

Removing local irregularities of triangular meshes with highlight line models

YONG Jun-Hai^{1,4} †, DENG Bai-Lin^{1,2,4}, CHENG Fuhua³, WANG Bin^{1,4},
WU Kun^{1,2,4} & GU Hejin⁵

¹School of Software, Tsinghua University, Beijing 100084, P. R. China

²Department of Computer Science and Technology, Tsinghua University, Beijing 100084, P. R. China

³Department of Computer Science, University of Kentucky, Lexington, KY 40506-0046, USA

⁴Key Laboratory for Information System Security, Ministry of Education of China, Beijing 100084, P. R. China

⁵Jiangxi Academy of Sciences, Nanchang 330029, P. R. China

Abstract

The highlight line model is a powerful tool in assessing the quality of a surface. Its presence increases the flexibility of an interactive design environment. In this paper, a method to generate a highlight line model on an arbitrary triangular mesh is presented. Based on the highlight line model, a fairing technique to remove local irregularities of a triangular mesh is then presented. The fairing is done by solving a minimization problem and performing an iterative procedure. The new technique improves not only the shape quality of the mesh surface, but the highlight line model as well. It provides an intuitive and yet suitable method for locally repairing a triangular mesh.

Keywords: highlight lines, mesh fairing, shape modification, model repair

摘要

高光线是检测自由曲面质量的有效工具，它为交互设计提高自由曲面质量提供直观的手段及其便利。本文提出了在任意三角形网格曲面上生成高光线的一种方法。基于该高光线模型，本文给出了一种对三角形网格局部不规则区域进行调整光顺的方法。这种三角形网格局部的光顺是通过对一个最小化问题的求解得到的，而且采用了一个迭代的过程。所提出的方法首先通过改善高光线模型的质量，然后对三角形网格的顶点进行调整，从而最终提高三角形网格的形状质量。该方法的求解过程非常直观，而且适合于局部修复三角形网格的形状质量。

关键词: 高光线, 网格光顺, 形状调整, 模型修复

1 Introduction

The highlight line model [1] is a powerful tool in assessing the quality of free-form surfaces, because the discontinuity on a surface is magnified by an order of one in the highlight line model [1]. And it has become increasingly popular in engineering design, especially in the design of automotive-body surfaces. Actually it has already been included as a design tool in several commercial geometric modeling systems, such as EDS' Unigraphics and Tsinghua University's TiGems. Recently there is a strong demand for efficient, dynamic highlight lines generation from the graphics side and video entertainment industry as well [2], because highlight lines can aid depth perception and, consequently, realism of a scene.

† Corresponding author (email: yongjh@tsinghua.edu.cn)

While non-uniform rational B-spline (NURBS) surfaces continue to be a major representation scheme in 3D modeling, triangular meshes have gained much popularity in graphics and geometric modeling recently. Triangular meshes have advantages over traditional parametric surfaces in several aspects. Unlike traditional parametric surfaces, the definition of a triangular mesh does not require a rectangular parametric domain. There is no restriction on the shape and topology of a triangular mesh. A triangular mesh is all that is needed to represent any solid object or surface. Besides, modern graphics hardware is optimized to render triangles, making triangular meshes important in the graphics processing pipeline. Triangular meshes have already been a primary surface/solid representation scheme in many areas, such as reverse engineering, rapid prototyping, conceptual design, and simulation, with three-dimensional scanners as a standard source for geometric data acquisition. There are other ways to produce a triangular mesh as well. Triangulation of free-form surfaces is usually necessary for rendering or manufacturing purpose. Subdivision schemes, which provide a new way to generate surfaces, may lead to triangular meshes as well, and have been used in some games and three-dimensional cartoons. Our goal here is to make the highlight line model available for triangular meshes so that it is possible to visually assess the quality of a triangular mesh, and to develop techniques to optimize the mesh faces where quality of the mesh is not satisfactory.

Due to the increasing importance of triangular meshes, various *mesh smoothing* techniques have been developed during the past decade to improve mesh surface quality. These techniques perform their tasks by changing the positions of mesh vertices without affecting their connectivity. Mesh smoothing has two different goals. The first one is to eliminate noises in mesh data. For example, meshes acquired by range scanners usually have high frequency noises in the vertex positions. And mesh smoothing methods are applied to smooth out these noises while preserving the overall shape of the mesh model. Such mesh smoothing methods are referred to as *mesh denoising*. Among them, *filtering techniques* iteratively apply local filters to mesh vertices to obtain their new positions. Taubin [3] defines the Laplacian operator on mesh vertices, and alternately applies two Laplacian filters with different scale factors to attenuate mesh shrinkage. Variants of Laplacian smoothing (e.g., [4][5]) have been proposed for improved performance such as automatic anti-shrinking effects. Other filtering techniques such as Wiener filters [6][7] and bilateral filters [8][9] have also been developed. Another class of denoising techniques, the *geometric flow methods*, evolve a mesh by determining the velocity of each mesh vertex as a function of the current geometry. Examples include the diffusion flow and mean curvature flow proposed by Desbrun et al. [10], and other works with different choices the velocity function [11][12][13]. To preserve the geometric features such as edges and corners while denoising the mesh, *anisotropic diffusion methods* are developed [14][15][16]. The basic idea is to smooth the mesh surface in a certain direction and retain or enhance sharp features in another direction. The above techniques modify vertex positions directly. *Normal filtering techniques*, instead, smooth mesh normals, and then evolve the mesh to fit the modified normals [17][18][19].

The other goal of mesh smoothing is to produce high quality surface that satisfies certain aesthetic requirements(i.e., a fair mesh). Such methods are usually called *mesh fairing*. Among these techniques, some tries to improve the shape quality of the whole mesh surface. For this purpose, *energy minimization* has been adapted from traditional computer aided geometric design(CAGD) techniques to perform mesh fairing. The idea is to minimize an energy functional that penalizes unaesthetic behaviors of the mesh shape [20][21][22]. Other mesh fairing techniques only modify part of the mesh

surface, which is usually inside a region specified by the user. We will refer to such fairing techniques as *local fairing*. Local fairing techniques usually determine new vertex positions inside modification region by high-order solving partial differential equations (PDEs), which characterize the properties of meshes with high quality shape and ensure geometric continuities of the mesh along the region boundaries. Schneider and Kobbelt [23] present an algorithm to create fair mesh surfaces with subdivision connectivity satisfying G^1 boundary conditions, by solving a fourth-order non-linear PDE. Later they extend the work onto irregular meshes[24]. Xu et al. [25] discuss the discretization and solution of several high-order non-linear PDEs for discrete surface modeling methods such as free-form mesh surface fitting with given boundary conditions. The surface diffusion flow method by Xu et al. in [25] essentially solves the same fourth-order PDE as the one solved by Schneider and Kobbelt in [23][24]. Therefore, the final surfaces obtained by these methods [23][24][25] have similar shape. For these PDE-based methods, the solution is affected by the boundary conditions of the PDE. Therefore, to acquire the desired shape of the modified surface with such PDE-based approaches, a user needs to be careful in specifying the modification region, in order to obtain appropriate boundary conditions. And the effect of the boundary specification on the final shape will not be known until the PDE is solved. Besides, as the examples in this paper indicate, on the new surface obtained from these local fairing techniques the highlight line model may not be of desired shape, which makes these techniques not suitable for applications where the shape of the highlight line model is critical.

