Generalized Metric Energies for Continuous Shape Deformation

Janick Martinez Esturo[1,2], Christian Rössl[1], and Holger Theisel[1]

[1] Visual Computing Group, University of Magdeburg, Germany
[2] Max Planck Institute for Computer Science, Saarland University, Germany

Abstract. High quality deformations of planar and volumetric domains are central to many computer graphics related problems like modeling, character animation, and non-rigid registration. Besides common "as-rigid-as-possible" approaches the class of nearly-isometric deformations is highly relevant to solve this kind of problems. Recent continuous deformation approaches try to find planar first order nearly-isometric deformations by integrating along approximate Killing vector fields (AKVFs). In this work we derive a generalized metric energy for deformation vector fields that has close-to-isometric AKVFs as a special case and additionally supports close-to-length-preserving, close-to-conformal as well as close-to-equiareal deformations. Like AKVF-based deformations we minimize nonlinear energies to first order using efficient linear optimizations. Our energy formulation supports nonhomogeneous as well as anisotropic behavior and we show that it is applicable to both planar and volumetric domains. We apply energy specific regularization to achieve smoothness and provide a GPU implementation for interactivity. We compare our approach to AKVF-based deformations for the planar case and demonstrate the effectiveness of our method for the $2d$ and $3d$ case.

Keywords: Shape Deformation, Isometry, Vector Field

1 Introduction

Persistent shape deformation is a classic problem in computer graphics and design. Even though numerous approaches haven been developed in the previous decades, is is still an important and active area of research. Applications for planar shape deformations include, e. g., image warping and cartoon animation. Deformation of $3d$ shapes is used in classic domains like in engineering for shape modeling or to create animations in the media industry, but also, e. g., for data registration in medical applications.

A recent trend is the development of continuous *nearly isometric* methods [21,29]. These deformations should preserve distances and, as a result, angles and area as much as possible. Intuitively, isometry is a good measure for the quality of a deformation: while the shape should accurately satisfy the constraints defining the deformation, it should not unnecessarily stretch or bend. Hence, near-isometric deformations yield intuitive and high quality results.

However, this does not come for free! Roughly speaking, high quality near-isometric deformations come for the price of solving *nonlinear* problems. This is a major issue especially for interactive applications, which are typical in computer graphics and are mandatory for interactive design. There is a competition with more efficient linear methods (see, e.g., [5]), which are based on simpler, often approximated differential quantities. It is well known that linear methods fail to handle isometry: most approaches either cope well with translations or with rotations – but not with both simultaneously. Also, there is no guarantee that the deformation does not induce local folds or self-intersections. We arrive at the conclusion that both, linear and nonlinear methods, have their own right to co-exist in shape deformation frameworks: the user has the choice between fast linear methods at the cost of sacrificing quality, and high quality nonlinear methods that are significantly more expensive to compute.

The user has to pay a certain price – higher computation times or smaller data sets – and therefore expects benefits from nonlinear methods. These include not only geometric properties of the deformation but also other important criteria related to usability. In summary, the computation of near-isometric shape deformations should fulfill a number of requirements, which make their computation a challenging problem:

- The isometric deformation problem is nonlinear. Nevertheless, computation must be *effective* and robust to guarantee a unique global optimum. In addition, computation must be *efficient* enough to enable real-time response to user input.
- Deformations must *interpolate* constraints, which can be defined for *any* point of the shape. Approximate satisfaction of "soft constraints" can be tolerated only if arbitrarily small tolerances are possible in principle.
- Ideally, the user can – globally and locally – attenuate isometry such that continuous blends from angle preservation to area preservation are possible. Anisotropic behavior is an additional design parameter for the user.
- Deformations must be *smooth* in a sense that the energy or metric error is distributed smoothly over the shape. In particular, the error must not concentrate near positional constraints.
- The discrete deformations must be independent of the particular partition of the shape or the domain. This implies *resolution/tessellation invariance.*
- Ideally, the formulation of the solution should be same for the $2d$ and the $3d$ case. This alleviates implementation.

So far, we are not aware of any isometry-preserving shape deformation method that meets all of the above design goals. In this paper, we present a new integral approach to continuous shape deformation that fulfills *all* requirements. Our approach is more general – but not more complicated – than previous methods. We define a generalized metric energy that has flows as minimizer that determine near-isometric, near-conformal, and to some extent near-equiareal deformations. In particular, we show that the recently proposed planar deformations based on as-Killing-as-possible vector fields (AKVF) [29] constitute a special case of our energy.

The derivation of error measures used in our method is neither based on the popular as-rigid-as-possible (ARAP) approaches nor on the recently used notion of discrete Killing fields. In contrast to iterative energy optimization required for ARAP, which converges to local optima only, our method is non-iterative. Instead, deformation is a time-dependent function, and we optimize for its derivative w.r.t. time and solve an initial value problem.

Our method can easily be integrated into existing tools. It is applicable to triangle meshes in $2d$ as well as to tetrahedral meshes in $3d$. It shares the common intuitive user interface where few points are fixed and few points act as handles, which can be dragged along paths in the domain by the user. In addition, the user can control metric properties of the deformation: we provide a single scalar parameter to obtain combinations of near-isometric and near-conformal deformations on a continuous scale. This parameter can be given globally as a single scalar or locally as a scalar field over the shape. Local anisotropic behavior is achieved by incorporating varying anisotropic energy norms.

2 Background and Related Work

The deformation of a shape consists of a map from the original shape to the deformed shape. Isometric maps preserve distances, which is equivalent to simultaneously preserving angles (conformal maps) and area (equiareal maps). For a rigorous introduction of the differential geometry of such maps we refer to [6]. Related to shape deformation is parametrization of surfaces, i.e., finding a map between a surface in $3d$ and a planar domain. Naturally, isometry is a desired property for such maps; a pioneering approach is the construction of most isometric parametrizations [14]. Liu et al. [24] present hybrid parametrizations that interpolate locally rigid or local similarity transformations, which is similar to our generalized framework for continuous deformations. In the following, we consider only shape deformation methods. For a discussion of parametrization methods we refer to the survey of Hormann et al. [15]. We restrict our review of related work to nonlinear deformation methods. For a review of linear methods and a discussion of differences to nonlinear methods we refer to the survey [5].

A popular approach to isometry preservation is to restrict deformations locally to rigid transformations, i.e., translation and rotation. (Reflection is undesired.) This leads to the notion of the well-established as-rigid-as-possible (ARAP) maps, which where initially introduced for shape interpolation [1] and later applied for shape deformation [17,31]. Until today, there have emerged numerous extensions like [3,32,20,8], to mention just a few. ARAP approaches minimize a nonlinear energy expressing rigidity subject to constraints like fixed and displaced points. The classic approach consists in an iterative algorithm, which repeatedly estimates local rotations to build the global deformation until convergence. There are also alternative nonlinear deformation energies that enforce rigidity in form of, e. g., the rest energy of coupled rigid prisms [4]. Independent of the energy and the particular numerical scheme, the deformation is obtained as the minimizer of a particular energy in the shape coordinates at a singular point in time: we refer to such methods as *single step* methods.

In contrast, *continuous* methods evolve the deformation over time: the energy minimizer at each infinitesimal time step determines the gradient of the deformation, and the final deformation is obtained as the solution of an ordinary differential equation. From a technical point of view, the iterative solvers for minimizing nonlinear energies are, roughly speaking, replaced by a numerical ODE integration method. The latter is a standard numerical problem that is well-understood and that can be solved efficiently and reliably. In addition, the mapping to the deformed shape is guaranteed to be locally bijective if the deformation gradient does not vanish, i.e., if the deformation flow does not contain critical points, and hence the deformations does not show local fold-overs. The main benefit of the continuous methods, however, consists in the fact that finding deformation gradients is a linear problem for near-isometric deformations.

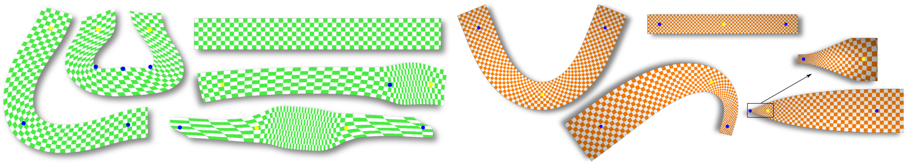
Isometry preservation is guaranteed for integration of exact Killing vector fields, see, e.g., [2]. Kilian et al. [21] approximate Killing vector fields for interpolation in a shape space, which yields deformations of a 2-manifold that is embedded in 3-space. Note that they compare this isometry preserving approach to continuous deformations based on the ARAP concept, which yields a related but different deformation class. Martinez et al. [25] extend their discretization towards tessellation independence and smoothness. Heeren et al. [13] use physical discrete shell energies to construct time-discrete geodesics in a different shape space. Solomon et al. [29] introduce the notion of as-Killing-as-possible (AKVF) deformations in planar domains. In contrast to the above approaches, they ensure smoothness by a post-process rather than by a regularization term, and instead of a standard ODE solver, they use planar holomorphic curves as a predictor to construct the trajectories. They obtain high quality deformations, which they compare to various other planar shape deformation methods. In summary, their results suggest that it is more than worthwhile and often preferable to consider near-isometric shape deformations.

Funck et al. [11] developed a remarkably different approach to continuous $3d$ shape deformation, which preserves volume by integration of divergence-free vector fields. Continuous deformation can also be obtained by fitting continuous shape manifolds to key frames [7].

There are various alternative methods for planar and volumetric shape deformation. One prominent class of methods is based on generalized barycentric coordinates, e.g., [16,19,23,18,33,34]. Besides isometry here is also a demand for conformal maps, which are produced by none of the above methods.

3 Continuous Metric Energies

In this section we introduce continuous deformations formally and derive energy terms that determine isometric and conformal deformations in $2d$ and $3d$. The section concludes with a generalized formulation of an integral energy that determines a one-parameter family of continuous deformations, which includes near-isometric, near-conformal, and close-to-equiareal deformations.



(a) Near-isometric.

(b) Near-conformal.

