EUROGRAPHICS 2009 / P. Dutré and M. Stamminger
(Guest Editors)

Volume 28 (2009), Number 2

Positional, Metric, and Curvature Control for Constraint-Based Surface Deformation

Michael Eigensatz Mark Pauly

Applied Geometry Group
ETH Zurich

Abstract

We present a geometry processing framework that allows direct manipulation or preservation of positional, metric, and curvature constraints anywhere on the surface of a geometric model. Target values for these properties can be specified point-wise or as integrated quantities over curves and surface patches embedded in the shape. For example, the user can draw several curves on the surface and specify desired target lengths, manipulate the normal curvature along these curves, or modify the area or principal curvature distribution of arbitrary surface patches. This user input is converted into a set of non-linear constraints. A global optimization finds the new deformed surface that best satisfies the constraints, while minimizing adaptable measures for metric and curvature distortion that provide explicit control of the deformation semantics. We illustrate how this approach enables flexible surface processing and shape editing operations not available in current systems.

Categories and Subject Descriptors (according to ACM CCS): Computer Graphics [I.3.5]: Computational Geometry and Object Modeling—

1. Introduction

Effective algorithms for surface deformation are of central importance in digital geometry processing. One of the most popular interaction metaphors allows the user to select subsets of the model as a control handle and specify an affine transformation for each handle region. The deformed surface is then computed such that the resulting positional constraints are satisfied, while preserving important properties of the original shape.

While intuitive and easy-to-learn, certain tasks remain difficult to achieve with handle-based interaction. For example, preserving or explicitly modifying first or second order properties of the surface, such as lengths, areas, or curvature, is cumbersome when deforming a shape by specifying positional constraints only. To address this problem, we introduce a new framework for surface processing that combines ideas from non-linear positional deformation [GHDS03, BPGK06] with recent work on curvature-domain shape processing [ESP08]. Our method provides advanced control of the surface metric and allows the user to directly constrain positions, lengths, areas, and curvatures. Target values for these quantities can be specified anywhere on the model, both point-wise or integrated over embedded curves and surface patches. Our system solves for a deformation of the surface that best satisfies the user constraints, while preserving the original shape of the model as well as possible. We argue that the concept of *shape preservation* comes in different flavors that strongly depend on the user's editing intent. To provide the necessary design flexibility, we enable explicit control over the shearing (conformal distortion), stretching (area distortion) and bending (curvature distortion) induced by the deformation mapping. We define the corresponding non-linear energies for each of these properties and show how a consistent discretization can be obtained for surfaces represented by triangle meshes. In addition, we show how significant improvements in performance can be achieved using a novel formulation of the conformal energy.

Related work. There exists a vast amount of literature on techniques to edit and manipulate geometry and we refer to recent surveys such as [AS06], [BPK*07], or [BS08] for an overview. Positional constraints are used successfully for shape manipulation with curves [WW92, SF98], multiresolution editing [ZSS97, KCVS98, BK04], and surface deformation based on differential coordinates [SCOL*04, LSCO*04, YZX*04, LSLCO05]. The latter achieve intuitive shape deformations by manipulating derived properties such as mesh gradients or Laplacian coordinates and reconstructing the deformed surface by solving a linear system. Sketch-based tools have become popular due to an intuitive shape design process centered around a sketching metaphor [IMT99]. The work by Nealen et al. [NISA07] allows the user to freely draw and modify control curves on the model and reconstructs the surface defined by these curve constraints using an optimization approach.

Various papers extended the idea of positional constraints to also support a more direct specification of derived properties. For example, constraint based design tools (e.g. [MS92]) have successfully used differential normal or curvature constraints at fixed points in space to create a set of interpolating parameterized surface patches from scratch. Miura and colleagues [MCW01] allow the user to scale derivatives (i.e. local lengths) of analytic curves and surfaces. Tosun and co-workers propose a system for shape optimization using reflection lines that allows modifying surfaces by specifying a target reflection function gradient [TGRZ07]. This approach has recently been extended in [GZ08] to support stroke-based editing of the shaded image to drive the deformation of the surface.

Our work is also related to optimization methods in surface parameterization [HLS07]. Ben-Chen and colleagues [BCGB08] and Springborn et al. [SSP08] have proposed algorithms for computing a conformal mapping based on metric scaling and curvature prescription. While conceptually similar in the sense that these methods 'deform' the surface into a plane while preserving conformality, the specific context of surface parameterization warrants a fundamentally different approach than our more general shape deformation setting.

Our implementation is based on the curvature-domain shape processing framework of Eigensatz and co-workers [ESP08] that showed how a variety of geometry processing operations can be performed by direct prescription of principal curvatures. We extend this method and provide new tools for direct control of local and global metric properties of the surface including a more flexible formulation to manage the trade-off between areal and angular metric distortion. Additional *directional* control on surface bending is obtained by prescribing the normal curvature of the surface along embedded curves. We illustrate with a number of examples how this approach enables novel editing functionality that complements and extends existing surface deformation tools. Furthermore, we show how a different conformal energy avoids the mesh-dependency of the formulation of [ESP08] and present a discretization method for this energy that significantly improves the convergence of the optimization.