In this paper, we propose to improve the quality of a triangular mesh with the help of a highlight line model. We start with defining a highlight line model for triangular meshes and proposing an efficient method for the construction of such a model. With the highlight line model, it is easy to identify shape irregularities of a triangular mesh. We then propose a method to remove local irregularities identified with the highlight line model, and produce a new mesh with better surface quality and highlight line model. Our method first constructs a set of smooth curves as the target shape of the highlight lines inside the modification region. Then we iteratively moves the mesh vertices by minimizing a target function which measures the shape quality of the new mesh surface as well as the difference between the new highlight lines and the target highlight lines. Note that for triangular meshes, irregularity can also refer to irregular distributions of vertices over the mesh surface. In this paper we do not consider such irregularities, and we only remove irregularities of mesh surface shape.

Our method assumes the mesh to be noise-free and seeks to improve the mesh surface quality inside user-specified regions. It falls into the category of local fairing techniques. In our method, the shape of the highlight line model magnifies the discontinuities on the mesh surface, which helps the user to locate the region with shape irregularities for subsequent optimization. The constructed target highlight lines reveal the shape of the new surface and new highlight lines without actually performing the optimization, and enables the user to decide whether the specification of modification region is appropriate. The target highlight lines are constructed using Optimized Geometric Hermite(OGH) interpolation [26], which is able to generate smooth interpolating curves without undesired loops, cusps, or folds. By optimizing a fairness function that considers the quality of the new surface as well as the new highlight line model, we obtain a modified surface with desired shape of highlight line model. Our approach leads to a more intuitive and flexible process of local fairing, especially when a high quality highlight line model is required.

Apart from the highlight line model, the reflection line model [27] can also magnify the discontinuities on a surface and has been used in mesh quality assessment. How-

ever, the reflection line model is dependent on both the viewpoint and the light sources. The highlight line model is a simplification of the reflection line model, which decouples the viewing operation from the manipulation of highlight lines [1]. This results in more effective interaction during the inspection of the surface. Therefore, we choose the highlight line model as the quality assessment tool. The optimization technique proposed in this paper can also be extended to work with the reflection line model.

The rest of the paper is organized as follows. Section 2 introduces the highlight line model for NURBS surfaces and its generalization to triangular meshes, and proposes a method to compute it. A method for improving the quality of a triangular mesh using the highlight line model is presented in Section 3. Implementation details and examples are provided in Section 4. Concluding remarks and possible future research directions are discussed in Section 5.

2 Highlight line model for triangular meshes

2.1 Highlight line model on NURBS surfaces and triangular meshes

Given a NURBS surface $\mathbf{P}(u, v)$, a highlight line is the imprint of a linear light source positioned above the surface. Let $\mathbf{L}(t)$ be the parametric representation of a linear light source

$$\mathbf{L}(t) = \mathbf{A} + t\mathbf{H}, \quad t \in \mathbb{R},$$

where \mathbf{A} is a point on $\mathbf{L}(t)$, and \mathbf{H} is a vector defining the direction of $\mathbf{L}(t)$. The imprint of $\mathbf{L}(t)$ on $\mathbf{P}(u, v)$ is a set of points of $\mathbf{P}(u, v)$, for which the perpendicular distance between the surface normal and $\mathbf{L}(t)$ is zero. More precisely, for any point \mathbf{B} on $\mathbf{P}(u, v)$, denote by $\mathbf{N}_{\mathbf{B}}$ the surface normal at \mathbf{B} . Then the line through \mathbf{B} along direction $\mathbf{N}_{\mathbf{B}}$ is given by

$$\mathbf{E}(s) = \mathbf{B} + s\mathbf{N}_{\mathbf{B}}, \quad s \in \mathbb{R}.$$

\mathbf{B} is in the imprint of $\mathbf{L}(t)$ if $\mathbf{E}(s)$ intersects $\mathbf{L}(t)$. This imprint is called a *highlight line* corresponding to $\mathbf{L}(t)$ (see Figure 1(a)). If a set of coplanar parallel linear light sources is used, the family of highlight lines corresponding to these light sources is called a *highlight line model* (see Figure 1(b)). A highlight line model is sensitive to the changes of surface normal directions, and thus can be used to detect surface normal/curvature irregularities [1].

We define a highlight line model for triangular meshes in a similar way. Given a linear light source $\mathbf{L}(t)$, the highlight line corresponding to $\mathbf{L}(t)$ is the set of points on the mesh surface where the perpendicular distance between the surface normal and $\mathbf{L}(t)$ is zero. A highlight line model on the mesh is a family of highlight lines corresponding to a set of coplanar parallel linear light sources, where the distance between adjacent light sources is constant. We compute a highlight line model for a triangular mesh with the following steps. First for each mesh vertex, the intersection point between its normal direction and the light source plane is located. Then on each mesh edge, we use linear interpolation to find the points whose normal direction intersects the light sources. We call such points *highlight nodes*. They are the intersection points of the highlight lines with the mesh edges. Finally, on each triangle, highlight nodes corresponding to the same light source are connected with line segments. Details of these steps are presented below.

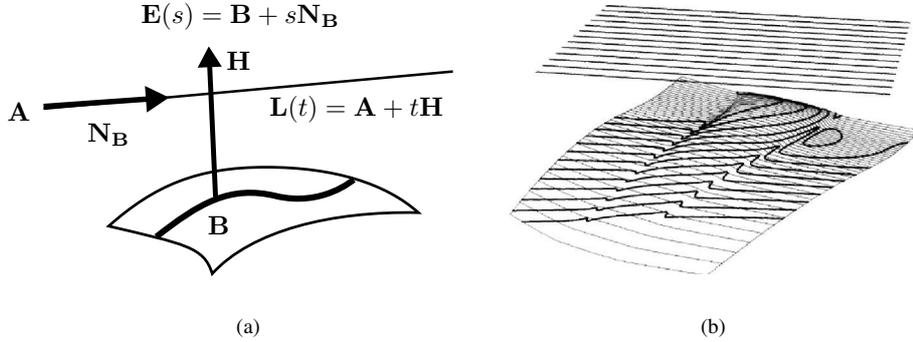


Figure 1: Illustration of a highlight line (a) and a highlight line model (b) on a NURBS surface ((b) is reproduced from [28]).

