Fairing Triangular Meshes with Highlight Line Model

Jun-Hai Yong[a], Bai-Lin Deng[a,b], Fuhua (Frank) Cheng[c] and Kun Wu[a,b]

[a] School of Software, Tsinghua University, Beijing 100084, P. R. China

[b] Department of Computer Science and Technology, Tsinghua University, Beijing 100084, P. R. China

[c] Department of Computer Science, University of Kentucky, Lexington, KY 40506-0046, USA

Abstract

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.

1 Introduction

The highlight line model [3] is a powerful tool in assessing the quality of non-uniform rational B-spline (NURBS) surfaces, and 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 [28], because highlight lines can aid depth perception and, consequently, realism of a scene. Hence, we have a design tool that can walk hand in hand to add unprecedented realism to free-form surfaces for artistic rendering and 3D modeling.

While NURBS surfaces continue to be a major representation scheme in 3D modeling, triangular meshes have gained much popularity in graphics and geometric modeling recently [5]. 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. On the other hand, to represent a complicated solid object or surface using traditional parametric schemes, one usually needs several parametric surfaces and this causes problems such as continuity and smoothness between different surfaces. Besides, modern graphics hardware is optimized to render triangles, making triangular meshes even more important in the graphics processing pipeline.

Triangular meshes have already been used as a primary surface/solid representation scheme in many areas, such as reverse engineering, rapid prototyping, conceptual design, and simulation, with three-dimensional scanners being a standard source for acquiring geometric data. 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. With the comprehensive applications of triangular meshes, this research certainly will bring greater flexibility to both the manufacturing and entertainment industries.

1.1 Previous works

During the past decade, various techniques have been developed to improve mesh surface quality. These techniques, referred to as *mesh smoothing*, perform their tasks by changing the positions of mesh vertices without affecting their connectivity. Among them, *filtering techniques* iteratively apply local filters to mesh vertices to obtain their new positions, and evolve the mesh into desired shape. Taubin

[21] defined the Laplacian operator on mesh vertices which can be utilized in smoothing. To attenuate the shrinkage of the mesh induced by such Laplacian filter, Taubin[21] proposed to alternately apply Laplacian filters with two different scale factors. Variants of Laplacian smoothing (e.g., [22][14]) have been proposed for improved performance such as automatic anti-shrinking effects. Other filtering techniques such as *Wiener filters*([17][1]) and *bilateral filters*([11][9]) have also been adapted from image smoothing techniques to perform mesh smoothing.

Another class of mesh smoothing techniques, the *geometric flow methods*, models the movement of mesh vertices with equation $\partial_t \mathbf{V} = \mathbf{F}(\mathbf{V}, t)$, where $\partial_t \mathbf{V}$ represents the velocity of vertex $\mathbf{V}$ at time t, and $\mathbf{F}(\mathbf{V}, t)$ is a function of the geometric information of the mesh at that time. The equation is solved iteratively to obtain the desired mesh shape. Desbrun et al. [8] proposed the *diffusion flow*, with $\mathbf{F}(\mathbf{V}, t)$ being the discrete Laplacian of $\mathbf{V}$, and solved the equation with an implicit integration approach for improved stability and efficiency. Also proposed in [8] is the *mean curvature flow*, where $\mathbf{F}(\mathbf{V}, t)$ is the mean curvature normal at $\mathbf{V}$. Other geometric flow methods (e.g., [15][29][4])with different choices of $\mathbf{F}(\mathbf{V}, t)$ have been developed.

To preserve the geometric features such as edges and corners while smoothing the mesh, *anisotropic diffusion methods* are developed. The basic idea is to smooth the mesh surface in a certain direction and retain or enhance sharp features in another direction. Clarenz et al.[7] proposed a diffusion tensor defined from the principal directions and principal curvatures of the surface to perform feature-preserving mesh smoothing. Bajaj and Xu[2] developed an anisotropic scheme to smooth the mesh surface as well as higher order functions on the surface. Hildebrandt and Polthier[10] presented an anisotropic diffusion smoothing method that preserves non-linear geometric features.