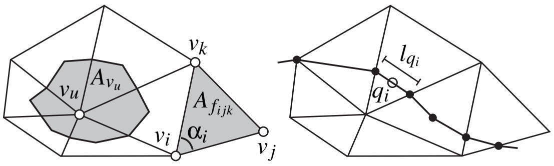


Figure 1: *Vertex and face areas, inner triangle angle (left). Length and center point of a discrete path segment (right).*

2. Problem Formulation

We formulate surface deformation as a nonlinear optimization problem, where the optimization constraints are derived from the user's input. The algorithm solves for the vertex positions of the deformed mesh such that the user constraints are satisfied and the shape of the original is preserved as well as possible. Shape preservation is encoded in metric and curvature energies that can be weighted to control the relative importance of area, angle, and curvature distortion. Besides shape preservation, these energies also provide direct control for shape deformation. For example, we can locally scale the metric to grow or shrink the shape, or control surface bending by prescribing principal curvatures.

Similar to existing handle-based deformation tools, our system offers position constraints that are computed from the displacement of a control handle. In addition, the user can draw curves and surface patches onto the surface and modify their global length and area, respectively, or change the normal curvature of the surface along the selected curves.

2.1. Surface Energies

We derive the discrete energies from their continuous counterparts. The surface $\mathcal{S}$ is discretized with a triangle mesh $\mathcal{M} = (\mathcal{V}, \mathcal{E}, \mathcal{F})$, where $\mathcal{V} = \{v_i\}$ denotes the set of vertices, $\mathcal{E} = \{e_{ij}\}$ the edge set, and $\mathcal{F} = \{f_{ijk}\}$ the face set with $1 \le i, j, k \le |\mathcal{V}|$. The position of vertex v_i is given by $\mathbf{v}_i \in \mathbb{R}^3$. A prime denotes quantities of the deformed mesh, all other values are computed on the input surface.

Metric energies. Let $J_\mathbf{p}$ be the Jacobian of the deformation function restricted to the tangent space at a point $\mathbf{p}$ on the surface $\mathcal{S}$. The metric distortion can be measured through the local anisotropic scaling that is encoded in the singular values σ_1, σ_2 of $J_\mathbf{p}$ [dC76]. The product $\sigma_1\sigma_2$ quantifies the change in area and the ratio σ_1/σ_2 measures the angular distortion. We use the symmetric energies

$$E_{areal_\mathbf{p}} = \sigma_1\sigma_2 + \frac{1}{\sigma_1\sigma_2} \quad (1)$$

$$E_{conf_\mathbf{p}} = \frac{\sigma_1}{\sigma_2} + \frac{\sigma_2}{\sigma_1} \quad (2)$$

to measure the areal and conformal distortion, respectively. As we will illustrate below, a weighted sum of these two energies provides more flexibility to control the deformation semantics compared to the method of [ESP08], where the use of either a conformal or an isometric distortion measure was proposed. If both E_{areal} and E_{conf} are weighted equally, their sum corresponds to an isometric distortion measure. In the discrete setting both energies are constant for each triangle f and can be written as [Hor01]

$$E_{areal_f} = \frac{A_f}{A_{f'}} + \frac{A_{f'}}{A_f} \tag{3}$$

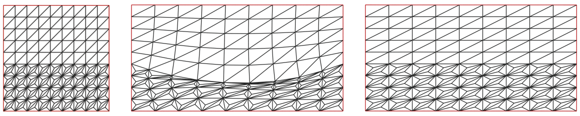
$$E_{conf_f} = \frac{\cot(\alpha_i)\|\mathbf{e}'_{jk}\|^2 + \cot(\alpha_j)\|\mathbf{e}'_{ik}\|^2 + \cot(\alpha_k)\|\mathbf{e}'_{ij}\|^2}{2A_{f'}}, \tag{4}$$

where A_f denotes the area of triangle $f = f_{ijk}$, α_i, α_j, and α_k are the inner angles of f at the corresponding vertices, and $\mathbf{e}_{ij} = \mathbf{v}_j - \mathbf{v}_i$ is the edge vector between the corresponding vertices (Figure 1). Summing the per-triangle energies over the entire mesh $\mathcal{M}$ with surface area $A_{\mathcal{M}}$ leads to the total areal energy

$$E_{areal_{\mathcal{M}}} = \frac{1}{A_{\mathcal{M}}} \sum_{f \in \mathcal{F}} A_f E_{areal_f}, \tag{5}$$

and, analogously, the total conformal energy $E_{conf_{\mathcal{M}}}$. Energies similar to E_{areal} have been used for surface parameterization and physically based shell models [DMK03, GHDS03]. The conformal energy E_{conf} corresponds to the MIPS energy introduced by Hormann in the context of mesh parameterization [Hor01, HG00]. To the best of our knowledge, this energy has never been used for surface deformation, partly because the corresponding optimization is rather involved. We resolve this issue by introducing a novel formulation of the energy in Section 3.