2.2 Intersection point calculation

Let \mathbf{S} be the light source plane, \mathbf{Z} the unit normal vector of \mathbf{S} , and \mathbf{H} the unit direction vector of the light sources. For a point \mathbf{P} on the mesh surface with the unit normal vector \mathbf{N}_P , the line through \mathbf{P} along direction \mathbf{N}_P is $\mathbf{E}_P(s) = \mathbf{P} + s\mathbf{N}_P, s \in \mathbb{R}$. Our task here is to locate the point where $\mathbf{E}_P(s)$ intersects \mathbf{S} . Instead of the exact position of the intersection point, we only need its *signed distance value* defined as follows. Choose one of the light sources \mathbf{L}_0 as the *base light source*, and let \mathbf{A}_0 be a point on \mathbf{L}_0 . For a point \mathbf{Y} on plane \mathbf{S} , the signed distance value of \mathbf{Y} to the base light source \mathbf{L}_0 is defined as

$$D_Y = (\mathbf{Y} - \mathbf{A}_0) \cdot (\mathbf{Z} \times \mathbf{H}).$$

For two points \mathbf{Y}_1 and \mathbf{Y}_2 on different sides of \mathbf{L}_0 , D_{Y_1} and D_{Y_2} are of different signs. Denote by d_P the signed distance value of the intersection point between line $\mathbf{E}_P(s)$ and plane \mathbf{S} . Then as shown in [2],

$$d_P = \frac{[(\mathbf{P} - \mathbf{A}_0) \times \mathbf{H}] \cdot \mathbf{N}_P}{\mathbf{Z} \cdot \mathbf{N}_P}. \quad (1)$$

We call d_P the *highlight distance value* of point \mathbf{P} . It has the following property. Let s be the distance between adjacent light sources in \mathbf{S} . If

$$d_P = s \times m \quad (2)$$

where m is an integer, then the intersection point is on the m th light source counting from \mathbf{L}_0 along direction $\mathbf{Z} \times \mathbf{H}$.

2.3 Highlight node calculation

We next compute and store the highlight nodes. From the above property of highlight distance values, the highlight nodes are those points on mesh edges whose highlight distance values satisfy Equation (2). We first calculate the highlight distance value for each mesh vertex, and then use linear interpolation to obtain the highlight distance values for interior points of a mesh edge. For each mesh vertex \mathbf{V} , we calculate its unit

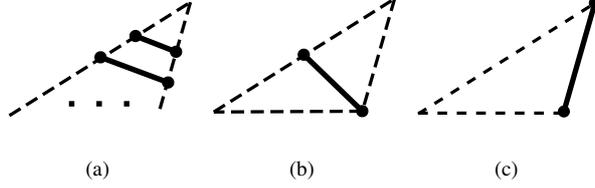


Figure 2: Possible cases of highlight node connection

normal vector as the normalized sum of the unit normal vector of all its adjacent triangles, weighted by their areas. This unit normal vector is used to obtain the highlight distance value $d_{\mathbf{V}}$ of \mathbf{V} from Equation (1). For an interior point $\hat{\mathbf{P}}$ of a mesh edge \mathbf{E}_i , the highlight distance value of $\hat{\mathbf{P}}$ is obtained by performing linear interpolation on highlight distance values $d_{\mathbf{V}_{i1}}$ and $d_{\mathbf{V}_{i2}}$ of the two vertices \mathbf{V}_{i1} and \mathbf{V}_{i2} of \mathbf{E}_i , i.e.,

$$d_{\hat{\mathbf{P}}} = \frac{\|\hat{\mathbf{P}} - \mathbf{V}_{i2}\|d_{\mathbf{V}_{i1}} + \|\mathbf{V}_{i1} - \hat{\mathbf{P}}\|d_{\mathbf{V}_{i2}}}{\|\mathbf{V}_{i1} - \mathbf{V}_{i2}\|}. \quad (3)$$

Now we have highlight distance values for all points on mesh edges, we can find out and store the points satisfying Equation (2) as highlight nodes. Let \mathbf{Q} be a highlight node where $d_{\mathbf{Q}} = s \times m_{\mathbf{Q}}$ for some integer $m_{\mathbf{Q}}$. We call $m_{\mathbf{Q}}$ the *index* of \mathbf{Q} . The index of a highlight node indicates the light source it corresponds to. For a mesh edge \mathbf{E}_i , if its two vertices \mathbf{V}_{i1} and \mathbf{V}_{i2} are highlight nodes with the same index m , then Equation (3) indicates that all points on \mathbf{E}_i are highlight nodes with the index m . We call such edge a *highlight edge*, and only store its two vertices as highlight nodes. Otherwise, there are a limited number of highlight nodes on \mathbf{E}_i . More precisely, for an edge \mathbf{E}_i that is not a highlight edge, there are highlight nodes on \mathbf{E}_i only if $\lceil \min(d_{\mathbf{V}_{i1}}, d_{\mathbf{V}_{i2}})/s \rceil \leq \lfloor \max(d_{\mathbf{V}_{i1}}, d_{\mathbf{V}_{i2}})/s \rfloor$, where $\lceil \cdot \rceil$ and $\lfloor \cdot \rfloor$ are the ceiling and floor functions, respectively. In this case, the index of any highlight node on \mathbf{E}_i is between $\lceil \min(d_{\mathbf{V}_{i1}}, d_{\mathbf{V}_{i2}})/s \rceil$ and $\lfloor \max(d_{\mathbf{V}_{i1}}, d_{\mathbf{V}_{i2}})/s \rfloor$. According to Equation (3), for each integer m in this range, there is exactly one highlight node with the index m on the edge \mathbf{E}_i , and its position can be computed as

$$\mathbf{Q} = \frac{(m - d_{\mathbf{V}_{i2}})\mathbf{V}_{i1} + (d_{\mathbf{V}_{i1}} - m)\mathbf{V}_{i2}}{d_{\mathbf{V}_{i1}} - d_{\mathbf{V}_{i2}}}. \quad (4)$$

For an edge that is not a highlight edge, we compute and store each of the highlight nodes on it with Equation (4).

2.4 Highlight node connection

After locating and storing the highlight nodes, we connect them to form segments of the highlight lines. Inside each triangle, we connect the highlight nodes with same index (see Figures 2 for examples). Note that in this way, any highlight edge will become one segment (see Figure 2(c)), and the highlight segments do not intersect inside a triangle.

The steps to compute a highlight line model for a triangular mesh is given in Algorithm 1. Figure 3 illustrates a highlight line and a highlight line model generated with this algorithm.

