

Continuous Deformations by Isometry Preserving Shape Integration

Janick Martinez Esturo, Christian Rössl, and Holger Theisel

Visual Computing Group, University of Magdeburg, Germany

Abstract. We introduce a novel continuous surface deformation method which relies on a time-dependent vector field over a triangular mesh. For every time step the piecewise linear vector field is obtained by least-squares minimization of the metric distortion induced by integration subject to boundary conditions. As an integral part of the approach, we introduce a new measure to describe local metric distortion which is invariant to the particular triangulation of the surface and which can incorporate smoothness of the field. Neither of these properties are met by previous work. A GPU implementation of the proposed algorithm enables fast deformations. The resulting deformations have lower metric distortions than deformations by existing (linear or non-linear) methods. This is shown for a number of representative test data sets.

Keywords: Shape Deformation, Isometry, Vector Field

1 Introduction

Shape deformations constitute a standard problem in modeling and computer graphics. A variety of approaches have been proposed in the recent decade, and shape deformation is still an active area of research. A good approach to deformations should be intuitive, visually convincing, geometrically or/and physically sound, and reasonably fast.

We consider shapes represented as triangle meshes. Most current approaches define deformations as a minimization problem. Given certain boundary conditions, the unknown vertex positions have to be determined. This yields a linear or non-linear optimization problem depending on the measure to be minimized. Proceeding this way, only the final positions of the vertices are computed without considering the deformation path leading to the final state. We call this kind of deformations *discrete deformations*. In contrast, *continuous deformations* integrate vertex positions along smooth vector fields. Here, the boundary conditions of the deformation are paths of parts of the shape. In this sense, we are aware of only two approaches to continuous deformations so far: [14] and its extension [15] which applies divergence-free vector fields for volume preserving shape deformation, and [18] which considers the deformations in a shape space.

Various measures have been proposed for minimization in discrete shape deformations. These measures typically reflect or simulate physical properties like

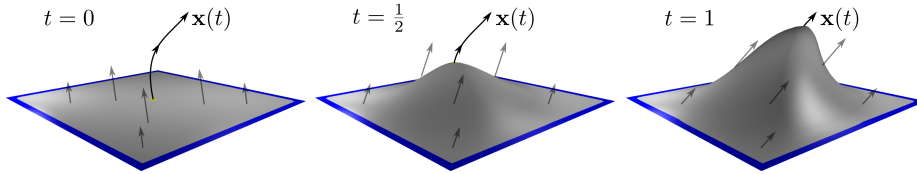


Fig. 1: Optimization Problem at three Time Steps of the Integration. The small handle region is marked yellow while the blue border corresponds to the fixed surface part. For every integration step, the current vertex positions and black vectors are given, while the gray vectors are obtained by solving a sparse linear system.

bending energies which may be simplified or linearized. Only recently, minimization of metric distortion has been considered [18, 17].

In this paper we present a new approach to continuous deformations which minimizes distortion of the surface and tries to preserve isometry. We define the deformation continuously over time, i.e., as a vector field which is determined in every time step as the result of a linear problem. The arising system matrices are sparse and can be solved reliably and efficiently supported by the GPU, which renders our approach reasonably fast. Figure 1 illustrates the main idea of our approach: following the standard deformation metaphor [4, 24], we define regions of full and zero deformation of the shape. For regions of full deformation, the deformation is given as parametric curves. Then in every time step a piecewise linear vector field is constructed such that metric distortion is minimized under integration. Boundary conditions are defined by the tangent vectors of the parametric curves describing the full deformation.

In summary, the main contributions of this work are: (1) design of a new discrete isometry measure which is invariant to surface tessellation and which extends naturally to incorporate smoothness. (2) Definition of continuous, isometry preserving shape deformations which are constrained by trajectories of surface regions. The whole approach is completely geometry-driven.

The remainder of the paper is organized as follows. Section 2 reviews related work. Section 3 introduces our approach and defines all measures while Section 4 discusses how the theoretic concepts are applied and implemented. Results are presented in Section 5 followed by a discussion of the method in Section 6 and final conclusions (Section 7).

2 Related work

There is a vast amount of literature on shape deformation, a proper review is far beyond the scope of this section. Instead, we point to the recent survey of Botsch and Sorkine [6]: they discuss and compare the most important classes of explicit surface deformation methods. In fact, we use their reference deformations as benchmarks. All reviewed methods implement discrete deformations which are

completely determined by boundary constraints, i.e., the placement of surface regions, fixed or handle regions, in 3-space. Deformation of the surface is then modeled in one of several ways:

- as a variational problem minimizing an energy functional which penalizes certain bending energies (see, e.g., [4]),
- as reconstruction from any kind of differential coordinates (see, e.g., [19]),
- as a projection or Poisson reconstruction after application of a "transformation field" to individual triangles thus over-determining vertex positions (see, e.g., [27]), or
- as simulation of forces to rigid and loosely coupled prism elements enveloping the surface [5].

Several methods are closely related, and they all share the goals of feature preservation and establishing smooth transitions towards deformed regions. All of these methods (except [5]) rely on the factorization of few or even only one single linear systems, a fact that renders these methods interactive. In particular, movement of handles requires only back-substitution for solving the system.

Furthermore, there is a variety of methods which determine a piecewise deformation where individual pieces are close to rigid transformations, i.e, they ought to be as rigid as possible. We refer to recent work by Sumner et al. [25] (and the references therein): here, a free-form deformation is determined based on a user provided "deformation graph", which defines the piecewise deformation. A non-linear minimization determines the degrees of freedom for the individual transformation pieces associated with the nodes. We note that our goal is not to obtain an as-rigid-as possible deformation but to stay as-isometric-as possible, and we refer to [18] for a more elaborate discussion. Furthermore our approach does not define a space warp but an explicit surface deformation (evaluation of our guiding vector fields is meaningful only for surface points), and we explicitly include the surface metric. Of course this review cannot be complete as there is a vast amount of literature on shape deformation. Among other non-linear methods we mention [3, 16, 26] as recent examples.

For all the above mentioned methods, deformation remains a *discrete* process: it is solely the rest position of the handles that determines the result but not their particular trajectory.

In contrast, von Funck et al. [14] introduce an approach based on integration of a surface along a time-dependent vector field. Their goal is volume preserving deformations. This is achieved in an elegant way: the guiding vector fields are constructed to show zero divergence. Remarkable in the context of our work is that this constitutes a *continuous* deformation which does depend on the particular trajectories of the handles. (Note that otherwise this method is fundamentally different to ours.) The method was extended in [15] by the introduction of deformation paths. An earlier approach [2] related to this does not rely on continuous vector fields but instead discretizes the deformation.