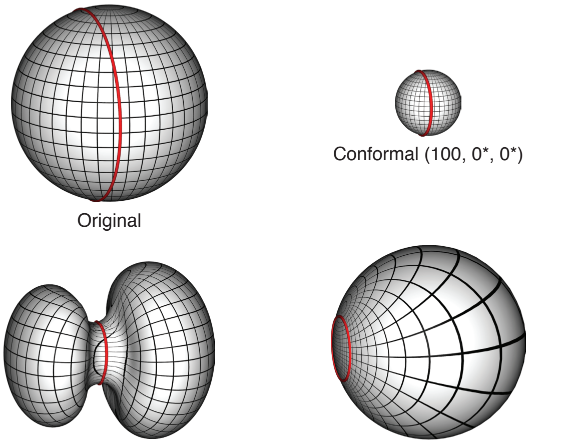
A distinct advantage of E_{conf} is illustrated in the figure below. Compared to the conformal energy of [ESP08], which measures the squared difference of inner triangle angles, our formulation is not sensitive to the specific discretization of the mesh. When stretching the non-uniformly tessellated patch along the horizontal axis, the local distortion is independent of the triangulation, which makes the method robust under re-sampling of the surface.



Original Conformal energy of [ESP08] Conformal energy of Eq. 4

Curvature energy. In addition to the surface metric, we explicitly control curvature using the discrete curvature energy of [ESP08]:

$$E_{pc_{\mathcal{M}}} = \sum_{v_i \in \mathcal{V}} A_{v_i} \left[(\kappa'_{1,i} - \kappa_{1,i})^2 + (\kappa'_{2,i} - \kappa_{2,i})^2 \right], \tag{6}$$



Original

Conformal (100, 0*, 0*)

Equiareal (1, 100, .001)

Curvature preserving (1, 0*, 100)

Figure 2: *Effect of different weights for the local surface energies. As a modeling constraint, the target length of the red curve is set to one third of its original value. The numbers in brackets denote the weighting terms* $(k_{conf}, k_{areal}, k_{pc})$*, where* 0^* *denotes a very small contribution (we use* $0^* = 10^{-5}$*) of the corresponding energy added for regularization.*

where $\kappa_{1,i}$ and $\kappa_{2,i}$ denote the signed maximal and minimal principal curvatures at vertex v_i, estimated using the method of Cohen-Steiner and Morvan [CSM03].

Combined surface energy. The total surface energy on a mesh is computed as a weighted combination of the metric and curvature energies

$$E_{m_{\mathcal{M}}} = k_{areal} E_{areal_{\mathcal{M}}} + k_{conf} E_{conf_{\mathcal{M}}} + k_{pc} E_{pc_{\mathcal{M}}} \tag{7}$$

with scalar weights k_{areal}, k_{conf}, and k_{pc}. The effect of different choices for these weights is illustrated in Figure 2. A dominant conformal weight leads to a uniform scaling of the sphere. A dominant areal term yields a constricting deformation with fairly strong distortion of angles and curvature, while a dominant curvature term leads to the curve 'sliding' along the sphere. This simple example illustrates how our approach supports different notions of *shape preservation*. The user can adapt the behavior of the optimization to preserve properties of the original shape that are important in the specific application context, enabling flexible surface processing and shape design.

2.2. Modeling Constraints

The surface energies defined above aim at preserving important properties of the input surface, i.e., control how the shape resists to change dictated by the user's modeling constraints. However, the energies can also be used to drive the deformation by replacing the values of the original shape by arbitrary target values. One of the unique features of our

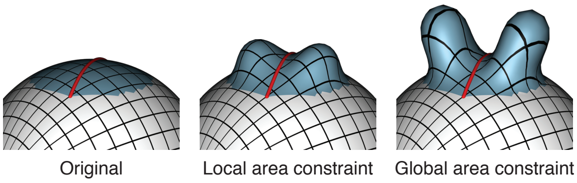


Figure 3: *Comparison of local vs. global area constraints. The area of the blue patch is scaled by a factor of 2.5 while constraining the length of the red curve to remain constant. Specifying the area change locally for each triangle does not offer enough degrees of freedom to satisfy both constraints. The global area constraint for the entire patch offers more flexibility to satisfy the constraints.*

method is the ability to explicitly control the surface metric, i.e., set constraints for lengths and areas on the surface. This can be done locally by specifying target values for each triangle in Equations 3 and 4, or as global least-squares constraints over curves and surface patches, as described below. In the following, user-specified target values are indicated by a hat symbol.

Area constraints. A global area constraint for a patch $\mathcal{P} \subset \mathcal{S}$, represented by a set of faces $\mathcal{F}_{\mathcal{P}} \subseteq \mathcal{F}$, is defined as

$$E_{area_{\mathcal{P}}} = \frac{1}{A^2_{\mathcal{P}}}\left(A_{\mathcal{P}'} - \hat{A}_{\mathcal{P}}\right)^2, \tag{8}$$

i.e., the total area $A_{\mathcal{P}'}$ of the patch $\mathcal{P}$ on the deformed surface should be equal to the specified target value $\hat{A}_{\mathcal{P}}$ provided by the user. In contrast to local area constraints, scaling the global area offers more degrees of freedom to satisfy conflicting user constraints, as illustrated in Figure 3.