Fig. 1. $2D$ Deformation Examples. On the straight strip some vertices were fixed (●) while some vertices were moved (●) (all models have the same scale). The deformations are generated by AMAP (8) and ACAP (12) vector fields. Note the approximate length preservation in (a), and the preservation of angles and the area deviation in (b).

3.1 Continuous Deformations

A continuous deformation is a time-dependent map $\mathbf{f} : \Omega_d^0 \times \mathrm{I\!R} \rightarrow \mathrm{I\!R}^d$ with $\Omega_d^0 \subseteq \mathrm{I\!R}^d$, i.e., a time-dependent map from a domain Ω_d^0 to $\mathrm{I\!R}^d$. We primarily consider the important dimensions $d = 2$ and $d = 3$ in this work but also provide some generalizations for higher dimensions. Let $\mathcal{X}_0 \subseteq \Omega_d^0$ be a point set defining some initial shape. Then the deformed shape at time t is expressed as the image $\mathbf{f}(\mathcal{X}_0, t)$. We use the short notation $\Omega_d = \mathbf{f}(\Omega_d^0, t)$ for the deformed domain at the current time t, which is clear from the context.

We define the velocity of $\mathbf{f}$ as the vector field $\mathbf{v}(\mathbf{x}, t) = \frac{\mathrm{d}}{\mathrm{d}t}\mathbf{f}(\mathbf{x}, t)$. Then $\mathbf{f}(\mathbf{x}, t)$ can be reconstructed from $\mathbf{v}$ by solving the initial value problem

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{x}(t) \;=\; \mathbf{v}(\mathbf{x}, t) \quad \text{with } \mathbf{x}(0) = \mathcal{X}_0 \ .$$

In the following, we derive conditions on $\mathbf{v}$ that lead to near-isometric and near-conformal maps $\mathbf{f}$. The conditions are characterized as the minimizers of certain energy terms w.r.t. interpolation constraints on $\mathbf{v}$. Figure 1 shows examples for deformations that were determined by this kind of vector fields.

3.2 Characteristic Deformations

For a single step (i. e., not time-dependent) deformation $\mathbf{f} : \Omega_d^0 \rightarrow \mathrm{I\!R}^d$ with deformation gradient $\mathbf{D} = \nabla\mathbf{f}$ the first fundamental form $\mathcal{I}$ of $\mathbf{f}$ has the particularly simple form

$$\mathcal{I} = \mathbf{D}^{\mathrm{T}}\,\mathbf{D} \ .$$

Therefore, the *singular values* σ_i of $\mathbf{D}$ are square roots of the *eigenvalues* λ_i of $\mathcal{I}$. Then the following equivalent local properties of the deformation map can be shown (see, e. g., the work of Floater and Hormann in the context of parameterizations for the case of $d = 2$ [10]):

$$1.\ \mathbf{f}\text{ is isometric} \quad \Leftrightarrow \quad \mathcal{I} = \mathbf{I} \quad \Leftrightarrow \quad \lambda_i = 1 \quad \Leftrightarrow \quad \sigma_i = 1, \tag{1}$$

$$2.\ \mathbf{f}\text{ is conformal} \quad \Leftrightarrow \quad \mathcal{I} = \mu\,\mathbf{I} \quad \Leftrightarrow \quad \frac{\lambda_i}{\lambda_j} = 1 \quad \Leftrightarrow \quad \frac{\sigma_i}{\sigma_j} = 1, \tag{2}$$

$$3.\ \mathbf{f}\text{ is equiareal} \quad \Leftrightarrow \quad \det\mathcal{I} = 1 \quad \Leftrightarrow \quad \textstyle\prod_{i=1}^d \lambda_i = 1 \quad \Leftrightarrow \quad \prod_{i=1}^d \sigma_i = 1 \ . \tag{3}$$

Note that surface parameterizations can be regarded as deformations between $2d$ and $3d$, and in this work we consider the instantaneous deformation energy.

When deformations are parameterized by time t (i. e., we have continuous $\mathbf{f}(\mathbf{x}, t)$ and $\mathbf{D}(\mathbf{x}, t)$) these properties can be differentiated in order to obtain defining conditions on the vector field of the continuous deformation. Specifically, we apply the matrix algebra described by Minka [27] to obtain matrix derivatives w. r. t. t. They define the differential $\mathrm{d}\mathbf{y}(\mathbf{x})$ to be the part of $\mathbf{y}(\mathbf{x}+\mathrm{d}\mathbf{x})-\mathbf{y}(\mathbf{x})$ that is linear in $\mathrm{d}\mathbf{x}$. Differentials are obtained by iteratively applying a set of differentiation rules. After transformation into canonical form the matrix derivative can directly be read off.

3.3 Isometric Energies

Exact isometric deformations that fulfill all user constraints are not always possible. Therefore, measures for the *deviation* from isometry are required and we continue to present two possible models: Killing and metric energies.

Killing energy. The matrix derivative of the isometry property is obtained by deducing the differential of (1), which gives

$$\mathrm{d}\mathbf{D}^{\mathrm{T}}\,\mathbf{D} + \mathbf{D}^{\mathrm{T}}\,\mathrm{d}\mathbf{D} = \mathbf{0}$$

using the product rule $\mathrm{d}(\mathbf{A}\,\mathbf{B}) = \mathrm{d}\mathbf{A}\,\mathbf{B} + \mathbf{A}\,\mathrm{d}\mathbf{B}$ and $\mathrm{d}\mathbf{I} = \mathbf{0}$. This equality has to hold for every time t of the continuous deformation. Specifically, for $t = 0$ we have $\mathbf{D}(\mathbf{x}_0, 0) = \mathbf{I}$ and by using $\mathrm{d}\mathbf{D} = \mathbf{J}^{\mathrm{T}}$, where $\mathbf{J}$ is the Jacobian of the tangent vector field of $\mathbf{f}$, we obtain

$$\mathbf{J}^{\mathrm{T}} + \mathbf{J} = \mathbf{0} \tag{4}$$

as the condition for $\mathbf{f}$ to be isometric expressed in the vector field of the continuous deformation. Equation (4) corresponds to the constraint that exact isometric deformations are generated by infinitesimal rotations, since the symmetric part of their Jacobian, which is skew-symmetric then, vanishes.

The L^2 deviation of (4) over a domain Ω_d

$$E_{\mathrm{AKVF}}(\mathbf{v}) = \int_{\Omega_d} \left|\left|\mathbf{J}^{\mathrm{T}} + \mathbf{J}\right|\right|_F^2 \,\mathrm{d}\mathbf{x} \tag{5}$$

is called *Killing energy* with the Frobenius norm $||\cdot||_F$. It is used by Solomon et al. [29] for the case $d = 2$ to define *as-Killing-as-possible* vector fields $\mathbf{v}$ that minimize E_{AKVF} and which therefore generate near-isometric planar deformations. Higher dimensional cases ($d > 2$) are also well-defined. Note that the Jacobian is linear in the unknown vector fields $\mathbf{v}$ as differentiation is a linear operation, i. e., there exists a gradient operator $\mathbf{G}$ on Ω_d with $\mathbf{J} = \mathbf{G}\,\mathbf{v}$. Therefore, the energy (5) is quadratic in $\mathbf{v}$ and the corresponding variational optimization of (5) leads to a linear system that can efficiently be solved for the optimal vector field.

Metric energy. The classic Killing energy (5) uses the Frobenius norm of (4) to measure deviation from isometry. We propose a related energy that measures another form of deviation from isometry that is not based on a L^2 deviation of (4). Informally spoken, our energy directly observes an infinitesimally small line segment and measures change of length under an infinitesimal integration step in $\mathbf{v}$. This is done for all possible infinitesimal segments, i.e., we integrate the (squared) change of length over all possible directions. We call this energy *metric* as distance variations are measured explicitly. We start with the derivation of the $2d$ case followed by the $3d$ case.

In order to measure the variation of length under integration in $\mathbf{v}$ we consider a line segment $\mathcal{S}$ between points $\mathbf{x}_0$ and $\mathbf{x}_1 = \mathbf{x}_0 + r_1 \mathbf{r}_1$ for a unit direction $\mathbf{r}_1$ and segment length r_1. The flow of $\mathcal{S}$ in $\mathbf{v}$ is given as $\mathbf{x}'_0(h) = \mathbf{x}_0 + \int_0^h \mathbf{v}(\mathbf{x}'_0(s))ds$ and $\mathbf{x}'_1(h) = \mathbf{x}_1 + \int_0^h \mathbf{v}(\mathbf{x}'_1(s))ds$. This induces the quadratic length variation

$$d^l(h) = ||\mathbf{x}_1 - \mathbf{x}_0||^2 - ||\mathbf{x}'_1(h) - \mathbf{x}'_0(h)||^2 \ .$$

Since we are interested in instantaneous variations (i.e., the length variation of an infinitesimal small line segment during an infinitesimal small integration) only, we consider the limit

$$d^l_0(\mathbf{r}_1) = \lim_{h\to 0,\, r_1 \to 0} \frac{d^l(h)}{r_1^2\, h} \ = \ \frac{\partial^3 d^l(h)}{\partial r_1^2 \, \partial h} \ .$$

d^l_0 measures the instantaneous quadratic length variation for the direction $\mathbf{r}_1$. We obtain the pointwise quadratic isometric energy $e_{\text{METR}}(\mathbf{x}_0, \mathbf{v})$ at $\mathbf{x}_0$ by considering all possible line segment directions given by $\mathbf{r}_1(\alpha) = (\cos(\alpha)\, , \sin(\alpha)\,)^{\mathrm{T}}$:

$$e_{\text{METR}}(\mathbf{x}_0, \mathbf{v}) = \frac{1}{2\pi} \int_0^{2\pi} d^l_0(\mathbf{r}_1(\alpha))^2 \, \mathrm{d}\alpha \ . \tag{6}$$

It can be shown that (6) has the following closed form solution that depends only on the Jacobian of $\mathbf{v}$[3]:

$$\begin{aligned} e_{\text{METR}}(\mathbf{x}_0, \mathbf{v}) &= u_x^2 + v_y^2 + \frac{1}{2}\,(u_y + v_x)^2 + \frac{1}{2}\,(u_x + v_y)^2 \\ &= c\, \left(\left|\left|\mathbf{J} + \mathbf{J}^{\mathrm{T}}\right|\right|_F^2 + 2\,(\mathrm{Tr}\ \mathbf{J})^2 \right) \ . \end{aligned} \tag{7}$$

Here $\mathbf{J} = \left[\begin{smallmatrix} u_x & u_y \\ v_x & v_y \end{smallmatrix}\right]$ denotes the Jacobian of $\mathbf{v}$ at $\mathbf{x}_0$, $\mathrm{Tr}\,\cdot$ is the trace of a matrix, and c is a constant factor. The total *metric energy* of a vector field $\mathbf{v}$ on Ω_2 is now given by

$$E_{\text{METR}}(\mathbf{v}) = \int_{\Omega_2} e_{\text{METR}}(\mathbf{x}, \mathbf{v})\, \mathrm{d}\mathbf{x} \ . \tag{8}$$

We call vector fields that minimize this energy *as-metric-as-possible* (AMAP) vector fields. Figure 1(a) shows examples for deformations that were determined by this kind of vector fields.