Kilian et al. [18] regard continuous deformations in a shape space: they solve a boundary value problem in order to find a time-dependent deformation between two poses of the same shape. They advocate to determine optimal de-

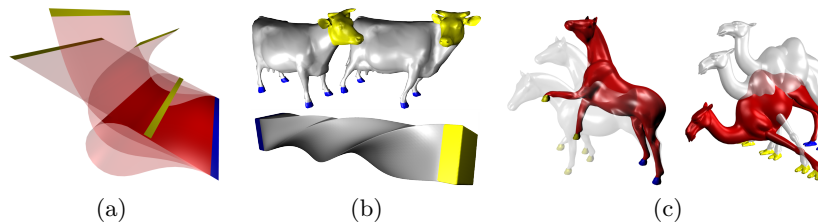


Fig. 2: Deformation examples. (a) Perfectly isometric deformation of developable plane. (b) Twisting deformation of the head of the cow and a strong twist of a bar by 280° which is not achievable by discrete deformations. (c) Continuous deformations using multiple handles.

formations by isometry preservation rather than a more traditional as-rigid-as-possible [1] criterion. Then a discrete version of Killing fields [8] characterizes time-dependent deformation vector fields. While the focus is on boundary value problems and their solution by a space-time multigrid approach, initial value problems are discussed briefly. In both cases, the approach is not interactive.

We consider the latter deformation approaches by Funck et al. [14] and Kilian et al. [18] as *continuous* deformation methods. Discrete methods mentioned above rely on the minimization of certain (potentially linearized) energy functionals which often results in solving associated Euler-Lagrange equations characterizing an equilibrium state. While this is clearly different from our setting, it is also obvious that such discrete methods which are not modeled by a time-dependent flow can easily be modified such that a single deformation is broken into multiple steps defined, e.g., along a path (Section 6 shows an experiment). In fact, this may be regarded as a simple approach to emulating parametrization-independent, non-linear operators [6].

Isometry preserving deformations have been studied extensively in differential geometry (see, e.g., [8]) and mathematics in general: Efimow [12] theoretically investigated infinitesimal first order and higher order deformations. In this work we consider the first order case for piecewise linear discrete surfaces.

Finally, we remark that there are scenarios where physical objects including thin shells or cloth are modeled and shapes (or solids) obtain measured or synthesized material attributes. In a series of papers inspired by physically-based settings, Qin et al. [22, 10] consider time-dependent surfaces using a dynamic FEM formulation for shape modeling. Emphasis is on the design of smooth surfaces, the process is governed by a mass-spring system. For recent work on cloth simulation in the context of developable surfaces we refer to [13]. Generally such deformations must be physically correct or at least plausible. While this is far from our goals we remark that such simulations may be seen as continuous processes, see, e.g., the survey of Nealen et al. [20].

3 Minimum distortion shape integration

Our method is based on penalizing metric distortion when integrating vertex positions along a piecewise linear vector field. In this section we describe our approach in detail. We start with the derivation of the error measure for a single triangle. Contributions of each triangle are accumulated on the entire shape. Finally, we show how the concept can be extended naturally to incorporate smoothness of the resulting vector fields.

3.1 Error measure on a single triangle

Reviewing the situation for a single triangle T is sufficient to explain the error measure. We consider a triangle with vertex positions $\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^3$ and associated vectors $\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2 \in \mathbb{R}^3$. The triangle surface \mathbf{x} and the vector field \mathbf{v} are obtained by linear interpolation as

$$\begin{aligned}\mathbf{x}_T(u, v) &= u \mathbf{x}_0 + v \mathbf{x}_1 + (1 - u - v) \mathbf{x}_2 \\ \mathbf{v}_T(u, v) &= u \mathbf{v}_0 + v \mathbf{v}_1 + (1 - u - v) \mathbf{v}_2.\end{aligned}$$

The subsequent derivation is motivated by the fact that integrating perfectly rigid vector fields yields zero distortion. Obviously they won't yield a reasonable deformation either. However, local rigid fields can be easily constructed and serve as a reference: the closer \mathbf{v}_T is to a rigid field, the more isometric the deformation.

We consider a 3D vector field $\mathbf{r}(\mathbf{x})$ describing a *rigid* vector field, i.e., it can be written as $\mathbf{r}(\mathbf{x}) = \mathbf{r}_t + (\mathbf{r}_r \times \mathbf{x})$. Here, \mathbf{r}_t and \mathbf{r}_r describe the translational part and the rotation axis, respectively. Note that even though \mathbf{r} is defined everywhere in \mathbb{R}^3 , we evaluate it only on the triangle. We define the error e_T as squared difference of \mathbf{r} and \mathbf{v} integrated over the triangle T :

$$e_T(\mathbf{x}, \mathbf{v}, \mathbf{r}) = \int_0^1 \int_0^{1-v} \|\mathbf{v}(u, v) - \mathbf{r}(\mathbf{x}(u, v))\|^2 du dv. \quad (1)$$

This can be expressed in closed form as

$$e_T(\mathbf{x}, \mathbf{v}, \mathbf{r}) = \frac{1}{6} \sum_{(i,j) \in \{(0,1), (1,2), (2,0)\}} \|\mathbf{v}_{ij} - \mathbf{r}(\mathbf{x}_{ij})\|^2$$

with $\mathbf{x}_{ij} = \frac{1}{2}(\mathbf{x}_i + \mathbf{x}_j)$ and $\mathbf{v}_{ij} = \frac{1}{2}(\mathbf{v}_i + \mathbf{v}_j)$.

Given \mathbf{x} , our goal is to compute the best fitting rigid vector field $\hat{\mathbf{r}}(\mathbf{x}, \mathbf{v})$ as a function of \mathbf{v} by minimizing $e_T(\mathbf{x}, \mathbf{v}, \mathbf{r})$ for all rigid fields \mathbf{r} :

$$\hat{\mathbf{r}}(\mathbf{x}, \mathbf{v}) = \underset{\mathbf{r}}{\operatorname{argmin}} e_T(\mathbf{x}, \mathbf{v}, \mathbf{r}). \quad (2)$$

This is a linear least-squares problem in the six coefficients of \mathbf{r} (see also Section 4.2). Its solution depends on both, the positions \mathbf{x}_i and vectors \mathbf{v}_i , $i = 0, 1, 2$.