Algorithm 1: Calculate the highlight line model of a triangular mesh

Input: A triangular mesh M , and an array of coplanar parallel linear light source**Output:** The highlight line model of M corresponding to the light sources

```
1 Assign the set  $S_N$  of highlight nodes an empty set;
2 for each vertex  $V_i$  of  $M$  do
3   Calculate the highlight distance value with Equation (1);
4 end
5 for each edge  $E_i$  of  $M$  do
6   if the two vertices of  $E_i$  are highlight nodes with the same index then
7     Add both vertices of  $E_i$  to  $S_N$ ;
8   else
9     Calculate the highlight nodes on  $E_i$  with Equation (4);
10    Add each highlight node on  $E_i$  to  $S_N$ ;
11  end
12 end
13 for each triangle  $T_i$  of  $M$  do
14   Connect any nodes in  $S_N$  that lie on the edges of  $T_i$  and have the same
      index;
15 end
```

3 Mesh fairing using highlight lines

With the highlight line model introduced in the previous section, we can identify regions of a triangular mesh with irregular normal/curvature by assessing the quality of the highlight lines. This is done by translating and rotating the mesh or the array of linear light sources, in an interactive environment, to sweep the highlight line model over the given mesh. We propose in this section a method to remove shape irregularities from a triangular mesh. The first step is to identify an *irregular region*. The second step is to move vertices in this region so that desired shape of the highlight lines can be constructed. The displacements of the mesh vertices are calculated by minimizing a target function that measures the fairness of the new mesh surface as well as the shape quality of the new highlight lines. Moving the vertices according to the computed displacements, we obtain a new mesh with improved surface shape and highlight line model. The above steps are iteratively repeated until the displacements converge to zero. If there are several irregular regions, we perform the above procedure to remove them, one at a time. The details of this method are presented below.

3.1 Irregular region identification

We identify an irregular region by assessing the quality of the highlight line model and interactively specifying the region that requires modification. See Figure 4 for an example. With this region we can determine the mesh vertices to be moved. Denote the region by R . Our goal is to improve the surface quality inside R , without affecting the surface or the highlight line model outside R . Denote by S_{vertex} and S_{node} the sets of mesh vertices and highlight nodes outside R , respectively. To keep the surface and highlight lines outside R unchanged, the movement of the vertices should not change any of the following properties:

- normal vectors and positions of vertices in S_{vertex} ;

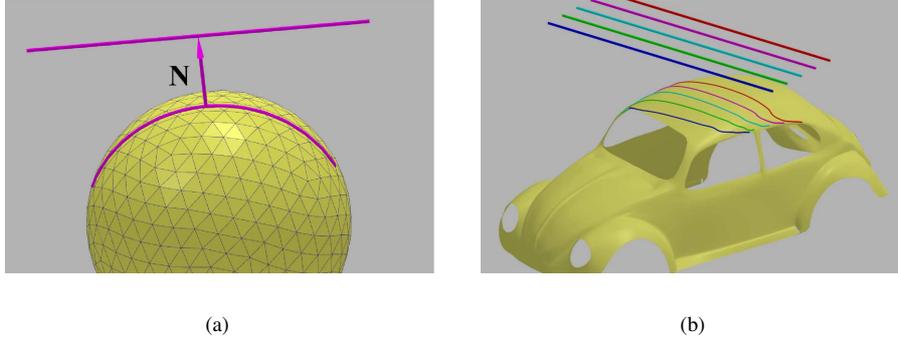


Figure 3: Illustration of a highlight line (a) and a highlight line model (b) on triangular meshes.

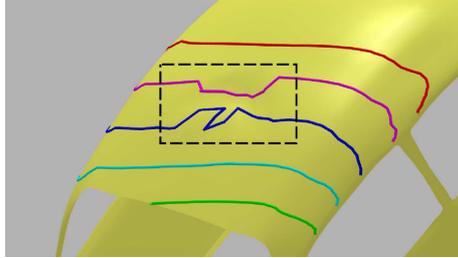


Figure 4: An irregular region of a mesh specified by the user.

- positions of highlight nodes in S_{node} .

Here we introduce the concept of *support vertices*. Given a mesh vertex or highlight node \mathbf{X} , the support vertices of \mathbf{X} are the mesh vertices that would affect the above properties of \mathbf{X} when any of these mesh vertices is moved. If \mathbf{X} is a mesh vertex, the support vertices include itself and the vertices adjacent to it, due to the way we compute vertex normals. If \mathbf{X} is a highlight node on an edge \mathbf{E}_i but not a mesh vertex, its support vertices include the vertices \mathbf{V}_{i1} and \mathbf{V}_{i2} of \mathbf{E}_i , and the support vertices of \mathbf{V}_{i1} and \mathbf{V}_{i2} , according to Equation (4). We can only move vertices of \mathbf{R} that do not belong to the support vertices of S_{vertex} and S_{node} . Those vertices will be called *movable vertices* of \mathbf{R} .

3.2 Desired highlight lines

To construct highlight lines with desired shape for the specified region \mathbf{R} , we replace the undesired portion of a highlight line with an interpolating curve of desired shape. We assume the mesh surface outside \mathbf{R} is of good quality, and will take interpolation conditions from this part of the surface.

For each highlight line crossing the specified region \mathbf{R} , find the highlight nodes on the highlight line that are outside but closest to \mathbf{R} . There are two of them, one on each side. Denote these two nodes by \mathbf{Q}_0 and \mathbf{Q}_1 (see Figure 5), and the tangent

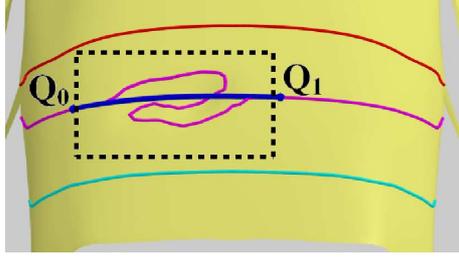


Figure 5: An OGH curve constructed for the specified region.

vectors of the highlight line at them by \mathbf{T}_0 and \mathbf{T}_1 , respectively. The interpolating curve should connect \mathbf{Q}_0 and \mathbf{Q}_1 , and have \mathbf{T}_0 and \mathbf{T}_1 as tangent vectors at these points. The traditional Hermite interpolation method is able to construct a Hermite curve satisfying these requirements. However, as pointed out in [26], a Hermite curve could have undesired loop, cusp, or fold. We will use an *optimized geometric Hermite (OGH) curve* [26] instead to design the interpolating curve segment. In contrast to a traditional Hermite curve, an OGH curve is not only mathematically smooth, i.e., with minimum strain energy, but also geometrically smooth, i.e., loop-, cusp- and fold-free [26]. The OGH curve segment satisfying the above interpolation conditions is of the following form

$$\mathbf{H}(t) = (2t+1)(t-1)^2\mathbf{Q}_0 + (-2t+3)t^2\mathbf{Q}_1 + (1-t)^2ta_0\mathbf{T}_0 + (t-1)t^2a_1\mathbf{T}_1, \quad t \in [0, 1], \quad (5)$$

where

$$\begin{cases} a_0 = \frac{6[(\mathbf{Q}_1 - \mathbf{Q}_0) \cdot \mathbf{T}_0] \cdot (\mathbf{T}_1^2) - 3[(\mathbf{Q}_1 - \mathbf{Q}_0) \cdot \mathbf{T}_1] \cdot (\mathbf{T}_0 \cdot \mathbf{T}_1)}{[4\mathbf{T}_0^2(\mathbf{T}_1^2) - (\mathbf{T}_0 \cdot \mathbf{T}_1)^2]}, \\ a_1 = \frac{3[(\mathbf{Q}_1 - \mathbf{Q}_0) \cdot \mathbf{T}_0] \cdot (\mathbf{T}_0 \cdot \mathbf{T}_1) - 6[(\mathbf{Q}_1 - \mathbf{Q}_0) \cdot \mathbf{T}_1] \cdot (\mathbf{T}_0^2)}{[(\mathbf{T}_0 \cdot \mathbf{T}_1)^2 - 4\mathbf{T}_0^2(\mathbf{T}_1^2)].} \end{cases}$$

Figure 5 shows an example of an OGH curve segment constructed in this way.