[3] The derivation of equivalence is lengthy but consists only of basic algebraic transformations and therefore is omitted in the paper. — We provide derivations in form of Maple scripts for *all closed form solutions of integrals* in this submission with the additional material.

We derive a similar energy for $d = 3$ dimensions using the same ansatz as above for $d = 2$. Again we take the integral over all possible configurations of an infinitesimal integration step of an infinitesimally small line segment between two points $\mathbf{x}_0$ and $\mathbf{x}_1 = \mathbf{x}_0 + r_1\,\mathbf{r}_1$. The main difference to the $2d$ case is that angles in the plane now have to be replaced by solid angles. For the spherical parametrization of the unit direction $\mathbf{r}_1(\alpha,\beta) = (\cos(\alpha)\cos(\beta)\,, \sin(\alpha)\cos(\beta)\,, \sin(\beta)\,)^{\mathrm{T}} \in \mathrm{I\!R}^3$ we obtain the pointwise quadratic metric energy as the integral

$$e_{\text{METR3D}}(\mathbf{x}_0,\mathbf{v}) = \frac{1}{4\pi}\int_0^{2\pi}\int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} \cos(\beta)\; d_0^l(\mathbf{r}_1(\alpha,\beta))^2\,\mathrm{d}\beta\,\mathrm{d}\alpha \;, \tag{9}$$

which again has the closed form solution

$$e_{\text{METR3D}}(\mathbf{x}_0,\mathbf{v}) = c\left(\left|\left|\mathbf{J}+\mathbf{J}^{\mathrm{T}}\right|\right|_F^2 + 2\,(\mathrm{Tr}\;\mathbf{J})^2\right) \;. \tag{10}$$

Interestingly the factors of (7) and (10) only differ in the constant c, although their dimensions differ. The total $3d$ metric energy is then obtained as

$$E_{\text{METR3D}}(\mathbf{v}) = \int_{\Omega_3} e_{\text{METR3D}}(\mathbf{x},\mathbf{v})\,\mathrm{d}\mathbf{x} \;.$$

We again call the minimizers of this energy *as-metric-as-possible* vector fields. In the following we will use the terms METR and METR3D synonymously whenever the context is clear.

3.4 Conformal Energy

The differential of (2) is given by

$$\mathrm{d}\mathbf{D}^{\mathrm{T}}\mathbf{D} + \mathbf{D}^{\mathrm{T}}\,\mathrm{d}\mathbf{D} = \mathrm{d}\mu\,\mathbf{I} \;.$$

We again evaluate it at $t = 0$, and by setting $\mathrm{d}\mu = \alpha$ we obtain

$$\mathbf{J}^{\mathrm{T}} + \mathbf{J} = \alpha\,\mathbf{I}$$

as the condition for the continuous deformation $\mathbf{f}$ to be conformal. Note that here α is an additional degree of freedom stating the fact that instantaneous uniform scaling is conformal for every scaling factor.

We derive a pointwise energy / energy density e_{CONF} that measures the L^2 deviation of this conformality condition. The construction of the energy holds for any dimension d from which important two and three-dimensional special cases can be obtained:

$$\begin{aligned}
e_{\text{CONF}} &= \left|\left|\mathbf{J}^{\mathrm{T}} + \mathbf{J} - \alpha\,\mathbf{I}\right|\right|_F^2 \\
&= \mathrm{Tr}\Big(\left(\mathbf{J}^{\mathrm{T}} + \mathbf{J} - \alpha\,\mathbf{I}\right)^{\mathrm{T}}\left(\mathbf{J}^{\mathrm{T}} + \mathbf{J} - \alpha\,\mathbf{I}\right)\Big) \\
&= \mathrm{Tr}\Big(\left(\mathbf{J}^{\mathrm{T}} + \mathbf{J}\right)^{\mathrm{T}}\left(\mathbf{J}^{\mathrm{T}} + \mathbf{J}\right)\Big) + \mathrm{Tr}\big(-2\alpha\left(\mathbf{J}^{\mathrm{T}} + \mathbf{J}\right) + \alpha^2\mathbf{I}\big) \\
&= \left|\left|\mathbf{J}^{\mathrm{T}} + \mathbf{J}\right|\right|_F^2 - 2\alpha\,\mathrm{Tr}\big(\mathbf{J}^{\mathrm{T}} + \mathbf{J}\big) + d\,\alpha^2 \\
&= \left|\left|\mathbf{J}^{\mathrm{T}} + \mathbf{J}\right|\right|_F^2 - 4\alpha\,\mathrm{Tr}\,\mathbf{J} + d\,\alpha^2
\end{aligned} \tag{11}$$

This energy formulation still depends on the scaling factor α. To obtain an expression that is independent of this parameter we consistently set it to the value that minimizes the value of the energy. That is, we solve $\nabla_\alpha\, e_{\text{CONF}} = 0$ for α, which gives $\alpha = \frac{2}{d}\operatorname{Tr}\mathbf{J}$. Inserting this result into (11) we obtain

$$e_{\text{CONF}} = \left|\left|\mathbf{J}^{\mathrm{T}} + \mathbf{J}\right|\right|_F^2 - \frac{4}{d}(\operatorname{Tr}\mathbf{J})^2$$

for the general d-dimensional pointwise conformal energy in the vector field of the continuous deformation. The total *conformal energy* of the vector field is then given by

$$E_{\text{CONF}}(\mathbf{v}) = \int_{\Omega_d} \left|\left|\mathbf{J}^{\mathrm{T}} + \mathbf{J}\right|\right|_F^2 - \frac{4}{d}(\operatorname{Tr}\mathbf{J})^2 \,\mathrm{d}\mathbf{x}\ .$$

Again, this energy is quadratic in the vector field. We call vector fields minimizing this energy *as-conformal-as-possible* (ACAP). The important low-dimensional special cases are

$$E_{\text{CONF2D}}(\mathbf{v}) = \int_{\Omega_2} \left|\left|\mathbf{J}^{\mathrm{T}} + \mathbf{J}\right|\right|_F^2 - 2\,(\operatorname{Tr}\mathbf{J})^2 \,\mathrm{d}\mathbf{x} \quad \text{and} \tag{12}$$

$$E_{\text{CONF3D}}(\mathbf{v}) = \int_{\Omega_3} \left|\left|\mathbf{J}^{\mathrm{T}} + \mathbf{J}\right|\right|_F^2 - \frac{4}{3}\,(\operatorname{Tr}\mathbf{J})^2 \,\mathrm{d}\mathbf{x}\ .$$

See Figure 1(b) for example deformations that were determined by these vector fields.

3.5 Equiareal Energy

In order to obtain the condition on the vector field for the continuous deformation to be equiareal, we differentiate (3) using the differentiation rule $\mathrm{d}\det\mathbf{A} = \det\mathbf{A}\ \operatorname{Tr}\big(\mathbf{A}^{-1}\,\mathrm{d}\mathbf{A}\big)$:

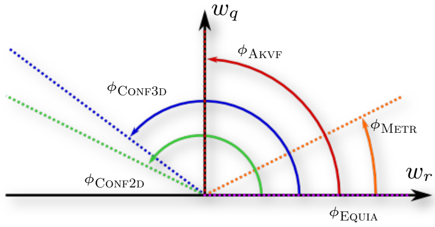
$$\begin{aligned}\mathrm{d}\det\big(\mathbf{D}^{\mathrm{T}}\mathbf{D}\big) &= 2\,(\mathrm{d}\det\mathbf{D})\ \det\mathbf{D}\\ &= 2\,\big(\det\mathbf{D}\ \operatorname{Tr}\big(\mathbf{D}^{-1}\,\mathrm{d}\mathbf{D}\big)\,\big)\ \det\mathbf{D}\\ &= \operatorname{Tr}\Big(2\,(\det\mathbf{D})^2\ \mathbf{D}^{-1}\,\mathrm{d}\mathbf{D}\Big)\end{aligned} \tag{13}$$

Evaluating (13) at $t = 0$ and using $\mathrm{d}\mathbf{D} = \mathbf{J}$ the equiareal condition on the vector field simplifies to

$$\operatorname{Tr}\mathbf{J} = 0,$$

which states that the vector field has to be divergence free as $\operatorname{Tr}\mathbf{J} = \nabla\cdot\mathbf{v}$. The corresponding L^2 pointwise equiareal energy $e_{\text{EQUIA}} = (\nabla\cdot\mathbf{v})^2$ yields the total equiareal energy

$$E_{\text{EQUIA}}(\mathbf{v}) = \int_{\Omega_d} (\nabla\cdot\mathbf{v})^2 \,\mathrm{d}\mathbf{x}\ .$$



Energy	w_q	:	w_r	ϕ
Metr $2d$ & $3d$	1	:	2	$\arctan \frac{1}{2}$
Conf $2d$	1	:	-2	$\pi - \arctan \frac{1}{2}$
Conf $3d$	1	:	$-\frac{4}{3}$	$\pi - \arctan \frac{4}{3}$
Akvf $2d$ & $3d$	1	:	0	$\frac{\pi}{2}$
Equia $2d$ & $3d$	0	:	1	0

Fig. 2. Energy Parameter Domain. The different energies obtained from the general metric energy $E_\phi(\mathbf{v})$ are linear subspaces in the visualized domain of weights w_q and w_r, i.e., every pair of weights in a subspace yields the same energy minimizer. However, energies may not have unique minimizers, like the Equia energy in the limit.