Length constraints. Another integrated measure is the length of a curve, discretized as a piecewise linear path $\mathcal{D} \subset \mathcal{M}$ (Figure 1). The corresponding energy is defined as

$$E_{length_{\mathcal{D}}} = \frac{1}{l^2_{\mathcal{D}}}\left(l_{\mathcal{D}'} - \hat{l}_{\mathcal{D}}\right)^2, \tag{9}$$

where $l_{\mathcal{D}}$ denotes the total length of the polyline.

Normal curvature constraints. Using the curvature energy of Equation 6, the user can directly modify principal curvatures as proposed in [ESP08]. However, access to principal curvatures does not provide *directional control* of surface bending. We enable more flexible editing by directly prescribing the normal curvature of the surface along arbitrary tangent directions. In order to do so, the user can freely draw curves on the surface and specify the desired target values for normal curvature (Figure 4).

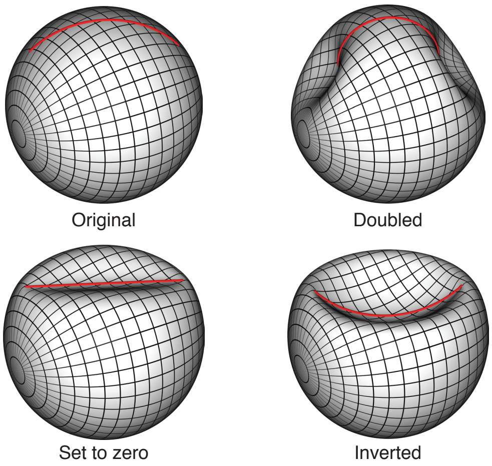


Figure 4: *Deformation of a sphere by prescribing normal curvature of the surface along the marked curve.*

We measure signed normal curvature as the change of surface normal along the curve. Continuous surface normal vectors are defined using barycentric interpolation of vertex normals. At a given point $\mathbf{q}_i$ on the curve we compute the osculating plane spanned by the surface normal vector $\mathbf{n}_i$ at $\mathbf{q}_i$ and the curve's tangent vector. For a discrete approximation of normal curvature $\kappa_{\mathcal{D}}$ we consider a small curve segment i of length l_{q_i} around $\mathbf{q}_i$ (Figure 1) and project the interpolated normal vectors at the ends of the curve segment onto the osculating plane (Figure 5), leading to

$$\kappa_{\mathcal{D},i} = \frac{2\sin(\beta_i/2)}{l_{q_i}}, \tag{10}$$

where β_i is the angle between the projected surface normals $\mathbf{n}^*_{i,1}$, $\mathbf{n}^*_{i,2}$. The normal curvature energy is then defined as

$$E_{nc_{\mathcal{D}}} = \sqrt{A_{\mathcal{M}}}\sum_i l_{q_i}\left(\kappa_{\mathcal{D}',i} - \hat{\kappa}_{\mathcal{D},i}\right)^2, \tag{11}$$

Position constraints. Our system also offers position constraints based on the standard handle paradigm. Since the vertex positions are the unknowns of the optimization, fixing vertices to a specified position in space is trivial. The

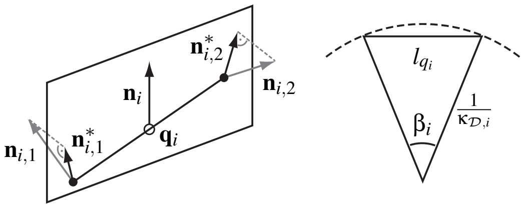


Figure 5: *Left: Discrete path segment with center point* $\mathbf{q}_i$*, interpolated mesh normals* $\mathbf{n}$ *and projections* $\mathbf{n}^*$ *onto the osculating plane. Grey elements do not lie in the plane. Right: Approximation of the normal curvature.*

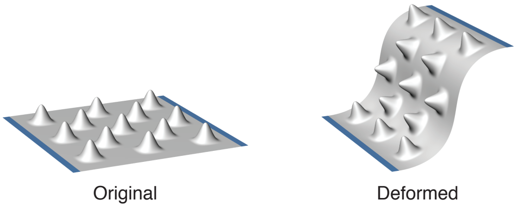


Figure 6: *Our system incorporates the handle-based deformation metaphor using position constraints (marked in blue). Local detail is accurately preserved due to the non-linear surface energies.*

affected vertices are displaced to the target location and the corresponding variables are removed from the optimization. To provide more flexibility when accommodating conflicting constraints we also include soft constraints, formulated using the least squares energy

$$E_{pos_{\mathcal{M}}} = \frac{1}{A^2_{\mathcal{M}}} \sum_{v_i \in \mathcal{V}} A_{v_i} \left\|\mathbf{v}'_i - \hat{\mathbf{v}}_i\right\|^2, \tag{12}$$

where A_{v_i} is the barycentric area around vertex v_i (Figure 1). Since all of the above energies except E_{pos} are invariant to rigid transformations, for all the examples in the paper we either fixed a number of vertices to their original position or added soft position constraints to all vertices using a small weight $k_{pos} = 0.001$. Figure 6 shows a test case for position constraints that illustrates how our non-linear surface energies lead to intuitive detail preservation, i.e., accurate rotation of surface detail even for purely translation-based handle displacements (cf. Figure 2 of [BPGK06]).

