Feature-Based Mesh Editing

Qingnan Zhou, Tino Weinkauf and Olga Sorkine

Courant Institute of Mathematical Sciences, New York University, USA

Abstract

Editing and deformation of irregular meshes have become standard tools in geometric modeling. Most approaches try to preserve low-level differential properties of the surface during editing, whereas the global structure and shape of the features are not explicitly taken into account. In this paper, we introduce a feature-driven editing approach that puts global structural properties of the shape into the center of attention. We start by segmenting the mesh by ridges and valleys and use the so-defined curves and surface regions as intuitive handles for all subsequent editing operations. Our framework supports manipulations the positions and curvature values of the handles and the various mesh regions. In order to preserve the existing features, prevent unwanted appearance of new features, and maintain or manipulate global aspects of the shape, we apply curvature optimization in the affected areas. We show that the combination of feature extraction and curvature optimization leads to an intuitive modeling tool for high-quality surface manipulation.

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

---

1. Introduction

Recent years saw many interactive mesh editing techniques, driven by the proliferation of irregular meshes as the dominant shape representation. Meshes are popular because of their flexibility: they can represent arbitrary shapes at arbitrary resolution and accuracy; yet they lack built-in modeling control mechanisms that allow editing the shape. Much research has concentrated on mesh deformation algorithms where the user can fix arbitrary parts of the surface and manipulate (translate, rotate) manually selected regions called handles. The deformation framework typically tries to preserve low-level differential properties of the edited surface that represent the local details; a lot of knowledge has been gained on how to achieve detail-preserving deformations, often through variational optimization [SB09].

Detail-preserving mesh editing remains a low-level modeling operation, where the global structure and features of the shape are not intentionally observed. Some recent modeling techniques developed more advanced control concepts to manipulate the shape on a higher level, for example through silhouette curves [ZNA07], arbitrary curves that define the shape as a piecewise smooth surface [NISA07] or deform any given surface [SF98], cages that roughly approximate the global shape [LLCO08] or sharp feature curves [GSMCO09]. However, with the exception of *i*WIRES [GSMCO09], the features are used as a manually defined and manipulated control object of a low-level deformation routine, without taking into account the high-level structure and the essence of the shape's features.

*i*WIRES uses feature curves as a way to internally constrain the shape deformation such that the properties of each curve and the inter-relationships are preserved where possible. The feature properties and relationships are a simple ad hoc set chosen for the particular application of editing man made shapes; a specific mechanism is therefore required to optimize each one (such as planarity or parallelism, membership in a shape class like circles or squares). *i*WIRES treats the set of feature curves entirely separately from the surface itself: they are deformed and rearranged in space with no participation of the surface, and are then used as handle constraints to a standard detail-preserving surface deformation.

In this work, we are interested in a modeling framework based on surface features, where the essence of the feature is preserved and/or explicitly controlled/manipulated by the user. We focus on one possible mathematical definition of features: lines where principal curvature is extremal (ridges and valleys). It has been shown that such feature lines capture important high-level information about the shape, allowing e.g. abstraction for depiction [OBS04], analysis and filtering [HP04]. We design a framework to edit a shape via

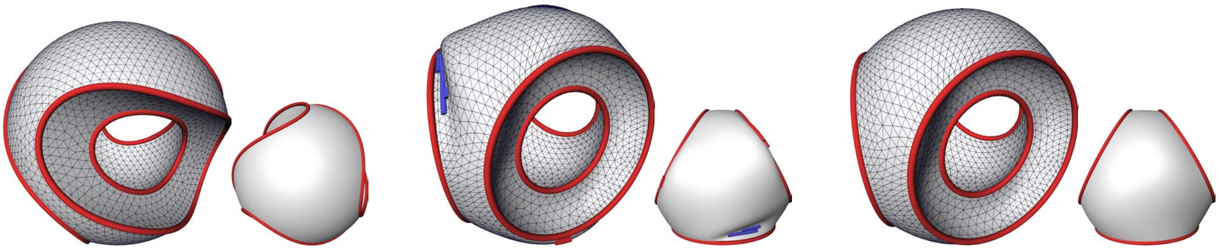

Figure 1: *The feature lines of the sculpt model (left) have been transformed into a planar and circular shape (middle, right). The As-Rigid-As-Possible approach creates a number of new blue valley lines (middle) and leaves the mesh in a rather unintuitively deformed state. After our feature-based curvature optimization, the new model contains only the original, but transformed features and is perfectly round.*