3.3 Vertex displacement calculation

With the desired highlight lines constructed, we now adjust some of the vertices of \mathbf{R} so that, afterward, the highlight line pattern of the region would be close to that of the constructed highlight lines and, consequently, the new shape of the region would have a better quality. Let $\{\mathbf{V}_i | i \in I_M\}$ be the set of movable vertices of \mathbf{R} . A vertex \mathbf{V}_i ($i \in I_M$) will be adjusted along the direction of its unit normal vector \mathbf{N}_i , which has been obtained during calculation of the highlight line model. Then we have its new position $\overline{\mathbf{V}}_i$ as $\overline{\mathbf{V}}_i = \mathbf{V}_i + x_i\mathbf{N}_i$, where x_i is the displacement of \mathbf{V}_i . Let \mathbf{X} be a displacement vector whose components are values $\{x_i | i \in I_M\}$. We will consider the new surface quality, after the adjustment of the vertices, as a function of the displacement vector \mathbf{X} , and obtain \mathbf{X} by optimization of the function value. We design the function as

$$F(\mathbf{X}) = \omega_1 f_{\text{fair}}(\mathbf{X}) + \omega_2 f_{\text{diff}}(\mathbf{X}),$$

where f_{fair} is a function that measures the fairness of the new mesh surface inside \mathbf{R} , and f_{diff} is a function that measures the difference between the highlight lines

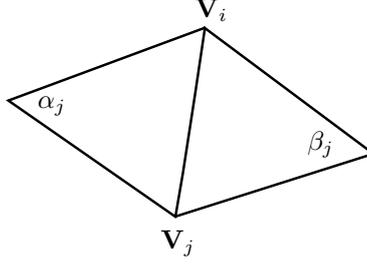


Figure 6: The angles α_j and β_j .

of the new mesh and the constructed desired highlight lines, with ω_1 and ω_2 being the weights. The details of construction and optimization of this target function are presented below.

3.4 Fairness function

We choose the fairness function to be the Willmore energy [29] of the new mesh surface. For a parametric surface with fixed boundary and fixed surface normals along the boundary, the Willmore energy is

$$E = \int H^2 dA,$$

where H denotes the mean curvature, and dA is the surface area element. For a connected region \mathbf{R} on a triangular mesh, let $\{\mathbf{V}_i | i \in I_R\}$ be the set of vertices inside \mathbf{R} . Then the Willmore energy for the mesh surface inside \mathbf{R} can be discretized as

$$E = \sum_{i \in I_R} H_i^2 A_i, \quad (6)$$

where H_i is the discrete mean curvature at vertex \mathbf{V}_i , and A_i is the mesh surface area associated with \mathbf{V}_i . Here A_i is computed as $\frac{1}{3}$ of the total areas of the triangles adjacent to \mathbf{V}_i . H_i^2 can be obtained as 2-norm of the discrete mean curvature normal operator $\mathbf{K}(\mathbf{V}_i) = H_i \mathbf{N}_i$ where \mathbf{N}_i is the unit normal vector at \mathbf{V}_i [30]. And $\mathbf{K}(\mathbf{V}_i)$ is calculated with the positions of \mathbf{V}_i and its adjacent vertices [30]:

$$\mathbf{K}(\mathbf{V}_i) = \frac{1}{A_i} \sum_{j \in N_1(i)} (\cot \alpha_j + \cot \beta_j)(\mathbf{V}_i - \mathbf{V}_j), \quad (7)$$

where A_i is the same as in Equation (6), $\{\mathbf{V}_j | j \in N_1(i)\}$ is the set of vertices adjacent to \mathbf{V}_i , and α_j and β_j are the two angles opposite to the edge $\mathbf{V}_i \mathbf{V}_j$, as illustrated in Figure (6). According to Equations (6) and (7), to derive the Willmore energy for new mesh surface in region \mathbf{R} , we need the new positions of the vertices in region \mathbf{R} and all their adjacent vertices. For such a vertex \mathbf{V}_j , its new position $\bar{\mathbf{V}}_j$ is

$$\bar{\mathbf{V}}_j = \begin{cases} \mathbf{V}_j + x_j \mathbf{N}_j, & \text{if } j \in I_M, \\ \mathbf{V}_j, & \text{otherwise.} \end{cases} \quad (8)$$

Now we have the expression of f_{fair} as a function of displacement $\{x_i | i \in I_M\}$.

3.5 Difference function

As described in Section 3.2, each highlight line crossing the irregular region is delimited by two highlight nodes, such as \mathbf{Q}_0 and \mathbf{Q}_1 in Figure 5. These two highlight nodes are the end points of its corresponding OGH interpolation curve. Their positions do not change after the adjustment of the movable vertices, but a new highlight line should be generated between them. Let $\mathbf{L}(s)$, $s \in [0, 1]$ and $\bar{\mathbf{L}}(s)$, $s \in [0, 1]$ be the normalized chord-length parameterization forms of the highlight line between these two delimiting nodes before and after the vertex adjustment, respectively. Let $\tilde{\mathbf{H}}(s)$, $s \in [0, 1]$ be the normalized arc-length parameterization form of the corresponding OGH curve. We define the difference function between the new highlight line $\bar{\mathbf{L}}(s)$ and its target shape $\tilde{\mathbf{H}}(s)$ as

$$f_{\mathbf{L}} = \int_0^1 \|\bar{\mathbf{L}}(s) - \tilde{\mathbf{H}}(s)\|^2 ds ,$$

and the difference function f_{diff} for the entire irregular region is the sum of the above function for all highlight lines crossing the region

$$f_{\text{diff}} = \sum_{\mathbf{L}} f_{\mathbf{L}} .$$

Function $f_{\mathbf{L}}$ can be discretized in the following way. Assume that during the generation of highlight lines with Algorithm 1, we have stored n highlight nodes on $\mathbf{L}(s)$ between the two delimiting nodes. Denote the two delimiting nodes by \mathbf{G}_0 and \mathbf{G}_{n+1} , and the nodes between them by \mathbf{G}_i ($i = 1, 2, \dots, n$), with $\mathbf{G}_0, \mathbf{G}_1, \dots, \mathbf{G}_n, \mathbf{G}_{n+1}$ being in the same order as they appear on $\mathbf{L}(s)$. First each node \mathbf{G}_i ($i = 1, 2, \dots, n$) is mapped to a point $\tilde{\mathbf{G}}_i$ on $\tilde{\mathbf{H}}(s)$. We call $\tilde{\mathbf{G}}_i$ the target position of \mathbf{G}_i . After adjustment of the vertices, the corresponding new position $\bar{\mathbf{G}}_i$ of \mathbf{G}_i can be computed, and $f_{\mathbf{L}}$ is given by