Finally, we express metric distortion of \mathbf{x} under instantaneous motion along \mathbf{v} as

$$d_T(\mathbf{x}, \mathbf{v}) = e_T(\mathbf{x}, \mathbf{v}, \hat{\mathbf{r}}(\mathbf{x}, \mathbf{v})). \quad (3)$$

3.2 Properties of d

The measure d_T is invariant under adding a rigid field to \mathbf{v} : let $\hat{\mathbf{r}}$ be the best fitting rigid field for \mathbf{x} and \mathbf{v} , and let $\mathbf{p}(\mathbf{x})$ be another arbitrary rigid vector field. Then the best fitting rigid field for \mathbf{x} and the modified vectors $\mathbf{v}'_i = (\mathbf{v}_i + \mathbf{p}(\mathbf{x}_i))$, $i = 0, 1, 2$, is $\hat{\mathbf{r}} + \mathbf{p}$. Furthermore, $d_T(\mathbf{x}, \mathbf{v}) = d_T(\mathbf{x}, \mathbf{v}')$.

The computation of $\hat{\mathbf{r}}$ is robust as long as the triangle is not degenerated, i.e., as long as the triangle area and the edge length ratios are bounded from below.

We emphasize that by construction $d_T(\mathbf{x}, \mathbf{v})$ measures metric distortion: isometric distortions of developable surfaces (see Figure 2a) indeed yield zero distortion although the deformation is not rigid. In particular this differs from as-rigid-as possible formulations (see [18] for a comparative discussion).

In the literature [18, 11], (discrete) metric distortion of a triangle under integration of its vertices is usually described as

$$\bar{d}_T(\mathbf{x}, \mathbf{v}) = h_0^2 + h_1^2 + h_2^2 \quad \text{with} \quad h_k = \mathbf{r}_k^\top (\mathbf{v}_{\pi_{k+1}} - \mathbf{v}_{\pi_k}), \quad \mathbf{r}_k = (\mathbf{x}_{\pi_{k+1}} - \mathbf{x}_{\pi_k}), \quad (4)$$

and $\pi_k = (k + 1) \bmod 3$. Then the summation of \bar{d}_T over all triangles is the global measure to be minimized. Our measure is related and compatible to this in the sense that $d_T(\mathbf{x}, \mathbf{v}) = (h_0, h_1, h_2) \mathbf{S} (h_0, h_1, h_2)^\top$. Let $r_{ij} = \mathbf{r}_i^\top \mathbf{r}_j$ and $\alpha = 4 \text{ area}(T)^2$. Then \mathbf{S} is a symmetric 3×3 matrix which depends only on \mathbf{x} and not on \mathbf{v} :

$$\mathbf{S} = \frac{-1}{144\alpha(r_{12} + r_{20} + r_{01})} \begin{bmatrix} 3r_{12}^2 + 4\alpha & 6r_{12}r_{20} - 4\alpha & 6r_{20}r_{01} - 4\alpha \\ 6r_{20}r_{12} - 4\alpha & 3r_{20}^2 + 4\alpha & 6r_{01}r_{12} - 4\alpha \\ 6r_{01}r_{20} - 4\alpha & 6r_{12}r_{01} - 4\alpha & 3r_{01}^2 + 4\alpha \end{bmatrix}.$$

Hence, generally no edge or area weighting-scheme exists which turns $\bar{d}_T(\mathbf{x}, \mathbf{v})$ into $d_T(\mathbf{x}, \mathbf{v})$, since \mathbf{S} is generally not diagonal. In contrast to $\bar{d}_T(\mathbf{x}, \mathbf{v})$ and roughly speaking, our measure $d_T(\mathbf{x}, \mathbf{v})$ also incorporate all mixed products $h_i h_j$, $i \neq j$.

The integration of quantities over triangles is essential to the design of our measure: this way, $d_T(\mathbf{x}, \mathbf{v})$ will be *invariant* to subdivision of triangles or generally independent of parametrization/tessellation. Figure 3a illustrates this by a simple example and compares $d_T(\mathbf{x}, \mathbf{v})$ to $\bar{d}_T(\mathbf{x}, \mathbf{v})$: we prescribe a vector field and apply the measure for different tessellations of the same shape, a unit sphere. Then tessellation-independence requires low *variance* of measured values. Geometrically the absolute values are meaningless for this experiment. However, they can physically be interpreted to be the applied membrane strain since isometric deformations are a geometric approximation of real-life, thin surfaces deforming with a very high Young's modulus [20]. Further experiments and a comparison are shown in Figures 6 and 3b (see Section 6). Tessellation-independence is generally an important requirement for many algorithms. It is essential for meaningful deformations also because coherence for time-dependent deformations is improved.

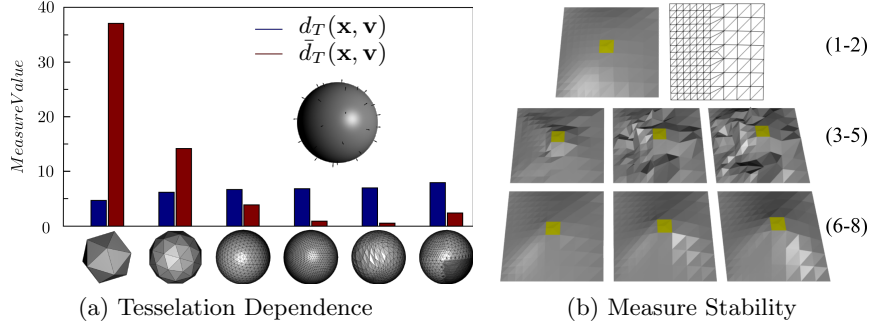


Fig. 3: Measure Comparison. (a) Different unit sphere tessellations and values of measures $d_T(\mathbf{x}, \mathbf{v})$ and $\bar{d}_T(\mathbf{x}, \mathbf{v})$ for the normal field. Important is the *variance* of the measure for different tessellations of the same shape which should be low. (b) The irregularly tessellated test surface (1-2) is Euler-integrated three steps by minimizing $\bar{d}_T(\mathbf{x}, \mathbf{v})$ (3-5), and by minimizing $d_T(\mathbf{x}, \mathbf{v})$ (6-8).

3.3 Error measure on the entire surface

The shape \mathcal{S} is given as a triangle mesh $\mathcal{M} = (\mathcal{V}, \mathcal{E}, \mathcal{T})$ with vertices \mathcal{V} , directed edges \mathcal{E} , and triangles \mathcal{T} . The embedding of the surface \mathbf{x} in 3-space is defined by vertex positions $\mathbf{x}_i \in \mathbb{R}^3$, furthermore the piecewise linear vector field \mathbf{v} is defined by vectors $\mathbf{v}_i, i = 1, \dots, |\mathcal{V}|$.

We define the global metric distortion as

$$d_1(\mathbf{x}, \mathbf{v}) = \sum_{(i,j,k) \in \mathcal{T}} d_T([\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k], [\mathbf{v}_i, \mathbf{v}_j, \mathbf{v}_k]) \quad (5)$$

With (3) it is evident that \mathbf{v} is a Killing field [8, 12, 18] of \mathbf{x} iff $d_1(\mathbf{x}, \mathbf{v}) = 0$. This also corresponds to intuition: if \mathbf{v} is a Killing field, the best fitting rigid field for every triangle will be identical to \mathbf{v} at each triangle, yielding $d_1(\mathbf{x}, \mathbf{v}) = 0$.