The above techniques modify vertex positions directly. Instead, *normal filtering techniques* first smooth the mesh normals, and evolve the mesh to fit the modified normals. Yagou et al. [26] applied mean and median filtering schemes to face normals of meshes to obtain a smooth normal field. Ohtake et al. [16] developed an adaptive and anisotropic Gaussian filter acting on the mesh face normals for normal smoothing. In the work of Tasdizen et al.[20], normal smoothing and subsequent surface reconstruction is performed in a level-set setting.

Besides, *energy minimization* is also used to perform mesh smoothing. The idea is to minimize an energy functional that penalizes unaesthetic behavior of the mesh shape. Welch and Witkin [23] constructed a fair triangular mesh by minimizing the squared principal curvature energy. Kobbelt [12] proposed the technique of *discrete fairing*, which uses energy minimization in constructing a fair, refined mesh that interpolates the vertices of a given original mesh. This work was extended by Kobbelt et al. [13] to meshes of arbitrary connectivity for mesh smoothing in a multi-resolution modeling environment.

Last but not least, some other techniques smooth meshes by solving high order partial differential equations (PDE), which characterize the properties of the meshes with high quality shape. Schneider and Kobbelt[18] presented 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 extended the work onto irregular meshes with the same PDE [19]. Xu et al. [25] used various non-linear PDE to perform surface modeling tasks such as free-form mesh surface fitting with given boundary conditions.

Depending on their goals, mesh smoothing techniques can be classified into two categories. The first one, *mesh denoising*, tries to smooth out the noises at vertex positions acquired during the generation of meshes (e.g., noises in measured data produced by a range scanner). The other one aims to produce a mesh surface that satisfies certain aesthetic requirements (i.e., a fair mesh), and is usually called *mesh fairing*. In this paper, we improve the mesh surface quality and construct fair meshes with the help of highlight line models. Our method falls into the category of mesh fairing.

1.2 Contributions

In this paper, we generalize the highlight line model defined for NURBS surfaces to highlight line model on arbitrary triangular meshes, and propose an efficient method for the construction of the model. With the highlight line model, it is easier to identify shape irregularities of a triangular mesh. Furthermore, we propose a method to remove local irregularities identified with the highlight lines, and produce a new mesh with better surface quality and highlight line model. Our method iteratively moves the mesh vertices by minimizing a target function which measures the shape quality of the mesh surface as well as the highlight line model. 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.

The rest of this 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 highlight lines 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 on 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 surface $\mathbf{P}(u, v)$ is a set of points on $\mathbf{P}(u, v)$, 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 $\mathbf{N_B}$ the surface normal at $\mathbf{B}$. Then the line through $\mathbf{B}$ along direction $\mathbf{N_B}$ is given by

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

$\mathbf{B}$ 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 [3].

We define 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.

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 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 $\mathbf{S}$. Denote this point by $\mathbf{X_P}$. Instead of calculating the exact position of $\mathbf{X_P}$, we only need to calculate its *signed distance value* defined as follows. Let $\mathbf{L}_0$ be one of the light sources in $\mathbf{S}$, and $\mathbf{A}_0$ a point on $\mathbf{L}_0$. We call $\mathbf{L}_0$ the base light source. 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

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

For two points $\mathbf{Y}_1$ and $\mathbf{Y}_2$ on different sides of $\mathbf{L}_0$, $D_{\mathbf{Y}_1}$ and $D_{\mathbf{Y}_2}$ are of different signs. To indicate the dependence of $D_{\mathbf{X_P}}$ on point $\mathbf{P}$, we denote $D_{\mathbf{X_P}}$ by $d_{\mathbf{P}}$, and call $d_{\mathbf{P}}$ the *highlight distance value* of point $\mathbf{P}$ to $\mathbf{L}_0$. As shown in