$$f_{\mathbf{L}} = \sum_{i=1}^n \|\bar{\mathbf{G}}_i - \tilde{\mathbf{G}}_i\|^2 l_i ,$$

where l_i is the length of the highlight line segments associated with \mathbf{G}_i . To determine the target position $\tilde{\mathbf{G}}_i$, we need the normalized chord-length parameter of \mathbf{G}_i on $\mathbf{L}(s)$, which is

$$c(\mathbf{G}_i) = \frac{\sum_{j=1}^i \|\mathbf{G}_j - \mathbf{G}_{j-1}\|}{\sum_{j=1}^{n+1} \|\mathbf{G}_j - \mathbf{G}_{j-1}\|} .$$

And $\tilde{\mathbf{G}}_i$ is determined as the point on $\tilde{\mathbf{H}}(s)$ with parameter $s = c(\mathbf{G}_i)$, i.e.,

$$\tilde{\mathbf{G}}_i = \tilde{\mathbf{H}}(c(\mathbf{G}_i)) .$$

The new position $\bar{\mathbf{G}}_i$ is computed as follows. Let \mathbf{E}_i be an edge that \mathbf{G}_i lies on, with $\mathbf{V}_{i1}, \mathbf{V}_{i2}$ being the vertices of \mathbf{E}_i . Since $\bar{\mathbf{G}}_i$ and \mathbf{G}_i correspond to the same light source, they should have the same index. According to Equation (4), $\bar{\mathbf{G}}_i$ can be obtained with the new positions and new highlight distance values of \mathbf{V}_{i1} and \mathbf{V}_{i2} , as well as the index m of \mathbf{G}_i ,

$$\bar{\mathbf{G}}_i = \frac{(m \cdot s - d_{\bar{\mathbf{V}}_{i2}})\bar{\mathbf{V}}_{i1} + (d_{\bar{\mathbf{V}}_{i1}} - m \cdot s)\bar{\mathbf{V}}_{i2}}{d_{\bar{\mathbf{V}}_{i1}} - d_{\bar{\mathbf{V}}_{i2}}} .$$

Here $\bar{\mathbf{V}}_{i1}$ and $\bar{\mathbf{V}}_{i2}$ are the new vertex positions obtained with Equation (8). $d_{\bar{\mathbf{V}}_{i1}}$ and $d_{\bar{\mathbf{V}}_{i2}}$ are the new distance values calculated with Equation (1). Finally, the associated

Algorithm 2: Remove local irregularities of a mesh using highlight lines

Input: A triangular mesh \mathbf{M} , a highlight line model of \mathbf{M} , an irregular region \mathbf{R} , a maximum number of iterations N_{\max} , and a threshold value ε

Output: A new mesh with irregularities in \mathbf{R}

- 1 Identify the set of movable vertices $\{\mathbf{V}_i | i \in I_M\}$ of \mathbf{R} ;
 - 2 Set the number of iterations $n = 0$;
 - 3 **repeat**
 - 4 Construct the target function F of displacements $\{x_i | i \in I_M\}$;
 - 5 Solve the minimization problem (9) to obtain the values of $\{x_i | i \in I_M\}$;
 - 6 **for each** $i \in I_M$ **do**
 - 7 Adjust the vertex \mathbf{V}_i according to x_i ;
 - 8 **end**
 - 9 Update the highlight line model of the mesh using Algorithm 1;
 - 10 Set $n = n + 1$;
 - 11 **until** $n > N_{\max}$ OR $\max_{i \in I_M} |x_i| / \bar{e} < \varepsilon$;
-

highlight segment length l_i of \mathbf{G}_i is calculated as half of the total length of the highlight line segments that it lies on,

$$l_i = \frac{1}{2} (\|\mathbf{G}_i - \mathbf{G}_{i-1}\| + \|\mathbf{G}_i - \mathbf{G}_{i+1}\|) .$$

Now we get f_{diff} as a function of the displacement \mathbf{X} .

3.6 Target function minimization

f_{fair} and f_{diff} defined in the previous sections are both highly non-linear in $\{x_i\}$. To speed up the minimization process, we use functions of a simpler form to approximate them. For f_{fair} , if we assume that A_i , α_j and β_j in Equation (7) are constants during adjustment of the vertices, then Equation (6) becomes a quadratic function q_{fair} of $\{x_i\}$. For f_{diff} , we perform Taylor series expansion of order 2 about point $\mathbf{X} = \mathbf{0}$ to obtain an approximation function q_{diff} , which is also quadratic in $\{x_i\}$. In addition, we put the following constraint on the components of the displacement vector

$$|x_i| \leq e_i/2, \text{ for all } i \in I_M,$$

where e_i is the minimum length of the edges adjacent to vertex \mathbf{V}_i . This constraint ensures that there will be no topological change on the mesh such as triangle flip-overs after vertex adjustment. The minimization problem now becomes

$$\begin{cases} \text{minimize } F = \omega_1 q_{\text{fair}} + \omega_2 q_{\text{diff}}, \\ \text{subject to } |x_i| \leq e_i/2, i \in I_M, \end{cases} \quad (9)$$

which is a bound constrained quadratic programming problem and can be solved using the active set method.

3.7 Iteration

We use an iterative procedure to gradually improve the quality of the irregular region. In each iteration step, the quadratic programming problem (9) is formed using current

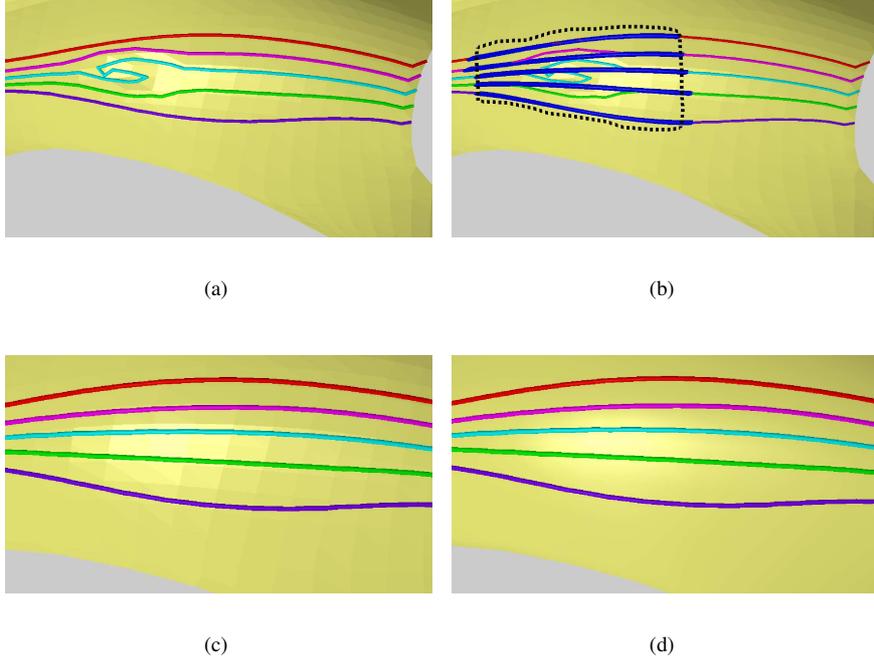


Figure 7: Example 1: **(a)** a mesh with the irregular highlight line model; **(b)** the selected modification region with the desired highlight lines (in blue); **(c)** the flat-shaded modification region after fairing, with its highlight line model; **(d)** the result with smooth shade.