An error measure which is based solely on the preservation of isometry is obviously not sufficient to determine meaningful surface deformation: for instance, foldings yield perfect isometry whereas for shape deformation they are considered unwanted artifacts. The measure d_1 does not exclude, e.g., foldings of a developable surface (see, e.g., [17]). Consequently, we require an additional measure which penalizes discontinuous deformation and enforces smoothness of the vector field \mathbf{v} .

A suitable measure that fits our setting should be derived from existing quantities. We take advantage of the fact that the best fitting rigid vector fields $\hat{\mathbf{r}}$ are defined not only on respective triangles but everywhere in \mathbb{R}^3 . In particular, we can evaluate $\hat{\mathbf{r}}$ for a certain triangle T on an adjacent triangle T' .

Let triangles $T = (r, s, t)$ and $T' = (s, r, t')$ be adjacent with vertex positions \mathbf{x}_ℓ , and associated vectors $\mathbf{v}_\ell, \ell \in \{r, s, t, t'\}$. Furthermore, let $\hat{\mathbf{r}}$ and $\hat{\mathbf{r}}'$ be the best fitting rigid fields on T and T' . Then, loosely spoken, $\hat{\mathbf{r}}$ and $\mathbf{v}_{T'}$ should not

differ too much for a meaningful deformation. We formalize this by applying $\hat{\mathbf{r}}$ to T' (and $\hat{\mathbf{r}}'$ to T respectively). We obtain

$$f_{T,T'}(\mathbf{x}, \mathbf{v}) = \int_0^1 \int_0^{1-v} \|\mathbf{v}_{T'}(u, v) - \hat{\mathbf{r}}(\mathbf{x}_{T'}(u, v))\|^2 du dv$$

where $\mathbf{x}_{T'}(u, v)$ and $\mathbf{v}_{T'}(u, v)$ denote the linear interpolation as before but on triangle T' whereas $\hat{\mathbf{r}}$ is the best fitting rigid field on T . This can be expressed in closed form

$$f_{T,T'}(\mathbf{x}, \mathbf{v}) = \frac{1}{6} \sum_{(m,n) \in \{(r,s), (s,t'), (t',r)\}} \|\mathbf{v}_{mn} - \hat{\mathbf{r}}(\mathbf{x}_{mn})\|^2$$

with $\mathbf{x}_{mn} = \frac{1}{2}(\mathbf{x}_m + \mathbf{x}_n)$ and $\mathbf{v}_{mn} = \frac{1}{2}(\mathbf{v}_m + \mathbf{v}_n)$. With the measure $f_{T,T'}$ on single triangles we define the global measure d_2 on the entire shape as

$$d_2(\mathbf{x}, \mathbf{v}) = \sum_{T, T' \in \mathcal{T} \text{ adjacent}} f_{T,T'}(\mathbf{x}, \mathbf{v}).$$

In general $f_{T,T'}(\mathbf{x}, \mathbf{v}) \neq f_{T',T}(\mathbf{x}, \mathbf{v})$ which both have to be added in d_2 . Botsch et al. [5] use a similar principle to match transformations of incident prisms.

With the derivation of d_1 and d_2 we finally define the error measure which will be minimized in our approach as

$$d(\mathbf{x}, \mathbf{v}) = (1 - \omega) d_1(\mathbf{x}, \mathbf{v}) + \omega d_2(\mathbf{x}, \mathbf{v}) \quad (6)$$

for a small weight $\omega > 0$ (see below).

We motivated the second measure d_2 by the fact that isometry does not always convey enough information for meaningful deformations. We remark that d_2 is required for another reason: in special cases, minimizing d_1 yields a singular or ill-conditioned operator, and d_2 acts as a regularization term. For instance, a planar surface constitutes this special case. Of course, then it applies only to the first integration step – after that the surface is probably no longer planar. However, to ensure robustness of our approach in any possible situation we require $\omega > 0$.

Our experiments show that a rather high value in $(0, 1]$ can be chosen for ω without spoiling minimization of distortion, i.e., the effect of d_1 . The reason for this is that the definition of d_2 contains essentials of d_1 just with the difference of extrapolating the rigid fields. We close this section with two final remarks: First, the errors d_1 and d_2 are compatible in a sense that comparable quantities are measured. Second, there is a bias in the weighting as summation is over $|\mathcal{T}|$ triangles for d_1 and over $|\mathcal{E}|$ adjacent triangles for d_2 . From the latter relation Euler's formula yields $\omega = \frac{1}{3}$ for an even weighting.

3.4 Shape integration

Given the shape with vertex positions \mathbf{x} , our goal is to find a piecewise linear vector field $\hat{\mathbf{v}}$ that minimizes the error functional $d(\mathbf{x}, \mathbf{v})$. This vector field

minimizes metric distortion under integration of \mathbf{x} due to the definition of d_1 . Additionally, the contribution of d_2 to d accounts for smoothness of $\hat{\mathbf{v}}$.

Our approach to isometry preserving shape deformation assumes time-dependent shape $\mathbf{x}(t)$ and vector field $\hat{\mathbf{v}}(\mathbf{x}, t)$ such that

$$\frac{\partial}{\partial t} \mathbf{x}(t) = \hat{\mathbf{v}}(\mathbf{x}, t) .$$

In every time step t (or generally every point t where the vector field is evaluated) we determine $\hat{\mathbf{v}}(t)$ for $\mathbf{x}(t)$. The associated optimization problem is linear in the unknowns $\mathbf{v}_i, i = 1 \dots, |\mathcal{V}|$.

4 Implementation

In this section we discuss the implementation of each stage of our approach. These are firstly, the specification of deformations, secondly, the GPU based linear framework for finding vector fields that minimize our error measures in every time step, and finally, the numerical integration of the shape over time.

4.1 Defining deformations

Deformations are defined by certain constraints. In our case these are constraints on the vector field $\mathbf{v}(\mathbf{x}, t)$, i.e., for a subset of vertices, we prescribe the associated time-dependent vectors. This way we implement the standard handle metaphor for shape modeling: the user selects surface regions which are either fixed, deformable, or displaced by a handle. In fixed regions, the vector field is constant zero at each time step. The deformable regions define the free parameters, there the vector field is determined by minimization. In handle regions, vertices are displaced over time along smooth curves. The tangents of these curves define the vector field at each handle and hence determine the movement of handle vertices.

From the user's point of view, fixed and handle regions are selected. There is no restriction on the number of such regions or their connectivity. Then she prescribes arbitrary parametric guidance curves for moving the handles. Note that a single curve prescribes the rigid motion of a full handle region: every handle vertex is associated with the curves' tangent.