3. Optimization

Editing constraints and shape preservation control are encoded in the energies introduced in the previous section. We find the corresponding deformed surface by solving for the vertex positions $\mathbf{v}^*_1 \dots \mathbf{v}^*_n$ such that

$$\mathbf{v}^*_1 \dots \mathbf{v}^*_n = \underset{\mathbf{v}'_1 \dots \mathbf{v}'_n}{\operatorname{argmin}} \sum_i k_i E_i, \tag{13}$$

where E_i and k_i is short-hand notation for the different energies and weighting constants defined above. We solve this nonlinear least squares problem using a Gauss-Newton solver [MNT04] which enables superlinear convergence without the need for second derivatives. The initial values of the variables are the vertex positions of the input surface. In all our experiments the results converge within about fifteen iteration steps. The derivation of analytic expressions for the gradients of the energies is discussed in the appendix. If curvatures are manipulated, the optimization falls into the category of composite non-smooth optimization as described in [ESP08]. We use the techniques proposed there to approximate the gradients. For further details on the implementation of a nonlinear least squares solver, we direct the interested reader to the comprehensive literature on this topic, e.g. [GMW89, Bjö96, MNT04].

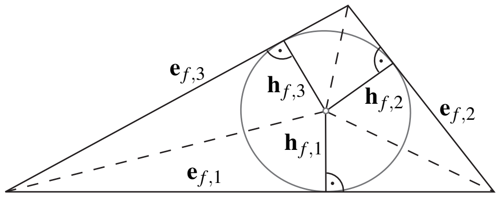


Figure 7: *Virtual subdivision of a triangle f avoids negative cotangent weights, since all angles are $\leq \pi/2$.*

Conformal energy. In our experiments we found that the convergence of the optimization is largely dominated by the conformal energy E_{conf}. In fact, a direct implementation of Equation 4 leads to prohibitively slow convergence. Since the rather involved formulas for the curvature energies make the use of an advanced solver that requires second derivatives undesirable, we make use of the following observation to obtain a practical implementation: As written in Equation 4, E_{conf} is the sum of three rational quadratic terms. Since the Gauss-Newton solver locally approximates the objective function with a quadratic, splitting the energy into three separate terms significantly improves the convergence [MNT04]. However, this separation is only admissible, if the single terms (i.e. the cotangent weights) are all positive as required by the Gauss-Newton Solver. To address this issue, we propose a re-formulation for the conformal energy based on a virtual subdivision of a triangle using its incircle center to generate six smaller triangles as depicted in Figure 7. By construction, negative cotangent weights cannot occur, hence we can perform the above split into separate terms. By exploiting symmetry, right angles, and invariance of the conformal energy under triangle subdivision, the resulting formula simplifies to a sum of six *positive* rational quadratics for each triangle:

$$E_{conf_f} = \sum_{u=1}^{3} \left(\frac{A_f}{3\|\mathbf{e}_{f,u}\|^2} \frac{\|\mathbf{e}'_{f,u}\|^2}{A_{f'}} + \frac{\phi_f^2}{12A_f} \frac{\|\mathbf{h}'_{f,u}\|^2}{A_{f'}} \right), \tag{14}$$

where ϕ_f denotes the circumference of the triangle, $\mathbf{h}_{f,u}$ are the vectors connecting points on the edges and the center of the incircle on the triangle of the original mesh (Figure 7). On the deformed mesh, these points are expressed with the barycentric coordinates of the points on the original mesh. The barycentric coordinates are computed as a preprocessing step at the beginning of the optimization. This reformulation is essential to achieve a practical speed of convergence with a Gauss-Newton type solver, as we demonstrate with a simple example: Scaling the normal curvature of the sphere along the curve depicted in Figure 4 takes 9 iteration steps using the virtual subdivision compared to 108 iterations without it. Since the terms of the conformal energy are better approximated by quadratic functions when using

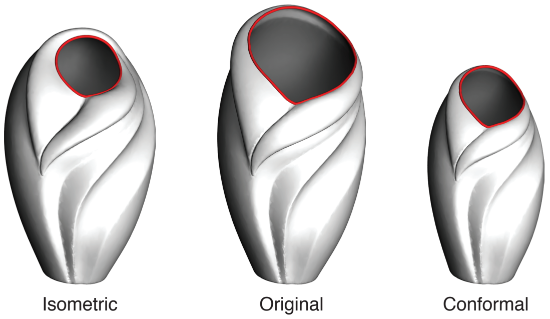


Figure 8: *The vase has been edited by reducing the length of the marked curve by a factor of two while keeping the bottom fixed. Different notions of shape preservation lead to different editing semantics.*

the subdivision, the stepsize has only to be scaled by an average factor of 0.43 compared to 0.00027 in the unsubdivided formulation. This reduces the cost of stepsize control in our implementation. For this simple test case, the overall speedup factor achieved by virtual subdivision was 47. For more complex operations, these performance differences become even more extreme which renders the unsubdivided energy impractical for use with a Gauss-Newton type optimization. Note that the straightforward way of subdivision by dividing the obtuse angle with a line orthogonal to the opposite edge is not suitable since the virtual triangles can become arbitrarily thin leading to numerical instabilities.

4. Results

Figures 2 to 6 illustrate specific properties of our framework for simple geometric shapes. In the following we show various constrained deformations performed on more complex models focusing on editing operations that are difficult to achieve with traditional, purely position-based approaches.