geometric information of the mesh. Then we solve the minimization problem, and adjust the vertices according to the solution to obtain a new mesh. The process is terminated when the number of iterations exceeds a given bound, or the maximum absolute values of the displacement vectors converge to zero, i.e.,

$$\frac{\max_{i \in I_M} |x_i|}{\bar{e}} < \varepsilon, \quad (10)$$

where \bar{e} is the average edge length inside region \mathbf{R} , and ε is a positive threshold value specified by the user. The iterative procedure is summarized in Algorithm 2.

4 Implementation and examples

Here we show implementation results of the presented method on some mesh models. In these examples, we set $\omega_1 = \omega_2 = 1$ for the target function, and set $\varepsilon = 0.001$ for the termination condition specified in Formula (10). Figure 7 shows the fairing of the mesh model of a Volkswagen Beetle (see Figure 3(b) as well). In Figure 7(a), an irregularity of the front right fender is illustrated by the highlight line model. Figure 7(b) shows the region specified for fairing, as well as the desired highlight lines. Figures 7(c) and (d) provide a closer view of the resulting modification region after fairing in flat and smooth shade, respectively. The new mesh surface in the faired region is of high

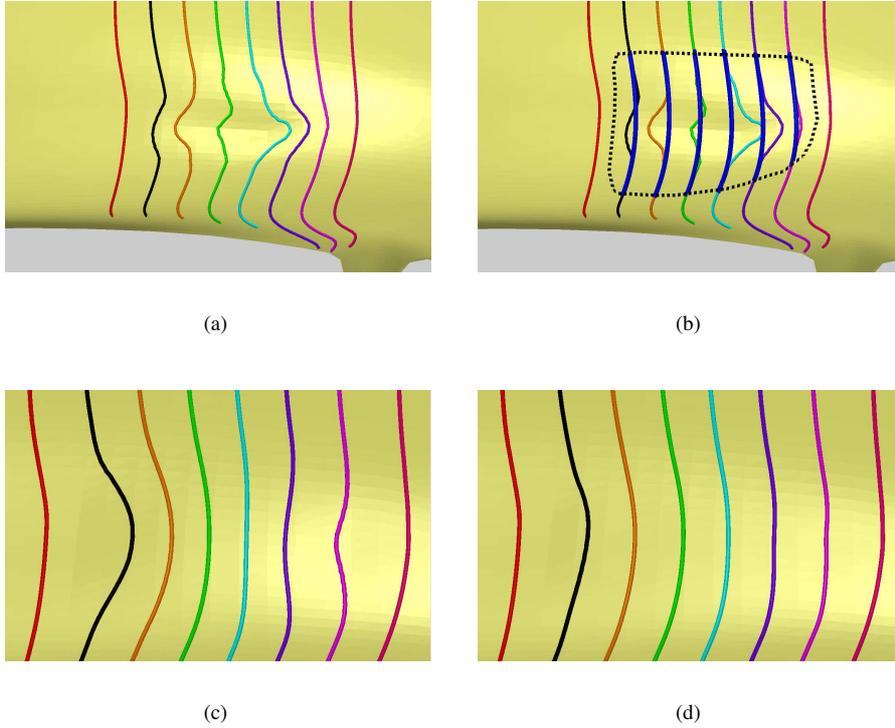


Figure 8: Example 2: **(a)** a mesh with irregular highlight line model; **(b)** the selected modification region with the desired highlight lines (in blue); **(c)** the resulting modification region from Xu et al.'s method, with its highlight line model; **(d)** the resulting modification region from our method, with its highlight line model.

quality; the new highlight lines are close to the desired ones. The smooth highlight lines indicate G^1 continuity of the resulting surface at boundaries of the modification region [1]. In Figure 8, we fair another irregular region on the roof of the Beetle model. To compare our method with other local fairing techniques, we need to reproduce other these techniques for the test case. Both the surface diffusion flow technique by Xu et al. [25] and the geometric fairing technique by Schneider and Kobbelt [23][24] are able to perform fairing in a user-specified region while satisfying G^1 boundary conditions. Essentially, both methods move each vertex inside the region by solving the fourth-order PDE $\Delta_B H = 0$, where Δ_B is the Laplace-Beltrami operator, and H is the mean curvature at a vertex. Since they lead to similar results, we only reproduce the surface diffusion flow technique by Xu et al. [25] for comparison. We first identify the region that requires adjustment, and perform fairing in that region with our method and Xu et al.'s method, respectively. Figures 8(a) and 8(b) show the irregular region and the selected region, respectively. Figures 8(c) and 8(d) are the resulting modification region from Xu et al.'s method and our method, respectively. On the mesh produced by Xu et al.'s method, the new highlight lines are curved toward the middle of the selected region. The highlight line shape has abrupt changes on the left and right boundary of the region. On the other hand, our method produces a mesh surface

with new highlight lines close to the desired ones. The new highlight lines naturally match the shape pattern of the highlight lines outside the selected region. This example shows that although existing local fairing techniques can generate high quality mesh surfaces, they do not guarantee the generation of high quality highlight line models. In our method, the fairness function helps to generate a fair surface, and the difference function makes the new highlight lines converge to the desired shape. Therefore, our method can improve the shape quality of both the mesh surface and the highlight line model.

5 Conclusions

A method to generate the highlight line model for a given triangular mesh is provided in this paper. With a highlight line model, the irregularity of the mesh surface is visualized by the irregularity of the highlight lines, which helps the user to identify the modification region for surface optimization. Subsequently, a method for removing local irregularities of a given triangular mesh is presented. The modification process is based on optimizing a fairness function that measures the shape quality of the mesh surface as well as the highlight line model. A set of target highlight lines are constructed as the target shape of the highlight line model, based on the geometry of the mesh surface along the boundary of the modification region. This target highlight line model enables the user to preview the shape of the new mesh surface as well as the new highlight line model, which helps the user to specify an appropriate modification region and leads to more intuitive control of the surface optimization process. The minimization of the fairness function guides the mesh surface towards a new shape with a highlight line model closed to the target highlight line model, which is not always available with previous local fairing techniques. The new method provides a whole set of tools from mesh surface quality assessment to mesh fairing, making itself a useful complement to geometric modeling techniques based on triangular meshes. It will bring greater flexibility to an interactive design environment for meshes.