Such guidance curves can be designed easily and intuitively [15]. Indeed, our approach is even more general in a sense that we are not restricted to parametric curves for specifying constraints: any piecewise linear time-dependent vector field can be applied to prescribe motion of handles. In particular, twisting and bending of the shape can be modeled easily.

4.2 Minimizing error measures

Error measures are quadratic forms in the unknown vector field $\mathbf{v} \in \mathbb{R}^{3|\mathcal{V}|}$. Our goal is to minimize an expression

$$d(\mathbf{x}, \mathbf{v}) = \mathbf{v}^\top [(1 - \omega) \mathbf{D}_1 + \omega \mathbf{D}_2] \mathbf{v} .$$

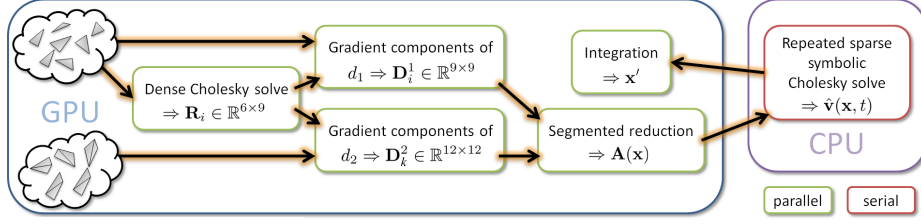


Fig. 4: Computational Pipeline. For each triangle on the GPU we compute the parameter mapping matrices \mathbf{R}_i which are used to determine the gradient components of the distortion and smoothness measures d_1 and d_2 . These coefficients are combined in the final sparse system matrix which is solved by the CPU using a precomputed symbolic factorization yielding the optimal vector field.

where \mathbf{D}_1 and \mathbf{D}_2 implement error measures d_1 and d_2 , respectively. The setup of the corresponding linear system $\mathbf{A}(\mathbf{x})$ is attended by considerable computational costs but is inherent parallizable. To guarantee fast execution times we opt for the combined GPU and CPU approach shown in Figure 4.

Best rigid fields. Both measures d_1 and d_2 depend on the evaluation of the best fitting rigid fields $\hat{\mathbf{r}}_i$ (parameterized by $\mathbf{p}_i = (\mathbf{r}_{t,i}^\top, \mathbf{r}_{r,i}^\top)^\top \in \mathbb{R}^6$) induced by the linear vector fields $\mathbf{v}_i = (\mathbf{v}_r^\top, \mathbf{v}_s^\top, \mathbf{v}_t^\top)^\top \in \mathbb{R}^9$ applied to single triangles $T_i = (r, s, t) \in \mathcal{T}$. For each triangle i we therefore find the linear maps $\mathbf{R}_i \mathbf{v}_i = \mathbf{p}_i$, $\mathbf{R}_i \in \mathbb{R}^{6 \times 9}$ relating linear and best rigid fields in the following way:

Switching to matrix notation (1) is expressed as $e_{T_i} = \|\mathbf{L}_i^\top (\mathbf{v} - \mathbf{E}_i \mathbf{p}_i)\|^2$. Here $\mathbf{E}_i \in \mathbb{R}^{9 \times 6}$ evaluates $\hat{\mathbf{r}}$ at the triangle vertices and \mathbf{L}_i^\top is the Cholesky factor of the matrix $\mathbf{N}_i = \mathbf{L}_i \mathbf{L}_i^\top \in \mathbb{R}^{9 \times 9}$ performing the integration along the triangle and which is defined by $N_{i,lm} = \begin{cases} \frac{\text{area}(T_i)}{6} & l = m \\ \frac{\text{area}(T_i)}{12} & (l - m) \bmod 3 = 0. \\ 0 & \text{else} \end{cases}$. Then

(2) is solved for \mathbf{R}_i by solving the linear system corresponding to $\nabla_{\mathbf{p}_i} e_{T_i} = \mathbf{0}$ yielding

$$\mathbf{R}_i = (\mathbf{E}_i^\top \mathbf{N}_i \mathbf{E}_i)^{-1} \mathbf{E}_i^\top \mathbf{N}_i.$$

We perform these independent computations in a parallized and numerically stable way for each triangle on the GPU by computing one dense Cholesky factorizations of $\mathbf{E}_i^\top \mathbf{N}_i \mathbf{E}_i$ and nine corresponding back-substitutions.

Distortion and smoothness gradients. For each triangle i its gradient components $\mathbf{D}_1^i \in \mathbb{R}^{9 \times 9}$ contributing to \mathbf{D}_1 are found by the evaluation of the gradient of $d_{T_i} = \|\mathbf{L}_i^\top (\mathbf{I} - \mathbf{E}_i \mathbf{R}_i) \mathbf{v}_i\|^2$ giving

$$\mathbf{D}_1^i = 2(\mathbf{I} - \mathbf{R}_i^\top \mathbf{E}_i^\top) \mathbf{N}_i (\mathbf{I} - \mathbf{E}_i \mathbf{R}_i).$$

Similarly we find $\mathbf{D}_2^k \in \mathbb{R}^{12 \times 12}$ contributing to \mathbf{D}_2 for each neighboring pair k of triangles T_i and $T_j = (s, r, t')$ by the evaluation of the gradient of

$f_{T_i, T_j} = \|\mathbf{L}_j^\top (\mathbf{I}' - \mathbf{E}_j \mathbf{R}_i \mathbf{P}_k) \mathbf{v}_{ij}\|^2$, where $\mathbf{I}' = [\mathbf{I}, \mathbf{0}]$ and with a permutation matrix $\mathbf{P}_k \in \mathbb{R}^{9 \times 12}$ selecting the vector \mathbf{v}_i corresponding to T_i out of $\mathbf{v}_{ij} = (\mathbf{v}_j^\top \mathbf{v}_{i'}^\top)^\top \in \mathbb{R}^{12}$ in the correct order. Computed in parallel on the GPU this yields

$$\mathbf{D}_2^k = 2 (\mathbf{I}'^\top - \mathbf{P}_k^\top \mathbf{R}_i^\top \mathbf{E}_j^\top) \mathbf{N}_j (\mathbf{I}' - \mathbf{E}_j \mathbf{R}_i \mathbf{P}_k).$$

Linear systems. In the last step the final sparse symmetric linear system \mathbf{A} is constructed as half of the Hessian of d in parallel by a weighted segmented reduction operation [23]. The non-zero entries are computed of all \mathbf{D}_1^i and \mathbf{D}_2^k by weighted sums according to

$$A_{ef} = \frac{1}{2} \frac{\partial^2 d}{\partial v_e \partial v_f} = \frac{1}{2} ((1 - \omega) D_{1,ef} + \omega D_{2,ef}).$$

We minimize $d(\mathbf{x}, \mathbf{v}) = \mathbf{v}^\top \mathbf{A} \mathbf{v}$ on the CPU subject to boundary conditions (see Section 4.1). \mathbf{A} is symmetric positive definite and sparse with about 1.5% non-zero entries. The linear systems are solved by state-of-the-art direct solvers, namely a sparse Cholesky factorization in combination with an approximate minimum degree reordering to reduce fill-in [9]. We exploit the fact that the structure of the linear system stays fixed in consecutive minimization steps, which allows the precomputation of a symbolic factorization that strongly accelerates the optimization. Experiments reveal that the direct CPU solver is two orders of magnitude faster than a GPU based sparse preconditioned conjugate gradient solver.