Figure 8 shows an editing operation using length constraints. The user draws a curve along the rim of the vase and interactively modifies the desired target length while the bottom of the vase is kept fixed. Two settings for the surface energies, one aiming at isometric, the other at conformal deformation, illustrate how different notions of shape preservation lead to different editing semantics. A benefit of our approach is that the user has explicit control of the shape-preserving energies and can thus easily adapt the deformation to a specific application context.

Figure 9 shows the creation of a cartoon horse. The belly and flanks have been enhanced by scaling the area of the blue patches. Length constraints are used to constrict the waist and enlarge the head. The normal curvature for each point of the curve along the spine has been set to the curve average, leading to a deformation that pushes the spine towards a circular arc.

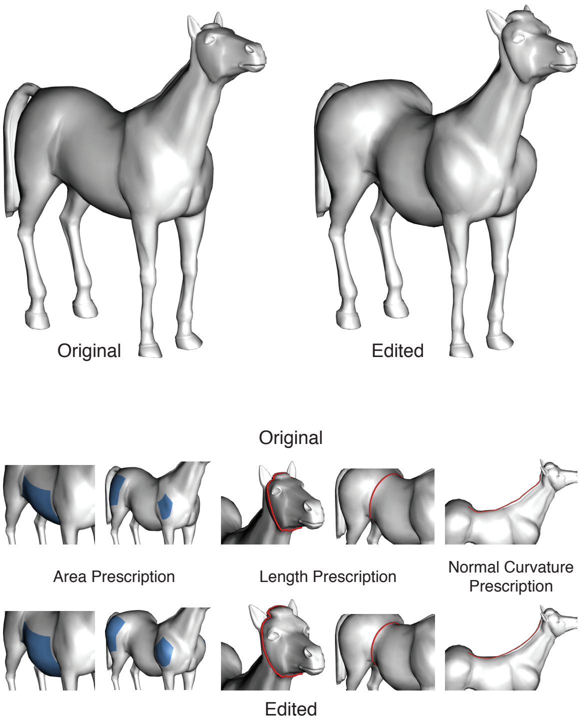


Figure 9: *The shape of the horse model has been edited by direct manipulation of length, normal curvature, and area of different curves and surface patches.*

Figure 10 shows various editing operations on the fandisk model. Direct control of metric and curvature is particularly useful in a CAD context, where the semantics of the model are often directly linked to lengths, areas, and curvature. In the special case of fixing the curvature along a curve to zero, it might be more desirable to constrain the curvature of the curve itself instead of the normal curvature of the surface, which significantly simplifies computation. This technique was used to generate the bottom left result of Figure 10. Figure 11 illustrates a potential application that we intend to explore in future work. We can evolve the fandisk towards a sphere by setting all principal curvatures to a constant. With a high conformal energy term, this yields a quasi-conformal spherical parameterization of the model that can be customized with the use of additional metric constraints. Conceptually, this approach is similar to recent methods for planar conformal mappings [BCGB08, SSP08].

Discussion. An important feature of our approach is the systematic definition of all shape preservation and constraint energies based on the following three requirements: (i) *Convergence*: the discrete energies should converge to the continuous ones for a suitable refinement of the mesh, (ii) *Local symmetry*: energies should be locally invariant when interchanging deformed and original surface, (iii) *Scale invari-*

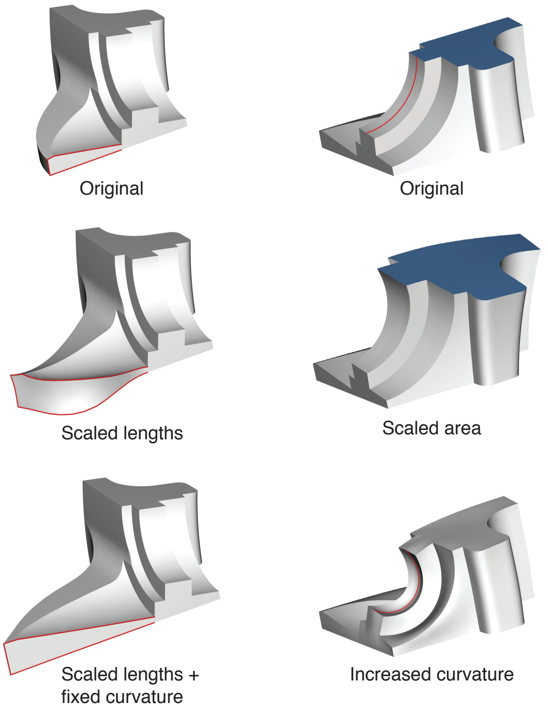


Figure 10: *Processing the fandisk model. Length, normal curvature, and area of the marked curves and patch have been scaled by a factor of two in the corresponding images. For the bottom left image, the curvature of the curves has additionally been constrained to remain zero for each of the marked segments.*

ance: energies should be invariant under global re-scaling. The last property is achieved using global normalization terms, while local symmetry follows immediately from the definition of the energy terms. Cohen-Steiner and Morvan have proved convergence for their estimation of principal curvatures [CSM03] that we apply to evaluate Equation 6. The area of a set of triangles naturally converges to the area of the corresponding patch if a smooth parameterized surface is sampled regularly with increasing density to generate a mesh. The same argument holds for the derived metric energies since they are constant for a triangle face and converge to the functions evaluated at a point of the smooth surface, if the three vertices converge to that point. Similarly, convergence for the measures of normal curvature and length of a curve is assured if the sampling density of the curve increases.