3.6 A Generalized Family of Energies

In the following we relate the near-isometric, near-conformal, and near-equiareal energies to derive a generalized energy. This is a one-parameter family of energies that determine smooth blends between the different types of deformation.

We define

$$q(\mathbf{x}) := \left\| \mathbf{J}(\mathbf{x}) + \mathbf{J}(\mathbf{x})^{\mathrm{T}} \right\|_F^2 \quad \text{and} \quad r(\mathbf{x}) := (\operatorname{Tr} \mathbf{J}(\mathbf{x}))^2 \ .$$

Then all energy densities introduced so far can be expressed as linear combinations of $q(\mathbf{x})$ and $r(\mathbf{x})$. Uniform scaling of such an energy does not change the minimizing vector field. Therefore, we can describe all energies as a one-parameter family of *generalized metric energies* depending on ϕ:

$$E_\phi(\mathbf{v}) = \int_{\Omega_{2/3}} w_q(\phi)\, q(\mathbf{x}) \ + \ w_r(\phi)\, r(\mathbf{x})\, \mathrm{d}\mathbf{x}$$

with the weights $w_q(\phi) := \sin(\phi)$ and $w_r(\phi) := \cos(\phi)$ having specific ratios. In $2d$, ϕ can vary in the interval $]0, \pi - \arctan \frac{1}{2}]$, while in $3d$, ϕ varies in $]0, \pi - \arctan \frac{4}{3}]$. Then the parameter of the isometric energies is given by $\phi = \phi_{\text{AKVF}} = \frac{\pi}{2}$, resp. $\phi = \phi_{\text{METR}} = \arctan \frac{1}{2}$, and the minimizers of $E_{\phi_{\text{AKVF}}}(\mathbf{v})$ and $E_{\text{AKVF}}(\mathbf{v})$, resp. $E_{\phi_{\text{METR}}}(\mathbf{v})$ and $E_{\text{METR}}(\mathbf{v})$ are equal. Furthermore, the conformal energy is given by $\phi = \phi_{\text{CONF2D}} = \pi - \arctan \frac{1}{2}$ in $2d$ and by $\phi = \phi_{\text{CONF3D}} = \pi - \arctan \frac{4}{3}$ in $3d$, respectively. The equiareal energy is recovered for $\phi = \phi_{\text{EQUIA}} = 0$. We note that volume preservation is not a sufficient condition for uniquely defining $\mathbf{v}$. However, adding a small amount of q to E_ϕ (i. e., choosing ϕ slightly above zero) gives unique solutions corresponding to near-equiareal deformations.

Figure 2 illustrates different choices of ϕ. Note that for $\phi > \frac{\pi}{2}$, E_ϕ contains negative quadratic terms. However, due to the definition of conformal energy density (11) as a squared matrix norm, it is guaranteed that E_ϕ is non-negative as long as $\phi \leq \pi - \arctan \frac{1}{2}$ ($2d$) and $\phi \leq \pi - \arctan \frac{4}{3}$ ($3d$), and that a unique minimizer exists.

Anisotropic Energies. The energy formulations presented so far are isotropic as distortions are measured in every direction in an uniform way. We model anisotropic behavior by replacing the isotropic Frobenius norm with an anisotropic norm $||\cdot||^2_{\mathbf{B}}$ defined by a rank-2 tensor field of symmetric positive definite matrices $\mathbf{B}$: $||\mathbf{A}||^2_{\mathbf{B}} = \mathrm{Tr}\big(\mathbf{A}^{\mathrm{T}}\,\mathbf{B}\,\mathbf{A}\big)$. For example, the pointwise energy (11) then becomes

$$\begin{aligned} e_{\text{CONF}} &= \left|\left|\mathbf{J}^{\mathrm{T}} + \mathbf{J} - \alpha\,\mathbf{I}\right|\right|^2_{\mathbf{B}} \\ &= \left|\left|\mathbf{J}^{\mathrm{T}} + \mathbf{J}\right|\right|^2_{\mathbf{B}} + \mathrm{Tr}\big(-2\alpha\left(\mathbf{J}^{\mathrm{T}}\,\mathbf{B} + \mathbf{B}\,\mathbf{J}\right)\big) + \mathrm{Tr}\big(\alpha^2\,\mathbf{B}\big) \\ &= \left|\left|\mathbf{J}^{\mathrm{T}} + \mathbf{J}\right|\right|^2_{\mathbf{B}} - 4\alpha\ \mathrm{Tr}(\mathbf{B}\,\mathbf{J}) + \gamma\,\alpha^2 \\ &= \left|\left|\mathbf{J}^{\mathrm{T}} + \mathbf{J}\right|\right|^2_{\mathbf{B}} - \frac{4}{\gamma}(\mathrm{Tr}(\mathbf{B}\,\mathbf{J})\,)^2\ , \end{aligned}$$

where we have set $\gamma = \mathrm{Tr}\,\mathbf{B}$ and used the identity $\mathrm{Tr}(\mathbf{A}\,\mathbf{B}) = \mathrm{Tr}(\mathbf{B}\,\mathbf{A})$ together with the solution of $\nabla_\alpha\, e_{\text{CONF}} = 0$, which is $\alpha = \frac{2}{\gamma}\ \mathrm{Tr}(\mathbf{B}\,\mathbf{J})$. In the special case of $\mathbf{B} = \mathbf{I}$ the isotropic case is recovered as then $||\cdot||_{\mathbf{B}} \equiv ||\cdot||_F$ and $\gamma = d$.

4 Discrete Setting

Let $\mathcal{P} = (\mathcal{V}, \mathcal{T}, \mathbf{x})$ be a partition of Ω_d (at a particular time t) with vertices $\mathcal{V}$ and cells $\mathcal{T}$ (triangles for $d = 2$ and tetrahedra for $d = 3$). Furthermore, let $m = |\mathcal{V}|$ denote the number of vertices, and $\mathbf{x}_i \in \mathbb{R}^d$ with $i \in \mathcal{V}$ denote vertex positions. We express a vector field $\mathbf{v}$ as piecewise linear functions on $\mathcal{P}$: $\mathbf{v}$ is given as nodal values $\mathbf{v}_i$, $i \in \mathcal{V}$; we write $\mathbf{v}$ as the a single column vector $\mathbf{v} = \left({\mathbf{v}_1}^{\mathrm{T}}, \ldots, {\mathbf{v}_m}^{\mathrm{T}}\right)^{\mathrm{T}} \in \mathbb{R}^{dm}$. Its piecewise constant Jacobian field is given as matrices $\mathbf{J}_c$ on cells $c \in \mathcal{T}$.

Energy Minimization. In the discrete setting, E_ϕ is a quadratic form in the unknown vector field: $E_\phi(\mathbf{v}) = \mathbf{v}^{\mathrm{T}}\,\mathbf{E}_\phi\,\mathbf{v}$. The matrix $\mathbf{E}_\phi \in \mathbb{R}^{dm\times dm}$ is the symmetric positive definite sparse matrix defining E_ϕ. With the Jacobians being constant on each cell, the coefficients of $\mathbf{E}_\phi$ are the sum of matrices $\mathbf{E}^c_\phi$ that capture the local error E^c_ϕ on cell c as

$$\begin{aligned} E^c_\phi(\mathbf{v}) &= \int_{\Omega_c} w_q(\phi)\ q(\mathbf{x})\ +\ w_r(\phi)\ r(\mathbf{x})\,\mathrm{d}\mathbf{x} \\ &= V_c\left(w_q\left|\left|\mathbf{J}_c + {\mathbf{J}_c}^{\mathrm{T}}\right|\right|^2_F + w_r\,(\mathrm{Tr}\ \mathbf{J}_c)^2\right) \quad = {\mathbf{v}_c}^{\mathrm{T}}\,\mathbf{E}^c_\phi\,\mathbf{v}_c\ . \end{aligned}$$

Here the vector $\mathbf{v}_c \in \mathbb{R}^6$ ($d = 2$) or resp. $\mathbf{v}_c \in \mathbb{R}^{12}$ ($d = 3$) is the concatenation of velocities of the vertices of c, $\mathbf{J}_c$ is the constant Jacobian on c, and V_c is the volume of the cell, triangle area or tetrahedral volume, which weights the constant expressions during integration over the discrete domain $\mathcal{P}$.

We use interpolation constraints on the flow $\mathbf{v}$. This means that the user prescribes trajectories $\gamma_k(t)$ that define the flow of some vertices $k \in \mathcal{V}$. This yields conditions $\mathbf{v}_k(t) = \frac{\mathrm{d}}{\mathrm{d}t}\gamma_k(t)$ as the flow along the trajectories is defined

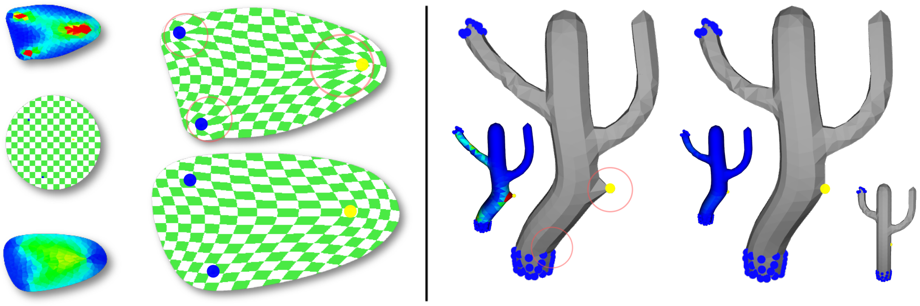


Fig. 3. Smoothness Energy. In $2d$ (left) and $3d$ (right) minimizers of the generalized metric energy are discontinuous near user constraints (fixed ● and handle ● vertices) leading to the highlighted local discontinuous deformations (small images show metric distortions, see Section 6). The smoothness energy term yields smooth vector fields and therefore smooth deformations. Note that both interior (left) and boundary vertices (right) can be constrained.

by their tangents. Not that this includes the special case of "fixed" vertices for which the trajectory is a constant domain point with $\mathbf{v}_k(t) \equiv \mathbf{0}$.