4.3 Numerical integration

In order to compute the deformation we require integration of vertex positions, i.e., we have to find the solution to an initial valued ordinary differential equation. Straightforward Euler integration yields visually pleasing results even for moderate time-steps. However, the lack of accuracy of this scheme would spoil our overall approach and render this scheme unacceptable. Instead we rely on higher order schemes.

For numerical integration we prefer a third order multistep predictor-corrector method with adaptive step size control: in contrast to single-step methods such as Runge-Kutta, the Adams-Bashforth-Moulton scheme (see, e.g., [21]) takes advantage of results from previous integrations steps. The initialization is provided by few fourth order Runge-Kutta steps. The main motivation for choosing this method is the relatively fewer number of expensive evaluations of the vector field required compared to single-step methods when assuming similar approximation errors. This choice was justified by experiments.

5 Analysis and results

In order to analyze our approach, we apply it to the four standard deformation problems defined by Botsch and Sorkine [6]. In particular, we compare our defor-

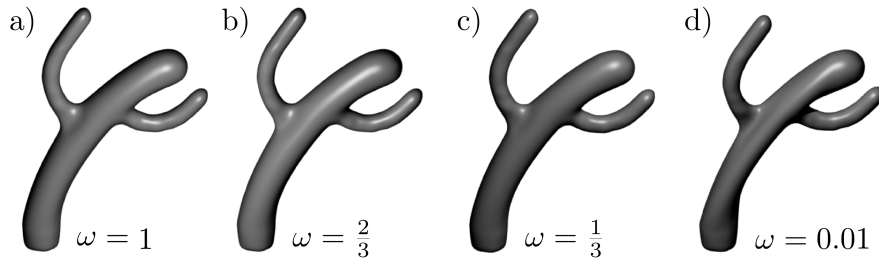


Fig. 5: Weighting of Error Measures. Smaller weights ω lead to small distortion (see table 1) but may produce deformation artifacts if the boundary conditions do not allow for an isometric deformation (d). Slightly higher ω values solve the problem (c).

mations with the (discrete) deformations described in [5] (PRIMO), [7] (THIN-SHELLS), [27] (GRADIENTED), [24] (LAPLACIANED), and [19] (ROTATIONINV). The results are summarized in Table 1. It first shows that our deformations look visually convincing (the images in Table 1 show the original and the deformed shape by using our method with $\omega = \frac{1}{3}$ for each of the four benchmark shapes).

In order to do an additional quantitative comparison, we compute an estimation of the final metric distortion by summing up the squared differences of the edge lengths in the original and deformed mesh (*metricErr*). In a similar way we compute the area distortion by considering the squared area difference over all triangles (*areaErr*), and the angular distortion by considering the squared angle differences over all triangles (*angleErr*). These values are measured between original and deformed shape for the reference deformation, and between original and final time step deformation for our method. For our method, we use four different values of ω . Since the absolute values of the distortion do not have a geometric meaning (because they depend on a particular triangulation), we normalized them by the distortion of our method with $\omega = \frac{1}{3}$. (A number above 100% in the table indicates a higher distortion than for our method with $\omega = \frac{1}{3}$.) For all examples, our technique shows significantly smaller metric distortion than any of the compared discrete deformation techniques. In fact, our continuous approach achieves even lower errors than the discrete, yet non-linear PRIMO approach. Moreover, the same statement holds for most of the techniques concerning area and angular distortion. Also note that our approach performs especially well on the bumpplane problem, since only at the small junctions of the bumps to the underlying plane minimal distortions are introduced and the remaining surface deforms isometrically with correct detail orientation.

Timings were measured on a 2.6GHz AMD Opteron system with 8GB RAM and a NVIDIA GTX 280 running CUDA. The number of required integration steps for each problem was 13 (cactus), 44 (bar), 46 (cylinder) and 76 (bump plane), and the number of vector-field evaluations was roughly twice as many. The first three models can be modified at interactive rates. However, performance is impaired for the very large bump plane model. Here the sparse solver becomes

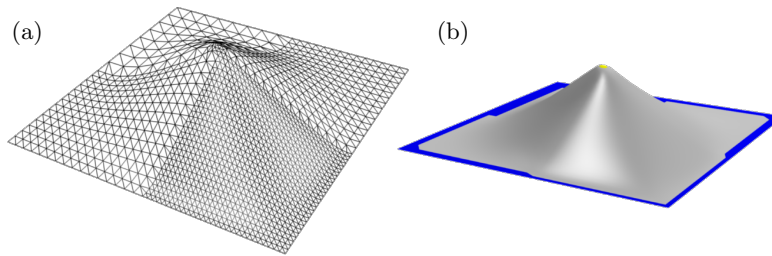


Fig. 6: Mesh Resolution Independence. Pulling the handle vertices up vertically produces a smooth deformation (b) independent of the inhomogeneous mesh resolution (a).

the bottleneck of the optimization due to the size of the arising linear system, which was not the case for all other examples in this paper.

Further examples of our approach are in Figures 2 and 8. Figure 2a shows a perfectly isometry-preserving deformation of a developable surface. Figure 2b (top) shows a twisting deformation of the head of a cow model. There, the body leans forward to compensate metric distortion. Figure 2c shows that the deformations can contain different handle paths: the front legs of the animals were moved in different directions, yielding realistic deformations. In Figure 8 a beetle car model is deformed in four antipodal directions giving four rather different deformation results.

The impact of the variation of the weight ω is illustrated in Figure 5 (error values are listed in Table 1). It shows that too small weights ω in (d) can lead to artifacts when deformations can not be perfectly isometric and metric distortion minimization is enforced at the expense of vector-field smoothness in (6).

6 Discussion

In this section we discuss several aspects of our approach.