Currently the major limitation of our method is speed. Ideally, the system should provide immediate feedback when the user modifies a constraint. Our current prototype implementation can only achieve interactive response for such edits on relatively small meshes. The curve bending on the fandisk with 6477 vertices takes less than 10 seconds on a 2.4 GHz Intel Core 2 Duo with 2GB of memory (currently, only the sparse linear system solver for the Gauss-Newton optimization exploits two-core parallelism). The edits on the horse and the vase (both around 8000 vertices) take 8 to 150 seconds each. We see two main approaches that can potentially lead to substantial performance improvements. A hierarchical representation of the constraints and the surface can be used to implement multi-level solvers in the spirit of multi-grid methods. In addition, we can exploit multi-core or GPU parallelism to speed up the computation. With these types of optimizations in place, we believe that realtime performance is within reach.

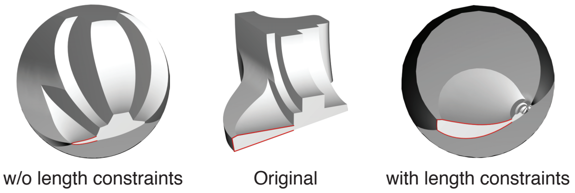


Figure 11: *Spherical conformal parameterization computed by prescribing constant curvature. On the right the marked curves are constrained to keep their original length.*

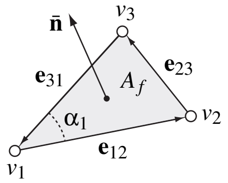
5. Conclusion

We have introduced an optimization framework that allows direct manipulation of derived surface properties. The user can modify lengths, areas and bending of the surface by drawing curves and patches on the surface and manipulating their metric or curvature properties. Local surface measures provide additional control on the deformation semantics. Our approach complements existing methods for shape manipulation and leads to versatile surface deformations for shape design.

Acknowledgments. We would like to thank Mirela Ben-Chen, Mario Botsch, Bálint Miklòs, Robert Sumner, and Camille Wormser for fruitful discussions about this research. Special thanks goes to Sandro Löschhorn for his exceptional help with parts of the implementation. This work has been supported by the Swiss National Science Foundation.

Appendix

Derivatives. We derive analytic expressions for the derivatives of different energies with respect to the vertex positions. For the principal curvature energy, we follow the descriptions given in [ESP08]. By applying the extended chain rule, the derivatives of all energies introduced in this paper can be broken down to weighted sums of derivatives of triangle areas, lengths of path/mesh edges, and angles between two incident edges (or vectors in general). Since the extended chain rule is a standard operator we concentrate on



providing expressions for these three basic components using the notation shown of the figure above: The gradient of the face area can be computed as

$$\frac{\partial}{\partial \mathbf{v}_1} A_f = \frac{1}{2}(\bar{\mathbf{n}} \times \mathbf{e}_{23}), \quad (15)$$

where $\bar{\mathbf{n}}$ is the normalized face normal. The gradients of the edge lengths are

$$\frac{\partial}{\partial \mathbf{v}_1}\|\mathbf{e}_{12}\| = -\bar{\mathbf{e}}_{12} \qquad \frac{\partial}{\partial \mathbf{v}_2}\|\mathbf{e}_{12}\| = \bar{\mathbf{e}}_{12} \quad (16)$$

where $\bar{\mathbf{e}}_{12}$ is the normalized edge. For an angle between two edges, the derivatives can be computed as

$$\begin{gathered}\frac{\partial}{\partial \mathbf{v}_2}\alpha_1 = \frac{\mathbf{e}_{12} \times \bar{\mathbf{n}}}{\|\mathbf{e}_{12}\|^2} \qquad \frac{\partial}{\partial \mathbf{v}_3}\alpha_1 = \frac{\mathbf{e}_{31} \times \bar{\mathbf{n}}}{\|\mathbf{e}_{31}\|^2} \\ \frac{\partial}{\partial \mathbf{v}_1}\alpha_1 = -\frac{\mathbf{e}_{12} \times \bar{\mathbf{n}}}{\|\mathbf{e}_{12}\|^2} - \frac{\mathbf{e}_{31} \times \bar{\mathbf{n}}}{\|\mathbf{e}_{31}\|^2}\end{gathered} \quad (17)$$

References

[AS06] ANGELIDIS A., SINGH K.: Space deformations and their application to shape modeling. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Courses* (2006).

[BCGB08] BEN-CHEN M., GOTSMAN C., BUNIN G.: Conformal flattening by curvature prescription and metric scaling. *Computer Graphics Forum 27*, 2 (2008).

[Bjö96] BJÖRCK Å.: *Numerical Methods for Least Squares Problems*. SIAM, Philadelphia, 1996.

[BK04] BOTSCH M., KOBBELT L.: An intuitive framework for real-time freeform modeling. *ACM Transactions on Graphics 23*, 3 (2004).