The vector field $\hat{\mathbf{v}}$ minimizing the energy is given as the solution to the linear system $\nabla E_\phi(\mathbf{v}) = \mathbf{0}$ subject to these constraints. In Section 5 we discuss how to setup $\mathbf{E}_\phi$ and solve the arising linear systems efficiently.

Enforcing Smoothness. The derived energies do not enforce smoothness of the solution. This means that even though we obtain a minimizer the residual energy is not distributed smoothly over the domain. In particular, this leads to high concentration of metric error near constrained vertices. Lipman observes this effect for finite ARAP deformations [22]. This problem was also already discussed by Solomon et al. [29]. Their solution consists in a post-process: they solve an additional linear system that diffuses the error to construct smooth vector fields. This consequences: firstly, an additional solving step is required, and secondly, the previously defined constraints can only be satisfied approximately. During time-integration, this approach can leads to significant drift from the user defined trajectories. Moreover, we show that the total resulting deformation error increases unnecessarily (see Section 6).

We take a different approach based on regularization. We define smoothness as the local first order energy variation. This way local deformation errors vary smoothly and do not concentrate, e. g., only at the constrained vertices. The local energy of a cell depends on its constant local Jacobian, i.e., there is variation only on the cell boundaries. Let $c_i, c_j \in \mathcal{T}$ be two neighboring cells with Jacobians $\mathbf{J}_i, \mathbf{J}_j$, and local energy parameters ϕ_i, ϕ_j, respectively. As E^c_ϕ depends only on $\mathbf{J}_c$ we obtain the integrated variation $E^{i,j}_S$ for the pair (c_i, c_j) as

$$E^{i,j}_S(\mathbf{v}, \phi) = B_{i,j} \left|\left| 4 \left(\mathbf{D}^q_{i,j} + {\mathbf{D}^q_{i,j}}^{\mathrm{T}} \right) + 2 \operatorname{Tr} \mathbf{D}^r_{i,j} \, \mathbf{I} \right|\right|^2_F$$

with $\mathbf{D}^s_{i,j} = w_{s_j} \mathbf{J}_j - w_{s_i} \mathbf{J}_i$ and $w_{s_i} = w_s(\phi_i)$ for $s \in \{q, r\}$, and $B_{i,j}$ denotes the length of the common edge of adjacent triangles $(d = 2)$ or the area of the

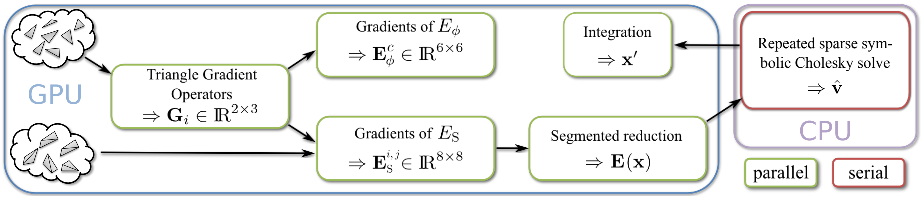

GPU
Triangle Gradient Operators
⇒ $\mathbf{G}_i \in \mathbb{R}^{2\times 3}$
Gradients of E_ϕ
⇒ $\mathbf{E}^c_\phi \in \mathbb{R}^{6\times 6}$
Gradients of E_S
⇒ $\mathbf{E}^{i,j}_S \in \mathbb{R}^{8\times 8}$
Segmented reduction
⇒ $\mathbf{E}(\mathbf{x})$
Integration
⇒ $\mathbf{x}'$
Repeated sparse symbolic Cholesky solve
⇒ $\hat{\mathbf{v}}$
CPU
parallel
serial

Fig. 4. GPU Pipeline. We use the GPU to setup linear systems and perform vector field integration. The linear systems are solved on the CPU using an efficient sparse solver. Operations marked (●) are performed in parallel on the GPU.

common triangle of adjacent tetrahedra ($d = 3$). See Appendix A for a derivation. Note that $E_S^{i,j}$ is quadratic in $\mathbf{v}$. The total discrete smoothness energy is then given by the sum over all adjacent cells

$$E_S(\mathbf{v}, \phi) = \sum_{i,j \in \mathcal{T}\text{adjacent}} E_S^{i,j}(\mathbf{v}, \phi) \ .$$

This energy has the quadratic form $E_S = \mathbf{v}^{\mathrm{T}}\, \mathbf{E}_S\, \mathbf{v}$ that acts as a regularization term in a weighted total energy in the deformation vector field

$$E(\mathbf{v}, \phi) = E_\phi(\mathbf{v}) + \lambda\, E_S(\mathbf{v}, \phi) \ .$$

Its quadratic form is $E(\mathbf{v}, \phi) = \mathbf{v}^{\mathrm{T}}\, (\mathbf{E}_\phi + \lambda\, \mathbf{E}_S)\ \mathbf{v} := \mathbf{v}^{\mathrm{T}}\, \mathbf{E}\, \mathbf{v}$. Hence, we compute a smooth minimizer of E_ϕ by solving $\nabla\, \hat{\mathbf{v}}^{\mathrm{T}}\, \mathbf{E}\, \hat{\mathbf{v}} \ = \ \mathbf{0}$. We use a factor of $\lambda = 0.1$ in all our examples. Figure 3 illustrates the effect of using the regularization term E_S in two and three dimensions. Note that smoothness of the vector field is preserved for handles in the interior as well as on the boundary of the domain.

Shape Integration. We are left with the problem of solving an ODE numerically: we solve $\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{x}_i(t) = \hat{\mathbf{v}}(\mathbf{x}_i(t), t)$ with initial vertex positions $\mathbf{x}_i(0)$, $i \in \mathcal{V}$, using a standard ODE solver. For every evaluation of the vector field the energy minimizing flow $\hat{\mathbf{v}}$ is computed from the current shape configuration.

5 Implementation

Modeling Metaphor. In contrast to finite deformation methods, continuous deformations require velocities as boundary constraints (cf. [29,21]). There are various ways to prescribe velocities. In the simplest case they are provided as zero vectors for fixed vertices. Translations can be modeled by constant velocities, rotations can be expressed by linear flows. A fairly general and intuitive approach is the definition of a space-time curve that acts acts as trajectory, i.e., velocity along the curve is defined as the tangent vector. It is easy to extend this approach to define a laminar “bundle” of trajectories that are defined by the Frenet frame of a single curve [12].

In addition to constraints, the user can model nonhomogeneous energies by changing the scalar parameter ϕ and the tensor field $\mathbf{B}$. This can be done globally or locally per cell, e.g., by a spatial blend (see Figure 7). From the users point of view, near-isometric deformations often behave similarly to stiff real materials, while near-conformal deformations often exhibit strong scaling components towards smaller and larger area.

GPU Implementation. We use the GPU to accelerate certain steps of our deformation algorithm. Figure 4 provides an overview, with matrix dimensions given for the $2d$ case. In summary, the setup of the linear system and the integration of vertices are performed in parallel on the GPU, and the sparse system is solved on the CPU. First, all triangle gradient operators $\mathbf{G}_i$ are computed in parallel. These are required to compute the energy terms E_ϕ and E_S. Then the energy gradients are computed in parallel by exploiting symmetry for each cell and for each pair of adjacent cells. The results are summed by a parallel segmented reduction operation to give the final linear system. The sparse system is downloaded to CPU memory, where it is solved using a state- of-the-art sparse Cholesky solver that uses a precomputed symbolic factorization and an approximate minimum degree preordering to reduce fill-in [9]. In our experiments this direct system solve is up to four times faster than solving the linear system on the GPU using an iterative sparse solver. Compared to a pure CPU implementation using the GPU is up to three times faster. This is because the cost for system setup are significant as multiple systems need to be solved during integration. Finally, shape integration along the optimal flow $\hat{\mathbf{v}}$ is performed on the GPU. We use a standard fourth-order Runge-Kutta integrator with adaptive step size control.

6 Analysis and Results

Energy Comparison. We evaluate the angle and volume quality of deformations using the following error terms

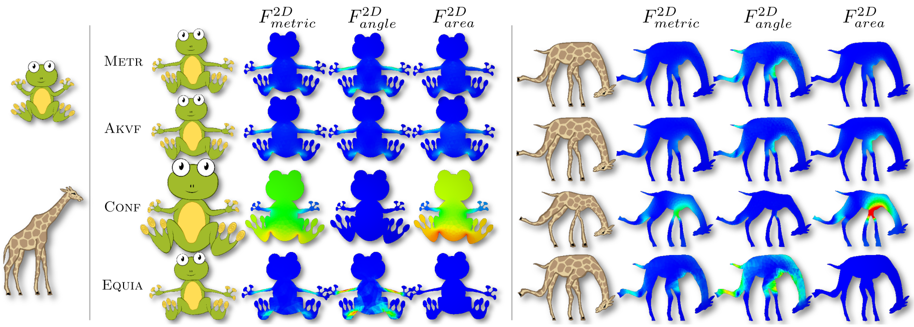
$$F^{2D}_{angle} = \sum_{c\in\mathcal{T}_2} \rho_c \left(\frac{\sigma^1_c}{\sigma^2_c} + \frac{\sigma^2_c}{\sigma^1_c} - 2 \right) \qquad F^{3D}_{angle} = \sum_{c\in\mathcal{T}_3} \rho_c \left(\sum_{(j,k)\in P_3} \left(\frac{\sigma^j_c}{\sigma^k_c} + \frac{\sigma^k_c}{\sigma^j_c} \right) - 6 \right)$$

$$F^{2D}_{area} = \sum_{c\in\mathcal{T}_2} \rho_c \left(\sigma^1_c\sigma^2_c + \frac{1}{\sigma^1_c\sigma^2_c} - 2 \right) \; F^{3D}_{volume} = \sum_{c\in\mathcal{T}_3} \rho_c \left(\sigma^1_c\sigma^2_c\sigma^3_c + \frac{1}{\sigma^1_c\sigma^2_c\sigma^3_c} - 2 \right) .$$