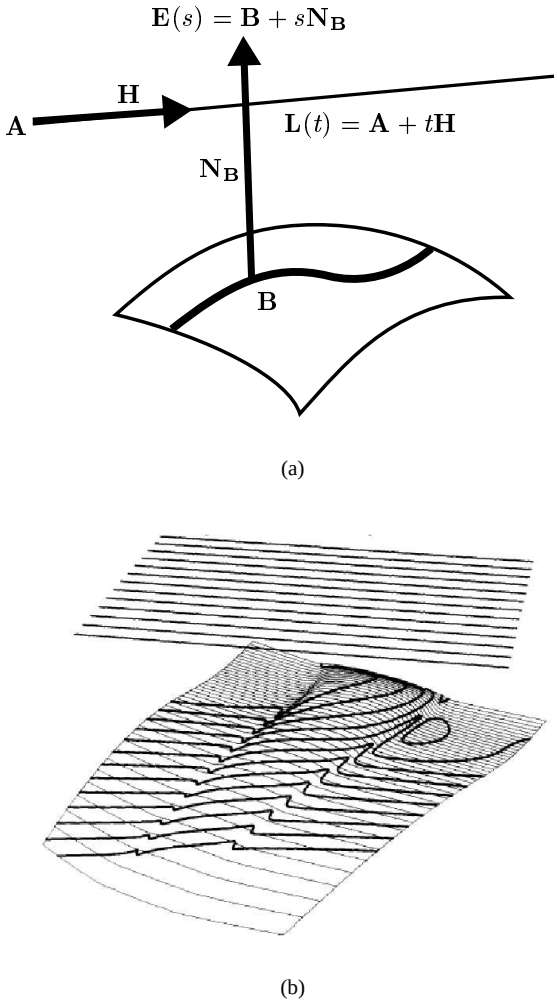


Figure 1. Illustration of a highlight line (a) and a highlight line model (b) on a NURBS surface ((b) is reproduced from [6]).

[28], $d_{\mathbf{P}}$ can be computed as

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

The highlight distance value has the following property. Let s be the distance between adjacent light sources in $\mathbf{S}$. If

$$d_{\mathbf{P}} = s \times m \quad (2)$$

where m is an integer, then intersection point $\mathbf{X}_{\mathbf{P}}$ is on the mth light source counting from $\mathbf{L}_0$ in the direction of $\mathbf{Z} \times \mathbf{H}$.

2.3 Highlight node calculation

We next compute all the highlight nodes on each mesh edge. From the definition of highlight lines on a triangular mesh, and the above property of highlight distance values, we know that highlight nodes are the points on mesh edges with their highlight distance values satisfying Equation (2). According to Equation (1), the determination of highlight distance value of a point requires the unit normal vector at that point. However, for a triangular mesh, we only know the normal vectors at mesh vertices. Therefore, 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 a mesh vertex $\mathbf{V}_i$, we calculate its unit normal vector $\mathbf{N}_{\mathbf{V}_i}$ as the normalized sum of the unit normal vector of all the adjacent triangles $\mathbf{T}_j$ of $\mathbf{V}_i$, weighted by their areas, i.e.,

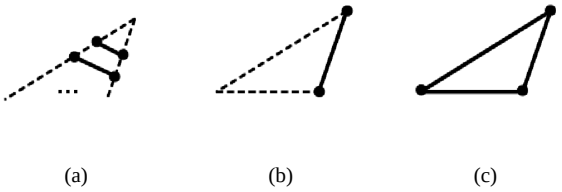
$$\mathbf{N}_{\mathbf{V}_i} = \frac{\sum_{j \in t(i)} A_{\mathbf{T}_j} \mathbf{N}_{\mathbf{T}_j}}{\| \sum_{j \in t(i)} A_{\mathbf{T}_j} \mathbf{N}_{\mathbf{T}_j} \|}, \quad (3)$$

where $t(i)$ is the index set of the adjacent triangles of $\mathbf{V}_i$, $\mathbf{N}_{\mathbf{T}_j}$ is the unit normal vector of triangle $\mathbf{T}_j$, and $A_{\mathbf{T}_j}$ is the area of $\mathbf{T}_j$. For an interior point $\hat{\mathbf{P}}$ of a mesh edge $\mathbf{E}_i$, with $\mathbf{V}_{i1}, \mathbf{V}_{i2}$ being the two vertices of $\mathbf{E}_i$, the highlight distance value of $\hat{\mathbf{P}}$ is obtained by performing linear interpolation on $d_{\mathbf{V}_{i1}}$ and $d_{\mathbf{V}_{i2}}$, i.e.,