feature manipulation, where the features are preserved as such, namely, the resulting deformed surface has them as its ridges and valley lines.

In our system, the user can edit the feature curves, including freeform deformation, automatic adjustment (straightening, planarization), smoothing and sharpening; features can be erased or added. This induces high-level operations on the surface, allowing to influence the structure of the shape. Moreover, the features often provide a natural partition of the surface into regions; we exploit this fact for providing the ability to localize the editing operations to particular areas, thus providing additional flexibility in controlling the deformation and yet alleviating the need for strictly manual region of interest selection.

Since our chosen definition of feature lines is formulated in terms of principal curvatures of the surface, to control the features we require curvature-based variational deformation. We apply the curvature optimization setup of Eigensatz et al. [ESP08, EP09] that deforms a surface such that the principal curvatures of the result are as close as possible to prescribed curvature values. In our framework, editing operations are converted into target principal curvature values, specifically designed in such a way that the desired feature manipulation can be carried out (in particular, that the curvature along a feature line remains extremal).

2. Mathematical Background

2.1. Feature Extraction

We extract feature lines of the mesh as ridges and valleys following the method of [HPW05]. The principal curvatures are extremal along these lines. Based on a discrete formulation of curvature (see [HPW05] for details), ridges and valleys are defined as zeros of the piecewise linear extremality field. Extremality is the directional derivative of a principal curvature w.r.t. its corresponding principal curvature direction. For a vertex p and the triangles T in its one ring, extremality is defined as

$$e_i(p) = \frac{1}{\text{area}(\text{star}(p))} \sum_{T \ni p} \text{area}(T) \left\langle \nabla \kappa_i(T), \vec{\kappa}_i(p) \right\rangle , \quad (1)$$

where κ_i denotes the minimal/maximal principal curvature $\kappa_{min}/\kappa_{max}$. Two additional requirements have to be fulfilled for a feature line to be *salient*: a ridge line segment through a triangle T is salient, if

$$\left\langle \nabla e_{max}, \textstyle\sum_{p_i \in T} \vec{\kappa}_{max}(p_i) \right\rangle < 0 \quad (2)$$

$$\text{and} \quad \left| \textstyle\sum_{p_i \in T} \vec{\kappa}_{max}(p_i) \right| > \left| \textstyle\sum_{p_i \in T} \vec{\kappa}_{min}(p_i) \right| . \quad (3)$$

Similarly for salient valley lines.

The sign of e_i is undetermined, but can be chosen consistently within a triangle as discussed in [HPW05]. To extract the feature lines reliably in the presence of noise, we apply modified Laplacian smoothing to the extremalities e_i, which takes the undetermined sign of e_i into account. Note that the extracted feature lines often provide a segmentation of the surface into regions; we employ this segmentation for more localized operations on the surface when desired.

2.2. Curvature-Based Optimization

The editing operations that we propose in this paper modify the principal curvature values around feature lines and in surface areas enclosed by feature lines. To obtain a surface adhering to these prescribed principal curvatures, we apply the curvature optimization setup of [ESP08, EP09]. This leads to a non-linear optimization framework involving curvature and conformal surface energies as well as positional constraints. They are described in the following.