These errors are established in the literature (see, e.g., [29]) and are based on Equations (1-3). Here σ^j_c is the jth singular value of the Jacobian of the map of triangle or tetrahedra c, $\rho_c = {}^{V_c}\!/\!\sum_{j\in\mathcal{T}} V_j$ with triangle area or tetrahedral volume V_c, and $P_3 = \{(1,2), (2,3), (3,1)\}$. To measure metric errors we introduce the error terms

$$F^{2D}_{metric} = \sum_{c\in\mathcal{T}_2} \rho_c \left(\left(\sigma^1_c - 1\right)^2 + \left(\sigma^2_c - 1\right)^2 - \frac{1}{4}\left(\sigma^1_c - \sigma^2_c\right)^2 \right) \tag{14}$$

$$F^{3D}_{metric} = \sum_{c\in\mathcal{T}_3} \rho_c \left(\sum_{j=1}^{3} \left(\sigma^j_c - 1\right)^2 - \frac{1}{5} \sum_{(j,k)\in P_3} \left(\sigma^j_c - \sigma^k_c\right)^2 \right) \tag{15}$$



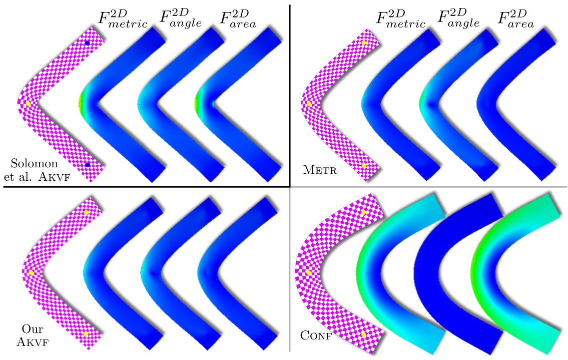
Energy	F^{2D}_{metric}		F^{2D}_{angle}		F^{2D}_{area}	
METR	0.0228	0.0216	0.036	0.0692	0.0155	0.0137
AKVF	0.028	0.0273	0.0211	0.0443	0.033	0.028
CONF	0.752	0.0902	0.0001	0.0098	0.978	0.273
EQUIA	0.0297	0.0489	0.1102	0.1864	$5 \cdot 10^{-6}$	$4 \cdot 10^{-6}$

Fig. 5. $2D$ Energy Evaluation. Two initial models (left) are deformed using the same boundary constraints for the different energy types. The plots visualize color coded local errors. The CONF error plots for the frog are downscaled to 75% size. The table gives total errors for each method and each model (frog left / giraffe right column).

that are the weighted sum of solutions of integrals of the form of (6) and (9). The difference to the previous derivation of energies is that the integrand is no pointwise infinitesimal quadratic length variation but the pointwise finite quadratic length variation induced by the map and integrated along all possible directions. In the optimal case all error terms are zero.

Figure 5 shows error values and error visualizations for two planar deformations. For the EQUIA results we used $\phi = \arctan 2^{-9}$. The METR energy generating near-isometric vector fields achieves lowest metric and area distortions at the cost of change of angle. Deformations based on the AKVF energy show better angle preservation compared to METR, but they also show greater errors in length and area variation. Almost no angle distortion is introduced by ACAP vector fields based on the CONF energy, however, this is at the cost of area errors. The opposite is true for the EQUIA deformation that introduces almost no area error but instead a large angular error. The experiment confirms that the parameter ϕ corresponds to balance between metric and area preservation on the one side and angular preservation on the other side (cf. Figure 2). No deformation can preserve all properties at the same time.

In Figure 6 we compare our energies (including AKVF) to the original method in [29] that uses "soft" handle constraints and achieves smoothness by a error diffusion. Note that the softly constrained vertices drifted significantly. To compensate for this effect and for a fair comparison, the constraints were selected such that the trajectories of all handles (●) end in the (optimally) fixed soft handles (●) after the same integration time. The two AKVF and the METR results look visually similar, however, all three error values indicate that our AKVF approach using a problem dependent smoothing term achieves deformation of lower error.



Energy	F^{2D}_{metric}	F^{2D}_{angle}	F^{2D}_{area}
[29] Akvf	0.084	0.075	0.080
Our Akvf	0.048	0.059	0.042
Metr	0.041	0.091	0.018
Conf	0.227	0.001	0.338

Fig. 6. Akvf Comparison. A symmetric strip deformation is used to evaluate our energies and compare them to the original AKVF formulation [29] that uses no energy based smoothing and only soft constraints. The color coded images visualize local error components, which are all scaled equally.

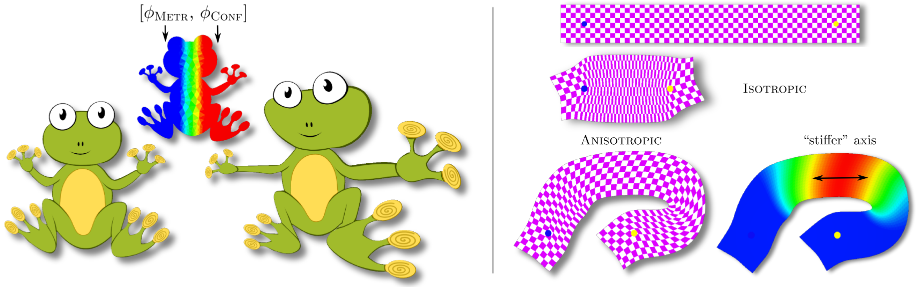


Fig. 7. Left: A nonhomogeneous parameter ϕ is given as a scalar field in form of a blend from ϕ_{Metr} on the left side of the frog's domain to ϕ_{Conf} on the right. Deformation constraints are defined symmetrically on both sides of the model. Right: An isotropic deformation compared to a deformation of anisotropic material with a locally "stiffer" axis direction. Equal deformation constraints were applied in both cases together with ϕ_{AKVF}.

Figure 10 shows frames of the animation of a volumetric mesh. The wings of the eagle were deformed symmetrically using three-dimensional AMAP and ACAP deformations as well as with an Equia vector field with $\phi = \arctan 2^{-9}$. Again, the conformal energy trades volumetric error for angle preservation while the isometric energy has better length and volume preservation properties at the expense of angular distortion. Best volume preservation but also most angular distortions is achieved by the equiareal deformation. This is also reflected in the error values of the animation steps (I) and (III) given in the table. Note that we also included the Akvf errors of the same $3d$ deformation, which is not shown.

Energy Parameter. Figure 7 (left) shows an example where a nonhomogeneous parameter ϕ is prescribed as a scalar field on the domain. In the example we use a spatial blend from near-isometric (left side) to conformal (right side). Defining symmetric constraints shows the nonhomogeneous effect of ϕ.

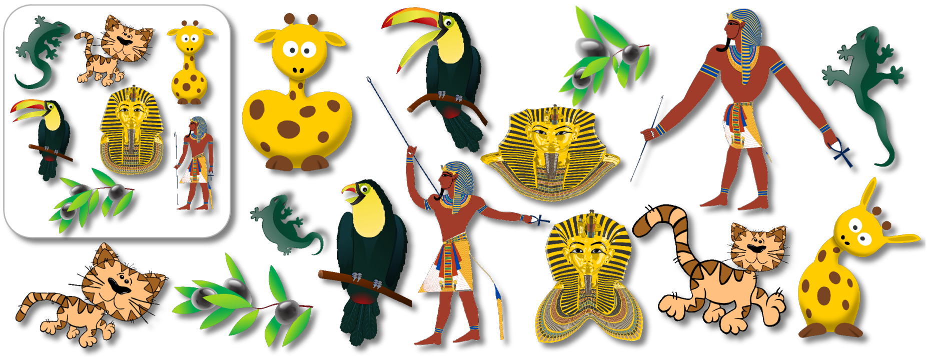
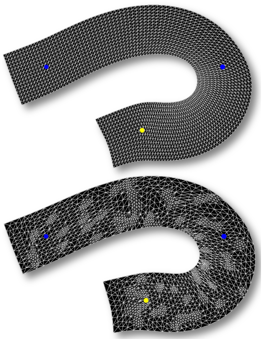


Fig. 8. $2D$ Deformation Examples. The triangulated models in the box were deformed using AMAP and ACAP vector fields.

In Figure 7 (right) we demonstrate the effect of using an anisotropic material compared to an isotropic one. Specifically, we define a region in the center of the strip that is “stiffer” along one prescribed axis modeled by a corresponding tensor field **B**. The material modification leads to two near-isometric deformations of different characteristics for the same boundary constraints.

Independence of Tessellation. The discretization of our energies are integrated measures on the discretized domain. We expect that the resulting deformations are independent of the partition, i.e., of the tessellation, as long as there are enough degrees of freedom available to represent the constrained deformation. This is confirmed by all our experiments (see the adjacent Figure for an example).



Modeling Results. Figure 8 shows initial $2d$ shapes and two deformed versions using AMAP and ACAP deformations. The model size ranges from 5k to 11k vertices and the modeling time was below four minutes in every example. More $2d$ examples are shown in Figures 1(a), 1(b), and 5. Besides the animation in Figure 10 we show further tetrahedral deformations in Figure 9. Again we have the initial shapes together with AMAP and ACAP deformations. The meshes contain between $1,500$ to 5000 vertices. Again modeling time of an inexperienced user ranges from a few seconds to a few minutes. None of our tests suffered from stability issues, not even for extreme deformations. In particular, we didn’t observe local folds or flips. This is due to the fact that the energy minimizing vector field generally does not vanish.