[BPGK06] BOTSCH M., PAULY M., GROSS M., KOBBELT L.: Primo: coupled prisms for intuitive surface modeling. In *Proceedings of SGP* (2006).

[BPK*07] BOTSCH M., PAULY M., KOBBELT L., ALLIEZ P., LÉVY B., BISCHOFF S., RÖSSL C.: Geometric modeling based on polygonal meshes. In *SIGGRAPH '07: ACM SIGGRAPH 2007 Courses* (2007).

[BS08] BOTSCH M., SORKINE O.: On linear variational surface deformation methods. *IEEE Transactions on Visualization and Computer Graphics 14*, 1 (2008).

[CSM03] COHEN-STEINER D., MORVAN J.-M.: Restricted delaunay triangulations and normal cycle. In *Symposium on Computational Geometry* (2003).

[dC76] DO CARMO M. P.: *Differential Geometry of Curves and Surfaces*. Prentice Hall, 1976.

[DMK03] DEGENER P., MESETH J., KLEIN R.: An adaptable surface parameterization method. In *Proceedings of 12th Int. Meshing Roundtable* (2003).

[ESP08] EIGENSATZ M., SUMNER R. W., PAULY M.: Curvature-domain shape processing. *Computer Graphics Forum 27*, 2 (2008).

[GHDS03] GRINSPUN E., HIRANI A. N., DESBRUN M., SCHRÖDER P.: Discrete shells. In *Proceedings of SCA* (2003).

[GMW89] GILL P. E., MURRAY W., , WRIGHT M.: *Practical Optimization*. Academic Press, London, 1989.

[GZ08] GINGOLD Y., ZORIN D.: Shading-based surface editing. *ACM Transactions on Graphics 27*, 3 (2008).

[HG00] HORMANN K., GREINER G.: MIPS: An efficient global parametrization method. In *Curve and Surface Design: Saint-Malo 1999*. 2000.

[HLS07] HORMANN K., LEVY B., SHEFFER A.: Mesh parameterization: theory and practice. In *SIGGRAPH '07: ACM SIGGRAPH 2007 Courses* (New York, NY, USA, 2007), ACM.

[Hor01] HORMANN K.: *Theory and Applications of Parameterizing Triangulations*. PhD thesis, Department of Computer Science, University of Erlangen, 2001.

[IMT99] IGARASHI T., MATSUOKA S., TANAKA H.: Teddy: A sketching interface for 3d freeform design. In *Proceedings of SIGGRAPH* (1999).

[KCVS98] KOBBELT L., CAMPAGNA S., VORSATZ J., SEIDEL H.-P.: Interactive multi-resolution modeling on arbitrary meshes. In *Proceedings of SIGGRAPH* (1998).

[LSCO*04] LIPMAN Y., SORKINE O., COHEN-OR D., LEVIN D., RÖSSL C., SEIDEL H.-P.: Differential coordinates for interactive mesh editing. In *Proceedings of SMI* (2004).

[LSLCO05] LIPMAN Y., SORKINE O., LEVIN D., COHEN-OR D.: Linear rotation-invariant coordinates for meshes. *ACM Transactions on Graphics 24*, 3 (2005).

[MCW01] MIURA K., CHENG F., WANG L.: Fine tuning: curve and surface deformation by scaling derivatives. In *Proceedings of Pacific Graphics* (2001).

[MNT04] MADSEN K., NIELSEN H., TINGLEFF O.: *Methods for Non-Linear Least Squares Problems*. Tech. rep., Technical University of Denmark, 2004.

[MS92] MORETON H. P., SÉQUIN C. H.: Functional optimization for fair surface design. In *Proceedings of SIGGRAPH* (1992).

[NISA07] NEALEN A., IGARASHI T., SORKINE O., ALEXA M.: FiberMesh: Designing freeform surfaces with 3D curves. *ACM Transactions on Grapphics 26*, 3 (2007).

[SCOL*04] SORKINE O., COHEN-OR D., LIPMAN Y., ALEXA M., RÖSSL C., SEIDEL H.-P.: Laplacian surface editing. In *Proceedings of SGP* (2004).

[SF98] SINGH K., FIUME E.: Wires: a geometric deformation technique. In *Proceedings of SIGGRAPH* (1998).

[SSP08] SPRINGBORN B., SCHRÖDER P., PINKALL. U.: Conformal equivalence of triangle meshes. *ACM Transactions on Graphics 27*, 3 (2008).

[TGRZ07] TOSUN E., GINGOLD Y. I., REISMAN J., ZORIN D.: Shape optimization using reflection lines. In *Proceedings of SGP* (2007).

[WW92] WELCH W., WITKIN A.: Variational surface modeling. In *Proceedings of SIGGRAPH* (1992).

[YZX*04] YU Y., ZHOU K., XU D., SHI X., BAO H., GUO B., SHUM H.-Y.: Mesh editing with poisson-based gradient field manipulation. *ACM Transactions on Graphics 23*, 3 (2004).

[ZSS97] ZORIN D., SCHRÖDER P., SWELDENS W.: Interactive multiresolution mesh editing. In *Proceedings of SIGGRAPH* (1997).