Acknowledgements

We thank Leif Kobbelt for providing us with the Volkswagen Beetle model. The research was supported by the National Science Foundation of China (60533070, 60625202), and Chinese 973 Program(2004CB719400). The first author was supported by the project sponsored by a Foundation for the Author of National Excellent Doctoral Dissertation of PR China (200342), and a Program for New Century Excellent Talents in University(NCET-04-0088). The third author was supported by NSF under grants DMI-0422126 and DMS-0310645 and KSTC under grant COMM-Fund-712.

References

- 1 BEIER K P, CHEN Y. Highlight-line algorithm for realtime surface-quality assessment. *Computer-Aided Design*, 1994. 26(4): 268–277
- 2 YONG J H, CHENG F, CHEN Y, *et al.*. Dynamic highlight line generation for locally deforming NURBS surfaces. *Computer-Aided Design*, 2003. 35(10): 881–892

- 3 TAUBIN G. A signal processing approach to fair surface design. In: Proceedings of SIGGRAPH 95. Association for Computing Machinery, 1995. 351–358
- 4 LIU X, BAO H, HENG P A, *et al.*. Constrained fairing for meshes. Computer Graphics Forum, 2001. 20(2): 115–123
- 5 VOLLMER J, MENCL R, MÜLLER H. Improved Laplacian smoothing of noisy surface meshes. Computer Graphics Forum, 1999. 18(3): 131–138
- 6 ALEXA M. Wiener filtering of meshes. In: WYVILL G, ed., Proceedings of Shape Modeling International 2002. IEEE Computer Society, 2002. 51–60
- 7 PENG J, STRELA V, ZORIN D. A simple algorithm for surface denoising. In: Proceedings of IEEE Visualization 2001. IEEE Computer Society, 2001. 107–112
- 8 FLEISHMAN S, DRORI I, COHEN-OR D. Bilateral mesh denoising. ACM Transactions on Graphics, 2003. 22(3): 950–953
- 9 JONES T R, DURAND F, DESBRUN M. Non-iterative, feature-preserving mesh smoothing. ACM Transactions on Graphics, 2003. 22(3): 943–949
- 10 DESBRUN M, MEYER M, SCHRÖDER P, *et al.*. Implicit fairing of irregular meshes using diffusion and curvature flow. In: Proceedings of SIGGRAPH 99. Association for Computing Machinery, 1999. 317–324
- 11 BOBENKO A I, SCHRÖDER P. Discrete Willmore flow. In: DESBRUN M, POTTMANN H, eds., Geometry Processing 2005. A K Peters, 2005. 101–110
- 12 OHTAKE Y, BELYAEV A, BOGAEVSKI I A. Polyhedral surface smoothing with simultaneous mesh regularization. In: MARTIN R, WANG W, eds., Geometric Modeling and Processing 2000: Theory and Applications. IEEE Computer Society, 2000. 229–237
- 13 YOSHIZAWA S, BELYAEV A G. Fair triangle mesh generations via discrete elastica. In: SUZUKI H, MARTIN R, eds., Geometric Modeling and Processing: Theory and Applications. IEEE Computer Society, 2002. 119–123
- 14 BAJAJ C L, XU G. Anisotropic diffusion of surfaces and functions on surfaces. ACM Transactions on Graphics, 2003. 22(1): 4–32
- 15 CLARENZ U, DIEWALD U, RUMPF M. Anisotropic geometric diffusion in surface processing. In: Proceedings of IEEE Visualization 2000. IEEE Computer Society, 2000. 397–406
- 16 HILDERBRANDT K, POLTHIER K. Anisotropic filtering of non-linear surface features. Computer Graphics Forum, 2004. 23(3): 391–400
- 17 OHTAKE Y, BELYAEV A, SEIDEL H P. Mesh smoothing by adaptive and anisotropic Gaussian filter applied to mesh normals. In: GREINE G, NIEMANN H, ERTL T, *et al.*, eds., Vision, Modeling, and Visualization 2002. Amsterdam: IOS press, 2002. 203–210
- 18 TASDIZEN T, WHITAKER R, BURCHARD P, *et al.*. Geometric surface smoothing via anisotropic diffusion of normals. In: Proceedings of IEEE Visualization 2002. IEEE Computer Society, 2002. 125–132

- 19 YAGOU H, OHTAKE Y, BELYAEV A. Mesh smoothing via mean and median filtering applied to face normals. In: SUZUKI H, MARTIN R, eds., Geometric Modeling and Processing: Theory and Applications. IEEE Computer Society, 2002. 124–131
- 20 WELCH W, WITKIN A. Free-form shape design using triangulated surfaces. In: Proceedings of SIGGRAPH 94. Association for Computing Machinery, 1994. 157–166
- 21 KOBBELT L. Discrete fairing. In: GOODMAN T, MARTIN R, eds., The Mathematics of Surfaces VII. Winchester: Information Geometers, 1997. 101–131
- 22 KOBBELT L, CAMPAGNA S, VORSATZ J, *et al.*. Interactive multi-resolution modeling on arbitrary meshes. In: Proceedings of the SIGGRAPH 98. Association for Computing Machinery, 1998. 105–114
- 23 SCHNEIDER R, KOBBELT L. Generating fair meshes with G^1 boundary conditions. In: MARTIN R, WANG W, eds., Geometric Modeling and Processing 2000: Theory and Applications. IEEE Computer Society, 2000. 251–261
- 24 SCHNEIDER R, KOBBELT L. Geometric fairing of irregular meshes for free-form surface design. Computer Aided Geometric Design, 2001. 18(4): 359–379
- 25 XU G, PAN Q, BAJAJ C. Discrete surface modelling using partial differential equations. Computer Aided Geometric Design, 2006. 23(2): 125–145
- 26 YONG J H, CHENG F. Geometric Hermite curves with minimum strain energy. Computer Aided Geometric Design, 2004. 21(3): 281–301
- 27 KLASS R. Correction of local surface irregularities using reflection lines. Computer-Aided Design, 1980. 12(2): 73–76
- 28 CHEN Y, BEIER K P, PAPAGEORGIOU D. Direct highlight line modification on NURBS surfaces. Computer Aided Geometric Design, 1997. 14(6): 583–601
- 29 WILLMORE T J. Riemannian Geometry. New York: Oxford University Press, 1993
- 30 MEYER M, DESBRUN M, SCHRÖDER P, *et al.*. Discrete differential-geometry operators for triangulated 2-manifolds. In: HEGE H C, POLTHIER K, eds., Visualization and Mathematics III. New York: Springer-Verlag, 2003. 35–57