Timings. The following tables lists timings of our approach for the smallest and largest models, $2d$ and $3d$, respectively. We measured the time for the initial fac-

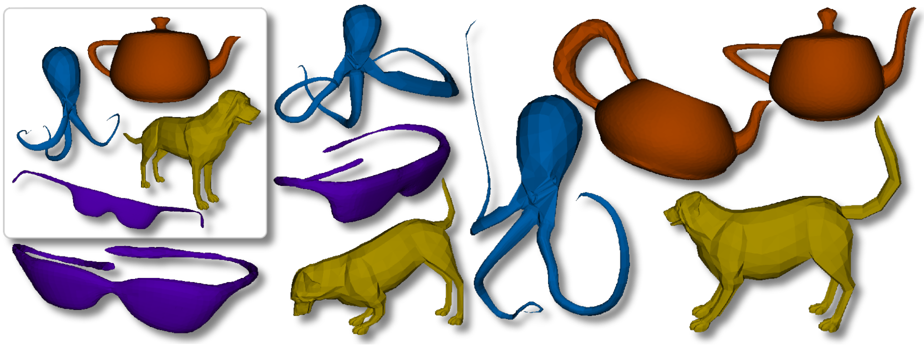


Fig. 9. 3D Deformation Examples. The tetrahedral models in the box were deformed using AMAP and ACAP vector fields to give the shown results.

torization of the linear system (t_1), system setup time (t_2), and time to solve the system (t_3), and the total time T to perform ten consecutive integration steps. Compared to a sole CPU implementation our parallel system setup using the GPU is up to three times faster even though the system has to be transferred to the CPU before solving it. Timings were measured on an AMD Phenom II 955 quad-core CPU with 3.2GHz clock speed equipped with a NVIDIA GTX 560 Ti GPU with 2 GB of memory. Our approach is interactive for reasonably sized models. However, as is true for most solvers of nonlinear measures, also has much higher computational costs compared to linear methods. Please see also the accompanying video.

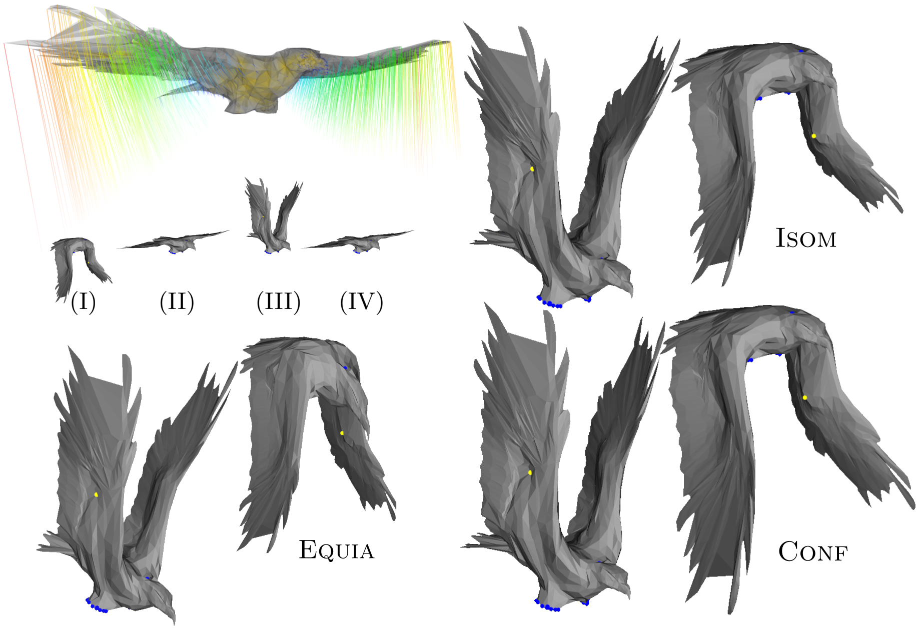
Model $(\lvert\mathcal{T}\rvert, \lvert\mathcal{V}\rvert)$	t_1(ms)	t_2(ms)	t_3(ms)	T(s)
Toucan (5.6k, 9.6k)	230	6	40	1.9
Cat (11k, 18k)	642	13	68	3.8
Octopus (1.5k, 5k)	168	19	51	2.9
Teapot (5k, 16k)	424	31	118	6.3

7 Discussion

Most existent geometrically-motivated approaches either optimize for near-isometric [17,31,21,29] or for near-conformal deformations [33,34]. In contrast to this our generalization combines both extrema in an integral formulation, and it can be applied the same way in $2d$ and in $3d$!

The results in Solomon et al. [29] indicate that their AKVF approach yields deformations of superior quality compared to related methods. It seems that for many shape deformation tasks near-isometry is the desired property. Our approach not only is able to reproduce their results but it shows even better behavior, getting even closer to isometric maps. At the same time our method is less complex and more efficient as we achieve smoothness by a regularization, enable true interpolation constraints and use standard ODE solvers.

Our main feature, however, is the ability to control the deformation by the parameter ϕ from conformal to equiareal with AKVF and our definition of near-isometric deformations in between. To the best of our knowledge, this approach



(I)/(III)	Metr		Conf		Akvf		Equia	
F^{3D}_{metric}	0.061	0.069	0.172	0.072	0.083	0.091	0.386	0.211
F^{3D}_{angle}	0.356	0.421	0.022	0.027	0.305	0.357	2.357	1.367
F^{3D}_{volume}	0.009	0.007	0.273	0.051	0.028	0.012	$9 \cdot 10^{-4}$	$4 \cdot 10^{-4}$

Fig. 10. 3D Eagle Deformation. A tetrahedral model of an eagle (top left, with instantaneous vector field) was deformed in an animation of steps I-IV. The closeups show intermediate steps for different energies. Note the greatest volume of the conformal deformation.

is the first that provides such range of deformations in a single and concise mathematical framework. Our Metr energies are an alternative way to measure deviation from isometry. In a direct comparison to AKVF deformations these new near-isometric energies show a better area preservation at the expense of a slightly higher angle deviation. We also note that close-to-equiareal deformations have not been studied thoroughly in the literature. Even though it is well known that these maps are not uniquely defined, for $\phi \to 0$ we get close to this limit, and even though the condition of system matrices degrades we obtain meaningful results.

Relation to Linear Elasticity. Our geometrically-motivated energy formulation can be related to physically-based theory of linear elasticity (see, e. g., [26]). This formalism assumes that a rest configuration with material coordinates $\mathbf{X}$ is deformed by a displacement field $\mathbf{u} = \mathbf{x} - \mathbf{X}$ into a deformed shape $\mathbf{x}$. The deformation results in an isotropic internal potential deformation energy $\psi = \mu ||\boldsymbol{\epsilon}||_F^2 + \frac{\lambda}{2} \operatorname{Tr} \boldsymbol{\epsilon}^2$ that depends on a local strain tensor $\boldsymbol{\epsilon}$, which is usually defined using the deformation gradient tensor $\mathbf{F} = \nabla_{\mathbf{X}} \mathbf{x} = \mathbf{H} + \mathbf{I}$ with displacement gradient tensor $\mathbf{H} = \nabla_{\mathbf{X}}\,\mathbf{u}$. Here μ and λ are the physical Lamé constants, which are related to stiffness and volume preservation, respectively. For small displacement gradients the Lagrangian finite strain tensor $\boldsymbol{\epsilon}$ can be approximated by the linearized small strain tensor

$$\boldsymbol{\epsilon} := \frac{1}{2}\left(\mathbf{F}^{\mathrm{T}}\mathbf{F} - \mathbf{I}\right) = \frac{1}{2}\left(\mathbf{H} + \mathbf{H}^{\mathrm{T}} + \mathbf{H}^{\mathrm{T}}\mathbf{H}\right) \overset{||\mathbf{H}||_F \ll 1}{\approx} \frac{1}{2}\left(\mathbf{H} + \mathbf{H}^{\mathrm{T}}\right) \ ,$$

and the linear elasticity energy becomes

$$\psi = \frac{\mu}{4}\left|\left|\mathbf{H} + \mathbf{H}^{\mathrm{T}}\right|\right|_F^2 + \frac{\lambda}{2}\operatorname{Tr}\mathbf{H}^2 \ .$$

It measures the potential energy of the deformed shape $\mathbf{x}$ relative to the rest configuration $\mathbf{X}$, which is different to our instantaneous deformation energies that doesn't use the notion of a rest post. Still, in the limit of instantaneous deformations, i. e., $\mathbf{x} \to \mathbf{X}$, we have $\mathbf{H} \to \mathbf{J}$, i. e., the displacement gradient becomes the vector field Jacobian. Both the physical linear elasticity model and our geometrically motivated energy formulation therefore coincide in this case with the relation of parameters $w_q = \frac{\mu}{4}$ and $w_r = \frac{\lambda}{2}$. However, as we don't need to consider deformed shapes in different coordinate systems, our instantaneous approach doesn't require artificial regularization method like corotational elasticity [28] to correct artifact of diverging coordinate systems $\mathbf{X}$ and $\mathbf{x}$. Additionally, our instantaneous approach is unconditionally stable and we can therefore apply standard explicit ODE solvers for integration and require no, e. g., implicit integration. Moreover, this derivation shows that as-Killing-as-possible deformations [29] can be regarded as a geometric special case of physically-based linear elasticity that describes near-isometric materials. Additionally, in this work we provide the parameters for materials that show near-conformal behavior, which might not always give physically plausible results.

Limitations and Future Work. Nonlinear methods are expensive. Although we use a parallelized GPU implementation it is impossible to outperform linear methods in terms of computation time. This is a general drawback, and the user must decide if the additional cost is worthwhile. Still, all shown examples were modeled interactively.

Until now we consider only space deformations. So far, there is no extension to the explicit deformation of surfaces that are embedded in $3d$ space. This is because the vector field Jacobians capture only the tangential components of the vector field. They do not measure variations normal to the surface. For the same reason approximate Killing vector fields are, until now, considered only tangentially for triangle meshes [30]. We believe that this is an interesting direction for future research direction.

So far we consider only the initial value problem for path constrained deformation. It is much harder to solve the boundary value problem to find an energy minimizing path between two poses, e.g., for interpolation between poses (cf. [21,13]). We would like to use our generalized energy in such settings.

We furthermore want to study the application of generalized energies for parametrization applications to allow locally varying conformal / equiareal parametrization. Moreover, the eigen-spectrum of the energy might allow a multiresolution (in the parameter ϕ) segmentation of shapes.