Curvature Energy. Given a discrete formulation of the principal curvatures, we strive to comply with the prescribed curvatures in a least squares sense [ESP08]:

$$E_{pc} = \sum_{p_i} A_{p_i} \left[(\kappa'_{max,i} - \kappa_{max,i})^2 + (\kappa'_{min,i} - \kappa_{min,i})^2 \right] , \quad (4)$$

where A_{p_i} denotes the Voronoi area of the vertex p_i. Although curvature does not uniquely define a surface, it is a powerful tool for editing a surface when combined with feature line manipulation, as we show in the next section.

Conformal Energy. To maintain good mesh quality during the curvature optimization, we utilize the conformal energy from [EP09]. For a triangle T it is defined as

$$E_{conf} = \sum_{u=1}^{3} \left(\frac{A_T}{3\|\mathbf{e}_{T,u}\|^2} \frac{3\|\mathbf{e}'_{T,u}\|^2}{A_{T'}} + \frac{\phi_T^2}{12A_T} \frac{\|\mathbf{h}'_{T,u}\|^2}{A_{T'}} \right), \quad (5)$$

where A_T is the area of T, $\mathbf{e}_{T,u}$ are its edges, ϕ_T its circumference, and $\mathbf{h}'_{T,u}$ are vectors perpendicular to the edges and pointing to the center of the inscribed circle of T.

Positional Constraints. We prescribe positions of feature lines as soft constraints in the least squares sense:

$$E_{pos} = \frac{1}{A^2} \sum_{\mathbf{p}_i} A_{\mathbf{p}_i} \|\mathbf{p}'_i - \hat{\mathbf{p}}_i\|^2, \quad (6)$$

where $\hat{\mathbf{p}}_i$ are the desired positions, A is the total area of the mesh and $A_{\mathbf{p}_i}$ the Voronoi area around the vertex $\mathbf{p}_i$. We give E_{pos} a high weight to maintain the positions of the feature lines as prescribed by the user.

Total Energy. The surface optimization is steered by a weighted sum of the above energies and constraints

$$E = k_{pc}E_{pc} + k_{conf}E_{conf} + k_{pos}E_{pos}, \quad (7)$$

where $k_{pc}, k_{conf}, k_{pos}$ are scalar weights. We usually use $k_{pc} = k_{conf} = 1$ and $k_{pos} = 10^3$.

3. Feature-Based Mesh Operations

Based on the previously described feature extraction we have a set of feature lines and possibly a segmentation of the mesh into distinct regions enclosed by them. In the following we propose a number of editing operations to change the curvature values around the feature lines or inside the regions. This allows to sharpen or blur selected edges, add or remove features, prevent the unwanted appearance of new features, and more. Some results of these operations can also be found in the additional material.

In contrast to [ESP08, EP09], where the whole surface is always exposed to the curvature optimization, our system is dedicated to feature-centered control. The consideration of curvature-based features, i.e., ridges and valleys, in our framework allows for an intuitive steering of the curvature optimization framework. Furthermore, limiting the optimization to a smaller subset of the surface makes it considerably faster. The surface segmentation due to ridges/valleys allows for an intuitive definition of such subsets.

3.1. Curvature Editing of Edges

To sharpen a ridge (valley), we multiply κ_{max} (κ_{min}) on the ridge with a factor $s > 1$ and provide a smooth transition to the curvature values in the neighborhood of the ridge, i.e., a band of triangles around the feature line.

To blur a feature curve, we consider a selected ridge or valley line and apply Laplacian smoothing to the curvature values in its neighborhood. After curvature optimization, the curve is blurred while its essence as feature line persists. Figure 2 shows this effect.