Isometry Measure. For our approach, it was necessary to develop a new discrete measure of metric distortion of a vector field acting on the surface. The usual approach (4) as used in [18, 11] fails because of two reasons: first, it does not consider the shape and size of the triangles. Figure 3b gives an illustration of this: for this experiment, the surface $z(x, y) = \frac{1}{2}(1+x)(1-x)(1+y)(1-y)$ was sampled over the interval $[-1, 1] \times [-1, 1]$ as shown in Figure 3b (1-2). Note that an irregular tessellation was chosen. The deformation was defined by keeping the boundary constant and translating the (yellow) region in the direction of the z -axis. Figures 3b (3-5) show three steps of an Euler integration by minimizing $\bar{d}(\mathbf{x}, \mathbf{v})$, while Figure 3b (6-8) shows the same steps by minimizing $d(\mathbf{x}, \mathbf{v})$ with $\omega = 0$. Even this very small example clearly shows that the measure $\bar{d}(\mathbf{x}, \mathbf{v})$ does not yield the desired results. Note that this is not due to missing regulariza-

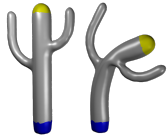
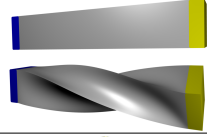

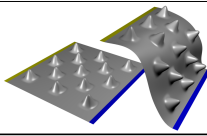
	PRiMo	THIN SHELLS	GRADIENTED	LAPLACIAN ED	ROTATION INV	New Method: ISOMETRY PRESERVING					
						$\omega = 1$	$\omega = \frac{2}{3}$	$\omega = \frac{1}{3}$	$\omega = 0.01$		
Cactus 70° Bend 5261V 10518T											
<i>metricErr</i>	(%)	128	215	199	831	242	101	101	100	90	
<i>areaErr</i>	(%)	125	196	194	739	96	95	100	81		
<i>angleErr</i>	(%)	167	272	275	1151	243	101	100	100	93	
Time	(s)						4	4	4	5	
Bar 135° Twist 6084V 12106T											
<i>metricErr</i>	(%)	411	2874	369	2586	333	109	107	100	32	
<i>areaErr</i>	(%)	130	10168	137	6951	878	107	105	100	55	
<i>angleErr</i>	(%)	331	96	298	1230	336	110	107	100	22	
Time	(s)						11	12	13	13	
Cylinder 120° Bend 4802V 9600T											
<i>metricErr</i>	(%)	184	1074	365	783	346	106	104	100	33	
<i>areaErr</i>	(%)	139	715	344	661	33	106	105	100	33	
<i>angleErr</i>	(%)	164	212	340	1016	242	105	104	100	31	
Time	(s)						8	8	9	9	
Bumpplane Translation 40401V 80000T											
<i>metricErr</i>	(%)	21478	24163	694283	366961	508844	118	112	100	10	
<i>areaErr</i>	(%)	18070	20611	513386	280151	458046	116	111	100	12	
<i>angleErr</i>	(%)	12911	11329	383755	654916	531962	116	111	100	13	
Time	(s)						212	241	275	287	

Table 1: Experimental Results. The tables summarizes the final errors of the benchmark problems. Measures are given relative to the absolute values of the shape integrated with parameter $\omega = \frac{1}{3}$. The images illustrate the deformation results of our approach using this parameter.

tion (the initial surface is not planar), the linear operators are sufficiently well conditioned.

The second reason for developing the new measure is that it offers a simple method to incorporate the smoothness of the surface, i.e., to prevent appearance or disappearance of sharp edges during the deformation. While we do not see a straightforward way to extend (4) in this direction, our measure can easily deal with it as shown in Section 3.3.

We conclude this aspect by visualization tessellation-independence of our deformation method for a simple example. Figure 6 shows a plane that is triangulated with different resolutions and then deformed trivially: the tessellation has no effect on the result due to the design of our measure (see also Figure 3a and Section 3.2).

Handle Path Dependency. The result of our deformation depends not only on the final position of the vertices building the boundary constraints but also on the paths on which they move from starting to final position. This is a significant difference to most existing deformation approaches. Figure 7 illustrates this. There, the shape (a) is deformed by moving the yellow boundary to the right

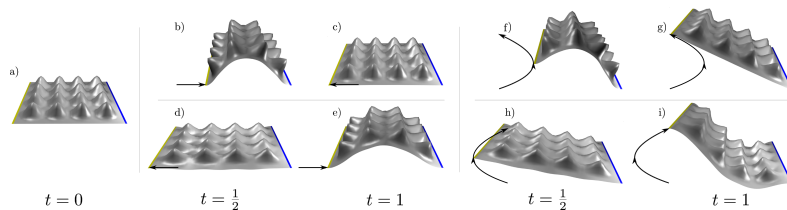


Fig. 7: Handle Path Dependency. Deforming the initial surface at $t = 0$ by different handle pathways from and to the same rest positions result in different deformation results at $t = 1$. The images (b–e) show two linear antipodal deformations, the images (f–i) two parabolic deformation curves.

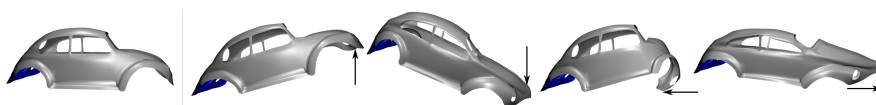


Fig. 8: Beetle Car Deformations. The original beetle model (left) is deformed by fixing the rear of the car and moving the handle at the engine hood into four different directions (right).

while keeping the blue boundary constant (b). The successive reverse deformation (c) gives almost the original shape. Contrary, moving the handle to the left first (Figure 7 (d)) ends up in a significantly different shape (e). Figures 7 (f–i) show the movement of the handle over two mirrored paths, yielding different final shapes.

While such a path dependence is not always the desired scenario, we believe that for a number of applications it opens a wider flexibility of the modeling process because it reflects the fact that real materials are never totally elastically deformed. The ”memory effect” of the deformation gives the look of a combined plastic and elastic deformation of a real material, even though only geometric measures of the surface are considered. Furthermore, it allows to obtain a strong twisting as shown in Figure 2b (bottom) which is impossible by path independent methods.

Comparing to time-dependent Laplacian. We point out that it is not sufficient to modify existent discrete and direct methods to operate in a continuous setting to minimize metric distortion. Consider Figure 9 as an example. For this deformation a bi-Laplacian operator (see [6]) was discretized for every time step, partial deformations were integrated within the same solver, boundary constraints are same as in Figure 6. Comparing results, metric distortion is still significantly higher – it didn’t improve much – than for our method. This is not surprising as different errors are minimized. We remark as bottom line that breaking a discrete bi-Laplacian deformation trivially into a ”continuous” deformation one cannot achieve the same effect as our continuous method and other discrete

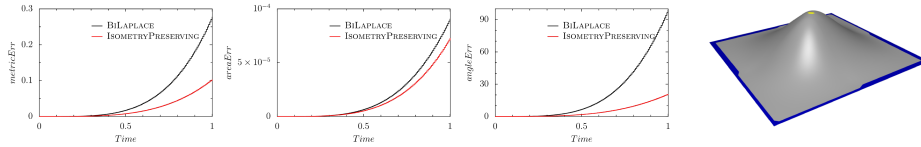


Fig. 9: Time-Dependent bi-Laplacian Deformation. Comparison of metric distortions over time for our method and time-dependent bi-Laplacian deformation.

state-of-the-art methods are likely to exhibit higher distortion by this strategy compared to our approach, too.