8 Conclusions

In this paper we introduce a novel generalized metric energy for continuous shape deformation. We obtain near-isometric and near-conformal deformations by integration of as-isometric-as-possible and as-conformal-as-possible vector fields. Our approach works for two and three dimensions, we have applied it for deformations of triangular and tetrahedral meshes. For the discretization of the energy we have introduced a first order smoothness criterion based on the energy itself that guarantees vector field differentiability at handle vertices. Our implementation uses the GPU to achieve interactivity and we support nonhomogeneous and anisotropic behavior.

A Smoothness Regularization

Given a generalized pointwise energy

$$e_\phi := w_q \left|\left|\mathbf{J} + \mathbf{J}^{\mathrm{T}}\right|\right|_F^2 + w_r \left(\mathrm{Tr}\,\mathbf{J}\right)^2$$

the derivative w.r.t. $\mathbf{J}$ is given by

$$\begin{aligned}\frac{\partial}{\partial \mathbf{J}} e_\phi &= \frac{\partial}{\partial \mathbf{J}}\ w_q \left|\left|\mathbf{J} + \mathbf{J}^{\mathrm{T}}\right|\right|_F^2 + w_r \left(\mathrm{Tr}\,\mathbf{J}\right)^2 \\ &= \frac{\partial}{\partial \mathbf{J}}\ w_q\ \mathrm{Tr}\left(\mathbf{J} + \mathbf{J}^{\mathrm{T}}\right)^{\mathrm{T}}\left(\mathbf{J} + \mathbf{J}^{\mathrm{T}}\right) + w_r \left(\mathrm{Tr}\,\mathbf{J}\right)^2 \\ &= \frac{\partial}{\partial \mathbf{J}}\ w_q \left(2\ \mathrm{Tr}\,\mathbf{J}^{\mathrm{T}}\mathbf{J} + 2\ \mathrm{Tr}\,\mathbf{J}\mathbf{J}\right) + w_r\ \mathrm{Tr}\,\mathbf{J}\,\mathrm{Tr}\,\mathbf{J} \\ &= 4\,w_q\,\mathbf{J} + 4\,w_q\,\mathbf{J}^{\mathrm{T}} + w_r\left(\mathbf{I}\ \mathrm{Tr}\,\mathbf{J} + \mathrm{Tr}\,\mathbf{J}\ \mathbf{I}\right) \\ &= 4\,w_q\left(\mathbf{J} + \mathbf{J}^{\mathrm{T}}\right) + 2\,w_r\ \mathrm{Tr}\,\mathbf{J}\ \mathbf{I}\ .\end{aligned}$$

Given two neighboring cells $c_i, c_j \in \mathcal{T}$ with local energy parameters ϕ_i, ϕ_j, we enforce smoothness by minimizing the variation of derivatives along a common edge ($n = 2$) or face ($n = 3$) $\mathcal{B}_{i,j} = \Omega_i \cap \Omega_j$, i. e., we regularize by minimizing smoothness energies of the form

$$\begin{aligned}E_S^{i,j}(\mathbf{v}, \phi) &= \int_{\mathcal{B}_{i,j}} \left|\left|\frac{\partial}{\partial \mathbf{J}_j} e_{\phi_j} - \frac{\partial}{\partial \mathbf{J}_i} e_{\phi_i}\right|\right|_F^2 \mathrm{d}\mathbf{x} \\ &= B_{i,j} \left|\left|\frac{\partial}{\partial \mathbf{J}_j} e_{\phi_j} - \frac{\partial}{\partial \mathbf{J}_i} e_{\phi_i}\right|\right|_F^2 \\ &= B_{i,j} \left|\left|4\left(\mathbf{D}_{i,j}^q + {\mathbf{D}_{i,j}^q}^{\mathrm{T}}\right) + 2\ \mathrm{Tr}\,\mathbf{D}_{i,j}^r\ \mathbf{I}\right|\right|_F^2\end{aligned}$$

with $\mathbf{D}_{i,j}^s = w_{s_j}\,\mathbf{J}_j - w_{s_i}\,\mathbf{J}_i$ and $w_{s_i} = w_s(\phi_i)$ for $s \in \{q, r\}$, and common edge length or face area $B_{i,j} = |\mathcal{B}_{i,j}|$. The integral can be simplified this way as the Jacobians are constant on each cell.

References

1. Alexa, M., Cohen-Or, D., Levin, D.: As-rigid-as-possible shape interpolation. In: Proc. SIGGRAPH. pp. 157–164 (2000)
2. Ben-Chen, M., Butscher, A., Solomon, J., Guibas, L.: On discrete killing vector fields and patterns on surfaces. In: Proc. SGP. pp. 1701–1711 (2010)
3. Ben-Chen, M., Weber, O., Gotsman, C.: Variational harmonic maps for space deformation. ACM Trans. Graph. 28(3), 34:1–34:11 (2009)
4. Botsch, M., Pauly, M., Gross, M., Kobbelt, L.: Primo: coupled prisms for intuitive surface modeling. In: Proc. SGP. pp. 11–20 (2006)
5. Botsch, M., Sorkine, O.: On linear variational surface deformation methods. IEEE TVCG 14(1), 213–230 (2008)
6. do Carmo, M.P.: Differential Geometry of Curves and Surfaces. Prentice–Hall (1976)
7. Cashman, T.J., Hormann, K.: A continuous, editable representation for deforming mesh sequences with separate signals for time, pose and shape. Comput. Graph. Forum 31(2), 735–744 (2012)
8. Chao, I., Pinkall, U., Sanan, P., Schröder, P.: A simple geometric model for elastic deformations. ACM Trans. Graph. 29(4), 38:1–38:6 (2010)
9. Chen, Y., Davis, T.A., Hager, W.W., Rajamanickam, S.: Algorithm 887: Cholmod, supernodal sparse cholesky factorization and update/downdate. ACM Trans. Math. Softw. 35(3), 22:1–22:14 (2008)
10. Floater, M.S., Hormann, K.: Surface parameterization: a tutorial and survey. In: Advances in Multiresolution for Geometric Modelling, pp. 157–186. Springer (2005)
11. von Funck, W., Theisel, H., Seidel, H.P.: Vector field based shape deformations. ACM Trans. Graph. 25(3), 1118–1125 (2006)
12. von Funck, W., Theisel, H., Seidel, H.P.: Explicit control of vector field based shape deformations. In: Proc. Pacific Graphics. pp. 291–300 (2007)
13. Heeren, B., Rumpf, M., Wardetzky, M., Wirth, B.: Time-discrete geodesics in the space of shells. Comp. Graph. Forum 31(5), 1755–1764 (2012)
14. Hormann, K., Greiner, G.: MIPS: An efficient global parametrization method. In: Curve and Surface Design 1999, pp. 153–162. Vanderbilt Press (2000)
15. Hormann, K., Polthier, K., Sheffer, A.: Mesh parameterization: Theory and practice. In: Proc. SIGGRAPH Asia (2008)
16. Hormann, K., Floater, M.S.: Mean value coordinates for arbitrary planar polygons. ACM Trans. Graph. 25(4), 1424–1441 (2006)
17. Igarashi, T., Moscovich, T., Hughes, J.F.: As-rigid-as-possible shape manipulation. ACM Trans. Graph. 24(3), 1134–1141 (2005)
18. Jacobson, A., Baran, I., Popović, J., Sorkine, O.: Bounded biharmonic weights for real-time deformation. ACM Trans. Graph. 30(4), 78:1–78:8 (2011)
19. Joshi, P., Meyer, M., DeRose, T., Green, B., Sanocki, T.: Harmonic coordinates for character articulation. ACM Trans. Graph. 26(3) (2007)
20. Karni, Z., Freedman, D., Gotsman, C.: Energy-based image deformation. In: Proc. SGP. pp. 1257–1268 (2009)
21. Kilian, M., Mitra, N.J., Pottmann, H.: Geometric modeling in shape space. ACM Trans. Graph. 26(3), 64:1–64:8 (2007)
22. Lipman, Y.: Bounded distortion mapping spaces for triangular meshes. ACM Trans. Graph. 31(4), 108:1–108:13 (2012)
23. Lipman, Y., Levin, D., Cohen-Or, D.: Green coordinates. ACM Trans. Graph. 27(3), 78:1–78:10 (2008)

24. Liu, L., Zhang, L., Xu, Y., Gotsman, C., Gortler, S.J.: A local/global approach to mesh parameterization. In: Proc. SGP. pp. 1495–1504 (2008)
25. Martinez Esturo, J., Rössl, C., Theisel, H.: Continuous deformations by isometry preserving shape integration. In: Proc. Curves and Surfaces 2010, pp. 456–473. LNCS 6920, Springer (2010)
26. McAdams, A., Zhu, Y., Selle, A., Empey, M., Tamstorf, R., Teran, J., Sifakis, E.: Efficient elasticity for character skinning with contact and collisions. ACM Trans. Graph. 30(4), 37:1–37:12 (2011)
27. Minka, T.P.: Old and new matrix algebra useful for statistics. Tech. rep., MIT (2001)
28. Müller, M., Dorsey, J., McMillan, L., Jagnow, R., Cutler, B.: Stable real-time deformations. In: Proc. SCA. pp. 49–54 (2002)
29. Solomon, J., Ben-Chen, M., Butscher, A., Guibas, L.: As-killing-as-possible vector fields for planar deformation. Comput. Graph. Forum 30(5), 1543–1552 (2011)
30. Solomon, J., Ben-Chen, M., Butscher, A., Guibas, L.: Discovery of intrinsic primitives on triangle meshes. Comput. Graph. Forum 30(2), 365–374 (2011)
31. Sorkine, O., Alexa, M.: As-rigid-as-possible surface modeling. In: Proc. SGP. pp. 109–116 (2007)
32. Sýkora, D., Dingliana, J., Collins, S.: As-rigid-as-possible image registration for hand-drawn cartoon animations. In: Proc. NPAR. pp. 25–33 (2009)
33. Weber, O., Ben-Chen, M., Gotsman, C., Hormann, K.: A complex view of barycentric mappings. Comput. Graph. Forum 30(5), 1533–1542 (2011)
34. Weber, O., Myles, A., Zorin, D.: Computing extremal quasiconformal maps. Comput. Graph. Forum 31(5), 1679–1689 (2012)