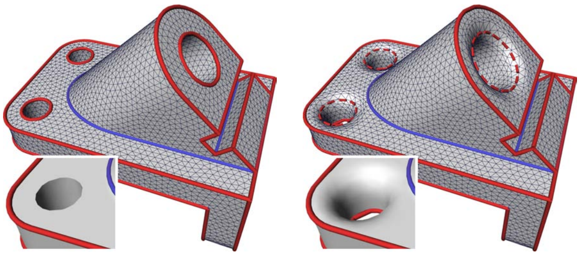

Figure 2: *Laplacian smoothing of curvature values in the neighborhood of a feature line and subsequent curvature optimization allows to blur feature curves. Left: original mesh. Right: mesh with three blurred edges.*

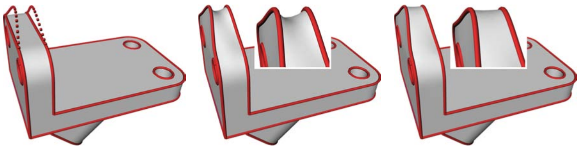

Figure 3: *Left: the original model is deformed using an As-Rigid-As-Possible deformation. Middle: this creates a dent in the surface between the two feature lines that have been used as handles. Right: a subsequent curvature optimization restores the curvature values in this area and cures the artificial distortion.*

3.2. Curvature Editing of Surface Regions

The segmentation of the mesh allows us to select a region enclosed by feature lines and modify its curvature. To do so, we prescribe positional constrains on the enclosing feature lines, which ensures that the rest of the mesh remains untouched. For the curvature values of the selected region we allow the following operations:

- ***Flatten:*** Replace κ_{min} and κ_{max} with constant zero. This flattens the selected region. In order to avoid conflicting constraints, the enclosing feature lines have to be planarized.
- ***Multiplication with a factor:*** Multiply κ_{min} and/or κ_{max} with some factor $s \neq 0$. This changes the shape of the selected region to some degree while keeping the subtleties of the original mesh.
- ***Convex ↔ Concave:*** We set $\kappa'_{min} = -\kappa_{min}$ and $\kappa'_{max} = -\kappa_{max}$ to transform a convex region into a concave one and vice versa.

3.3. Combination with Standard Modeling Tools

The extracted feature lines are intuitive handles for standard surface-based deformation approaches [SB09], for example *As-Rigid-As-Possible* [SA07] or *Laplacian surface editing* [SLCO*04]. Moreover, the underlying segmentation provides a natural way to limit the active deformation area to the surface areas connected to the selected feature line.

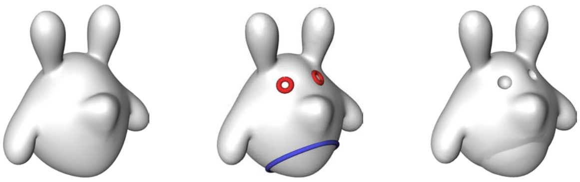

Figure 4: *New features can be added to the mesh by multiplying the curvature in an area around the drawn line with a certain factor.*

Standard deformation approaches do not strive to maintain the properties of surface features and areas: as evident in Figure 3 (middle), the formerly flat area between the two feature lines contains a dent after the As-Rigid-As-Possible deformation. To restore the original properties, we prescribe the original curvature values in the active area on the deformed mesh. The curvature optimization yields the desired result (Figure 3 right). Depending on the application, this result might be more intuitive than the result of the As-Rigid-As-Possible deformation.

Non-manual transformations of feature lines are another powerful modeling tool. In Figure 1 we transformed the feature lines from a non-planar elliptical shape to a planar circular shape. We provide interactive feedback of these modifications using the As-Rigid-As-Possible approach, but this creates a number of new features shown in Figure 1 (middle). In particular, a number of valley lines (blue) have been created. The curvature optimization is able to cure these defects: we prescribe the transformed feature lines as positional constraints and prescribe the curvature of the original mesh to maintain the original properties. The result is an intuitive modification of the mesh, as shown in Figure 1 (right).

3.4. Adding Features

To add a ridge or valley to the surface, we draw a line on the mesh and modulate the curvature values in its vicinity. For a ridge, we multiply κ_{max} with a factor $s > 1$ such that $|\kappa_{max}| > |\kappa_{min}|$. For a valley, we multiply κ_{min} with a factor $s > 1$ such that $|\kappa_{max}| < |\kappa_{min}|$. To achieve a smooth transition, we define an influence area, i.e., a band of triangles around the drawn line, and interpolate between the new curvature values on the line and the original curvature values at the borders of the influence area. Figure 4 shows an example of this operation.