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

Now we have highlight distance values for all points on mesh edges, we can choose the points satisfying Equation (2) to be the 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 (2) indicates that all points on $\mathbf{E}_i$ are highlight nodes with index m. We call such edge a *highlight edge*. Otherwise, there are a limited number of highlight nodes on $\mathbf{E}_i$. More precisely, there are highlight nodes on $\mathbf{E}_i$ only if $\lfloor d_{\mathbf{V}_{i1}}/s \rfloor$ and $\lfloor d_{\mathbf{V}_{i2}}/s \rfloor$ are different, where $\lfloor t \rfloor$ is the largest integer smaller than or equal to t. In this case, the index of any highlight node on $\mathbf{E}_i$ is between $\lfloor d_{\mathbf{V}_{i1}}/s \rfloor$ and $\lfloor d_{\mathbf{V}_{i2}}/s \rfloor$. According to Equation (2), for each integer m between $\lfloor d_{\mathbf{V}_{i1}}/s \rfloor$ and $\lfloor d_{\mathbf{V}_{i2}}/s \rfloor$, the position of the highlight node $\mathbf{Q}$ on $\mathbf{E}_i$ with index m can be located as

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



(a) (b) (c)

Figure 2. Possible cases of highlight node connection

2.4 Highlight node connection

After locating all the highlight nodes on the mesh, we connect them to form segments of the highlight lines. For each triangle, if there is no highlight edge, we connect highlight nodes on different edges with the same index (see Figure 2(a)). If at least one of the edges is a highlight edge, we connect the vertices of each highlight edge so that the highlight edge itself becomes a segment (see Figures 2(b) and 2(c)) of a highlight line. With such highlight node connection scheme, 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.

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

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

Input: A triangular mesh $\mathbf{M}$, and an array of coplanar parallel linear light source

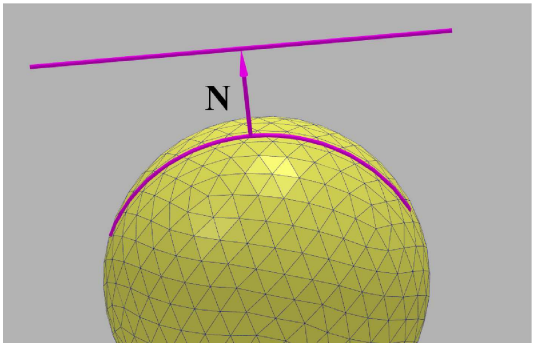
Output: The highlight line model of $\mathbf{M}$ corresponding to the light sources

```
Assign the set S_N of highlight nodes an empty set;
for each vertex V_i of M do
    Calculate the normal vector with Equation (3);
    Calculate the highlight distance value with Equation (1);
end
for each edge E_i of M do
    if the two vertices of E_i are highlight nodes with the same index then
        Add both vertices of E_i to S_N;
    else
        Calculate the highlight nodes on E_i with Equation (4);
        Add each highlight node on E_i to S_N;
    end
end
for each triangle T_i of M do
    Connect any nodes in S_N that lie on the edges of T_i and have the same index;
end
```

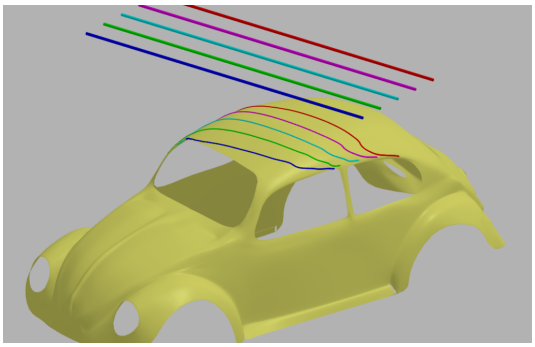
vertices in this region so that desired shape of the highlight lines can be constructed. The displacements of the mesh vertices are calculated by minimizing a target function that measures the fairness of the new mesh surface as well as the shape quality of the new highlight lines. Moving the vertices 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 irregular region by $\mathbf{R}$. Our goal is to improve the surface quality inside $\mathbf{R}$, without affecting the surface or highlight line model outside $\mathbf{R}$. Denote by S_{vertex} and S_{node} the sets of mesh vertices and highlight nodes outside $\mathbf{R}$, respectively. To keep the geometric properties and highlight lines outside $\mathbf{R}$ unchanged, the movement of the vertices should not change any of the following properties:



(a)



(b)

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

- normal vectors and positions of vertices in S_{vertex};
- 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, according to Equation (3). If $\mathbf{X}$ is a highlight node on an edge $\mathbf{E}_i$, 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). For us to improve the surface quality of $\mathbf{R}$ without affecting the surface or highlight line model outside $\mathbf{R}$, we can only move vertices of $\mathbf{R}$ that do not belong to the support vertices of

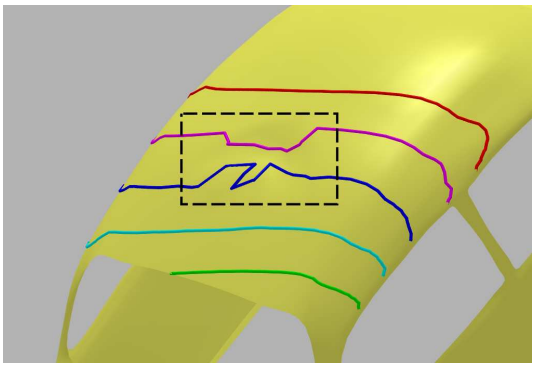


Figure 4. An irregular region of a mesh specified by the user.

S_{vertex} and S_{node}. Those vertices will be called *movable vertices* of $\mathbf{R}$.

3.2 Desired highlight lines

In this section we show how to construct highlight lines with desired shape for the specified region $\mathbf{R}$. The idea is to replace the undesired portion of a highlight line with an interpolating curve with desired shape. We assume the mesh surface outside $\mathbf{R}$ is of good quality, and we 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. These highlight nodes are called $\mathbf{Q}_0$ and $\mathbf{Q}_1$ (see Figure 5). They will be used as the end points of an interpolating curve. The tangent vectors of the highlight line at $\mathbf{Q}_0$ and $\mathbf{Q}_1$ will be called $\mathbf{T}_0$ and $\mathbf{T}_1$, respectively. The interpolating curve to be constructed 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 [27], a Hermite curve could have undesired loop, cusp, or fold. We will use an *optimized geometric Hermite (OGH) curve* [27] 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 [27]. The OGH curve segment satisfying the above interpolation conditions is of the following form

$$\begin{aligned}\mathbf{H}(t) &= (2t+1)(t-1)^2\mathbf{Q}_0 + (-2t+3)t^2\mathbf{Q}_1 \\ &\quad +(1-t)^2 t a_0 \mathbf{T}_0 + (t-1)t^2 a_1 \mathbf{T}_1, t \in [0,1],\end{aligned} \tag{5}$$

where

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

This curve segment will be used to replace the segment of the highlight line between $\mathbf{Q}_0$ and $\mathbf{Q}_1$. Figure 5 shows an example of an OGH curve segment constructed in this way.

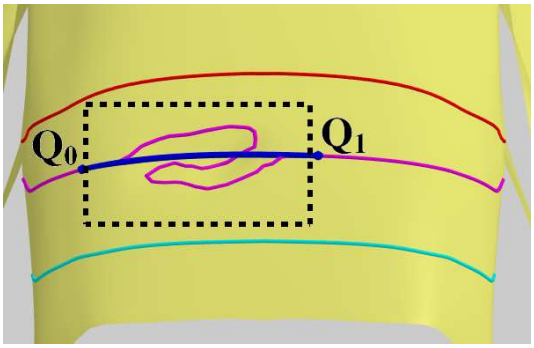


Figure 5. An OGH curve constructed for the specified region.

