Graphics Forum*, **31**(8):2277-2287. [doi:10.1111/j.1467-8659.2012.03153.x]

Weber, O., Gotsman, C., 2010. controllable conformal maps for shape deformation and interpolation. *ACM Trans. Graph.*, **29**:78:1–78:11. [doi:10.1145/1778765.1778815]

Winkler, T., Drieseberg, J., Alexa, M., Hormann, K., 2010. multi-scale geometry interpolation. *Computer Graphics Forum*, **29**(2):309-318. [doi:10.1111/j.1467-8659.2009.01600.x]

Yu, Y.Z., Zhou, K., Xu, D., Shi, X.H., Bao, H.J., Guo, B.N., Shum, H.-Y., 2004. mesh editing with poisson-based gradient field manipulation. *ACM Trans. Graph.*, **23**(3):644-651. [doi:10.1145/1015706.1015774]

Zhang, M., Wei, Z., Xin, S.Q, Gu, X.F., 2012. stable geodesic surface signatures. *Tsinghua Science and Technology*, **17**(4):471-480. [doi:10.1109/TST.2012.6297593]

Zorin, D., Schröder, P., Sweldens, W., 1997. Interactive multiresolution mesh editing. Proceedings of the 24th annual conference on Computer graphics and interactive techniques, New York, NY, USA, p.259-268. [doi:10.1145/258734.258863]