4. Discussion and Conclusions

We developed an intuitive, feature-centered modeling approach for irregular meshes. The extraction of ridges and valleys combined with curvature optimization provide us with a number of editing operations that maintain features and prevent unwanted features from appearing. This novel combination leads to an intuitive modeling tool for high-quality surface manipulation.

We implemented our method in C++. The running time for the feature extraction is in the order of a few seconds. The curvature optimization for the examples shown in this paper takes between 2 minutes (Figure 4) and 5 minutes (Figure 1). We have run these tests in a single thread on a MacBook Pro with a 2.53 GHz Intel Core 2 Duo.

Our framework does not yet check for conflicting constraints. An example for conflicting constraints would be a surface region where the principal curvatures are prescribed to be zero, but the enclosing feature lines are constrained to some non-planar shape. In such situations, the curvature optimization will get stuck in a local minimum. A way to solve such issues is to relax either the curvature or the positional constraints of the feature curves in conflicting areas.

We also require relatively high mesh quality to ensure accurate curvature computation and ridge/valley extraction. For non man-made models, feature curves are often less sharp. Thus, for future work, we are looking into modifications that allow us to process models with less pronounced features.

References

[EP09] EIGENSATZ M., PAULY M.: Positional, metric, and curvature control for constraint-based surface deformation. *Computer Graphics Forum 28*, 2 (2009), 551–558. 2, 3

[ESP08] EIGENSATZ M., SUMNER R., PAULY M.: Curvature-domain shape processing. *Computer Graphics Forum 27*, 2 (2008), 241–250. 2, 3

[GSMCO09] GAL R., SORKINE O., MITRA N., COHEN-OR D.: iWIRES: An analyze-and-edit approach to shape manipulation. *ACM Trans. Graph. 28*, 3 (2009). 1

[HP04] HILDEBRANDT K., POLTHIER K.: Anisotropic filtering of non-linear surface features. *Computer Graphics Forum 23*, 3 (2004), 391–400. 1

[HPW05] HILDEBRANDT K., POLTHIER K., WARDETZKY M.: Smooth feature lines on surface meshes. In *Symposium on Geometry Processing* (2005), pp. 85 – 90. 2

[LLCO08] LIPMAN Y., LEVIN D., COHEN-OR D.: Green coordinates. *ACM Trans. Graph. 27*, 3 (2008). 1

[NISA07] NEALEN A., IGARASHI T., SORKINE O., ALEXA M.: FiberMesh: Designing freeform surfaces with 3D curves. *ACM Trans. Graph. 26*, 3 (2007), 41. 1

[OBS04] OHTAKE Y., BELYAEV A., SEIDEL H.-P.: Ridge-valley lines on meshes via implicit surface fitting. *ACM Trans. Graph. 23* (2004), 609 – 612. 1

[SA07] SORKINE O., ALEXA M.: As-rigid-as-possible surface modeling. In *Symposium on Geometry Processing* (2007), pp. 109–116. 3

[SB09] SORKINE O., BOTSCH M.: Tutorial: Interactive shape modeling and deformation. In *Eurographics* (2009). 1, 3

[SF98] SINGH K., FIUME E.: Wires: a geometric deformation technique. In *Proc. of ACM SIGGRAPH* (1998), pp. 405–414. 1

[SLCO*04] SORKINE O., LIPMAN Y., COHEN-OR D., ALEXA M., RÖSSL C., SEIDEL H.-P.: Laplacian surface editing. In *Symposium on Geometry Processing* (2004), pp. 179–188. 3

[ZNA07] ZIMMERMANN J., NEALEN A., ALEXA M.: SilSketch: automated sketch-based editing of surface meshes. In *Proc. of Sketch-based Interfaces and Modeling* (2007), pp. 23–30. 1