3.3 Vertex displacement calculation

With the desired highlight lines constructed, we are now ready to improve the shape of $\mathbf{R}$. This is done by adjusting some of the vertices of $\mathbf{R}$ so that, afterward, highlight line pattern of the region would be close to that of the constructed highlight lines and, consequently, 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 $\mathbf{V}_i$ will be adjusted along the direction of its unit normal vector $\mathbf{N}_i$, computed with Equation (3) during calculation of the highlight line model. For each $i \in I_M$, let $\overline{\mathbf{V}}_i$ be the new position of $\mathbf{V}_i$ after the adjustment. We have $\overline{\mathbf{V}}_i = \mathbf{V}_i + x_i\mathbf{N}_i$, where x_i is the displacement of $\mathbf{V}_i$. Let $\mathbf{X}$ be a displacement vector whose components are values $\{x_i | i \in I_M\}$. The basic idea of our method is to 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 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 [24] 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

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

where H_i is the mean curvature at vertex $\mathbf{V}_i$, and A_i is the mesh surface area associated with $\mathbf{V}_i$. Here the area A_i is computed as

$$A_i = \frac{1}{3} \sum_{j \in t(i)} A_{\mathbf{T}_j}, \tag{7}$$

where $t(i)$ and $A_{\mathbf{T}_j}$ are the same as in Equation (3), i.e., A_i equals $\frac{1}{3}$ of the total areas of the triangles adjacent to $\mathbf{V}_i$. H_i^2 can be obtained with 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 $\mathbf{V}_i$ [**?**]. Then

$$H_i^2 = \|H_i \mathbf{N}_i\|^2 = \mathbf{K}(\mathbf{V}_i) \cdot \mathbf{K}(\mathbf{V}_i).$$

Here $\mathbf{K}(\mathbf{V}_i)$ is calculated using the positions of $\mathbf{V}_i$ and its adjacent vertices [**?**]:

$$\mathbf{K}(\mathbf{V}_i) = \frac{1}{A_i} \sum_{j \in u(i)} (\cot \alpha_j + \cot \beta_j)(\mathbf{V}_i - \mathbf{V}_j), \tag{8}$$

where A_i is the same as in Equation (6), $u(i)$ denotes the index set of the 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). To derive the Willmore energy for mesh surface in region $\mathbf{R}$ after adjustment of the vertices, we need the new positions of the vertices $\{\mathbf{V}_j | j \in I_R\}$, which are

$$\overline{\mathbf{V}}_j = \begin{cases} \mathbf{V}_j + x_j \mathbf{N}_j, & \text{if } j \in I_M, \\ \mathbf{V}_j, & \text{otherwise.} \end{cases} \tag{9}$$

Substituting these new positions into Equations (6), (7) and (8), we have the expression of f_{fair} as a function of displacement $\{x_i | i \in I_M\}$.

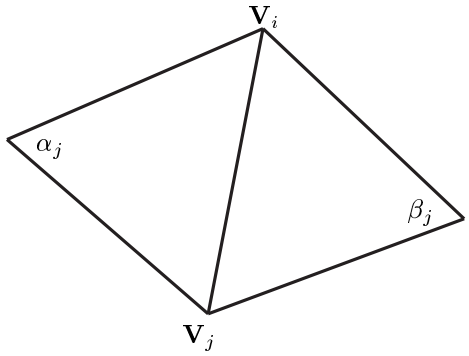


Figure 6. The angles α_j and β_j.

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. They are outside the irregular region. 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 $\overline{\mathbf{L}}(s), s \in [0,1]$ be the normalized chord-length parameterization forms of the highlight line between these two delimiting nodes before and after the vertex adjustment, respectively. Let $\widetilde{\mathbf{H}}(s), s \in [0,1]$ be the normalized arc-length parameterization form of the corresponding OGH curve. We define the difference function between the new highlight line $\overline{\mathbf{L}}(s)$ and its target shape $\widetilde{\mathbf{H}}(s)$ as

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

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

$$f_{\text{diff}} = \sum_{\overline{\mathbf{L}}} f_{\overline{\mathbf{L}}} \, . \tag{10}$$

Function $f_{\overline{\mathbf{L}}}$ can be discretized in the following way. First each highlight node $\mathbf{Q}_i$ on the original highlight line $\mathbf{L}(s)$ is mapped to a point $\widetilde{\mathbf{Q}}_i$ on $\widetilde{\mathbf{H}}(s)$. We call $\widetilde{\mathbf{Q}}_i$ the target position of $\mathbf{Q}_i$. After adjustment of the vertices, the corresponding new position $\overline{\mathbf{Q}}_i$ of $\mathbf{Q}_i$ is computed, and $f_{\overline{\mathbf{L}}}$ is given by

$$f_{\overline{\mathbf{L}}} = \sum_i \|\overline{\mathbf{Q}}_i - \widetilde{\mathbf{Q}}_i\|^2 l_i \, , \tag{11}$$

where l_i is the length of the highlight line segment associated with $\mathbf{Q}_i$. The following will provide the details on how to obtain $\widetilde{\mathbf{Q}}_i$, $\overline{\mathbf{Q}}_i$ and l_i.