Limitations. We see the main limitation of the approach in the relatively high computation times: despite the fact that we accelerate our approach using the GPU, the technique is far less interactive than state-of-the-art linear frameworks when applied to large models and thus does not scale to very large meshes yet. We also mention that for the linear operators applied memory footprint is probably higher.

The path dependence of our method can as well be seen as a limitation. We claim it is a feature and an integral property of continuous deformations. However, we are aware that depending on the application path dependence may be also interpreted as an artifact.

7 Conclusions

In this paper, we made the following contributions: we defined the deformation by prescribing paths along which certain regions move over time. Then for every time step a piecewise linear vector field is constructed by applying a quadratic energy minimization approach. As a measure for metric distortion, we introduced a new approach which considers the size of the triangles and can be extended to incorporate the surface smoothness. The deformation tries to preserve isometry. It shows significantly lower distortion of length, angles, and area for a set of representative shapes compared to existing standard (linear and non-linear) deformations. Moreover, the results look visually pleasing. Our modeling metaphor defines handle paths. Both the final position of the handles and the path influence the deformation.

The most prominent issue in future research is to further improve the performance. In fact, we see reasonable chances to obtain higher frame rates using a multiresolution approach for the error measures and solvers. Another interesting challenge is the boundary value problem of path planning where the optimal path between two poses is determined.

References

1. Alexa, M., Cohen-Or, D., Levin, D.: As-rigid-as-possible shape interpolation. In: Proc. SIGGRAPH. pp. 157–164 (2000)

2. Angelidis, A., Cani, M.P., Wyvill, G., King, S.: Swirling-sweepers: Constant volume modeling. In: Pacific Graphics (2004)
3. Au, O.K.C., Tai, C.L., Liu, L., Fu, H.: Dual laplacian editing for meshes. *IEEE TVCG* 12(3), 386–395 (2006)
4. Botsch, M., Kobbelt, L.: An intuitive framework for real-time freeform modeling. *ACM Trans. Graphics* 23(3), 630–634 (2004)
5. Botsch, M., Pauly, M., Gross, M., Kobbelt, L.: Primo: coupled prisms for intuitive surface modeling. In: SGP. pp. 11–20 (2006)
6. Botsch, M., Sorkine, O.: On linear variational surface deformation methods. *IEEE Transactions on Visualization and Computer Graphics* 14(1), 213–230 (2008)
7. Botsch, M., Sumner, R., Pauly, M., Gross, M.: Deformation transfer for detail-preserving surface editing. In: Proc. of VMV. pp. 357–364 (2006)
8. do Carmo, M.P.: *Riemannian Geometry*. Birkhäuser Boston (1992)
9. Davis, T.A.: *Direct Methods for Sparse Linear Systems*. SIAM (2006)
10. Du, H., Qin, H.: Direct manipulation and interactive sculpting of PDE surfaces. In: CGF (Proc. Eurographics). vol. 19(3), pp. 261–270 (2000)
11. Eckstein, I., Pons, J.P., Tong, Y., Kuo, C.C.J., Desbrun, M.: Generalized surface flows for mesh processing. In: SGP. pp. 183–192 (2007)
12. Efimow, N.: *Flächenverbiegungen in Grossen*. Akadmie-Verlag (1957), (German)
13. English, E., Bridson, R.: Animating developable surfaces using nonconforming elements. *ACM Trans. Graphics* 27, 1–5 (2008)
14. von Funck, W., Theisel, H., Seidel, H.P.: Vector field based shape deformations. In: Proc. SIGGRAPH. pp. 1118–1125 (2006)
15. von Funck, W., Theisel, H., Seidel, H.P.: Explicit control of vector field based shape deformations. In: Proc. Pacific Graphics. pp. 291–300 (2007)
16. Huang, J., Shi, X., Liu, X., Zhou, K., Wei, L.Y., Teng, S.H., Bao, H., Guo, B., Shum, H.Y.: Subspace gradient domain mesh deformation. *ACM Trans. Graphics* 25(3), 1126–1134 (2006)
17. Kilian, M., Flöry, S., Chen, Z., Mitra, N.J., Sheffer, A., Pottmann, H.: Curved folding. *ACM Trans. Graphics* 27(3), 1–9 (2008)
18. Kilian, M., Mitra, N.J., Pottmann, H.: Geometric modeling in shape space. *ACM Trans. Graphics* 26(3) (2007), Proc. SIGGRAPH
19. Lipman, Y., Sorkine, O., Levin, D., Cohen-Or, D.: Linear rotation-invariant coordinates for meshes. In: Proc. SIGGRAPH. pp. 479–487. ACM Press (2005)
20. Nealen, A., Mueller, M., Keiser, R., Boxerman, E., Carlson, M.: Physically based deformable models in computer graphics. *CGF* 25, 809–836 (2006)
21. Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P.: *Numerical Recipes: The Art of Scientific Computing*. Cambridge University Press (2007)
22. Qin, H., Mandal, C., Vemuri, B.C.: Dynamic catmull-clark subdivision surfaces. *IEEE Transactions on Visualization and Computer Graphics* 4(3) (1998)
23. Sengupta, S., Harris, M., Zhang, Y., Owens, J.D.: Scan primitives for gpu computing. In: GH '07: Proc. of the 22nd ACM SIGGRAPH/EUROGRAPHICS symposium on Graphics hardware (2007)
24. Sorkine, O., Cohen-Or, D., Lipman, Y., Alexa, M., Rössl, C., Seidel, H.P.: Laplacian surface editing. In: SGP. pp. 175–184 (2004)
25. Sumner, R.W., Schmid, J., Pauly, M.: Embedded deformation for shape manipulation. *ACM Trans. Graphics* 26(3), 80:1–80:7 (Jul 2007)
26. Xu, W., Wang, J., Yin, K., Zhou, K., van de Panne, M., Chen, F., Guo, B.: Joint-aware manipulation of deformable models. In: Proc. SIGGRAPH (2009)
27. Zayer, R., Rössl, C., Karni, Z., Seidel, H.P.: Harmonic guidance for surface deformation. *CGF (Proc. EUROGRAPHICS)* 24(3), 601–609 (2005)