To determine the target position of a highlight node $\mathbf{Q}_i$, we need the normalized chord-length parameter of $\mathbf{Q}_i$ on

$\mathbf{L}(s)$. Denote the two delimiting nodes of $\mathbf{L}(s)$ by $\mathbf{Q}_0$ and $\mathbf{Q}_{n+1}$, and the n highlight nodes between them by $\mathbf{Q}_i$ $(i = 1, 2, \ldots, n)$, with $\mathbf{Q}_0, \mathbf{Q}_1, \ldots, \mathbf{Q}_n, \mathbf{Q}_{n+1}$ being in the same order as they appear on the highlight line. For each node $\mathbf{Q}_i (i = 1, 2, \ldots, n)$, its normalized chord-length parameter $c(\mathbf{Q}_i)$ is given by

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

Now the target position $\widetilde{\mathbf{Q}}_i$ is determined as the point on $\widetilde{\mathbf{H}}(s)$ with parameter $s = c(\mathbf{Q}_i)$, i.e.,

$$\widetilde{\mathbf{Q}}_i = \widetilde{\mathbf{H}}(c(\mathbf{Q}_i)) . \tag{12}$$

The new position $\overline{\mathbf{Q}}_i$ of highlight node $\mathbf{Q}_i$ is calculated as follows. Let $\mathbf{E}_j$ be an edge that $\mathbf{Q}_i$ lies on, with $\mathbf{V}_{i1}$, $\mathbf{V}_{i2}$ being the vertices of $\mathbf{E}_j$. Since $\overline{\mathbf{Q}}_i$ and $\mathbf{Q}_i$ correspond to the same light source, they should have the same index. According to Equation (4), the new position $\overline{\mathbf{Q}}_i$ of $\mathbf{Q}_i$ 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 $\mathbf{Q}_i$,

$$\overline{\mathbf{Q}}_i = \frac{(m \cdot s - d_{\overline{\mathbf{V}}_{i2}})\overline{\mathbf{V}}_{i1} + (d_{\overline{\mathbf{V}}_{i1}} - m \cdot s)\overline{\mathbf{V}}_{i2}}{d_{\overline{\mathbf{V}}_{i1}} - d_{\overline{\mathbf{V}}_{i2}}} . \tag{13}$$

Here $\overline{\mathbf{V}}_{i1}$ and $\overline{\mathbf{V}}_{i2}$ are the new vertex positions obtained with Equation (9). $d_{\overline{\mathbf{V}}_{i1}}$ and $d_{\overline{\mathbf{V}}_{i2}}$ are the new distance values calculated from Equation (1). $d_{\overline{\mathbf{V}}_{i1}}$ and $d_{\overline{\mathbf{V}}_{i2}}$ are both determined by the new vertex positions, and $\overline{\mathbf{Q}}_i$ is a function of the displacement $\mathbf{X}$. Note that if $\mathbf{Q}_i$ is a vertex of the mesh, then $\mathbf{E}_j$ can be any edge adjacent to the vertex. In this case, Equation (13) still holds, and we obtain that $\overline{\mathbf{Q}}_i$ is at the new position of the vertex obtained by Equation (9).

For a highlight node $\mathbf{Q}_i$, its associated highlight segment length l_i is calculated as half of the total length of the highlight line segments that it lies on. Note that $\mathbf{Q}_i$ should have exactly two neighboring highlight nodes on $\mathbf{L}(s)$. Denote these two neighboring nodes by $\mathbf{Q}_{i-}$ and $\mathbf{Q}_{i+}$, then

$$l_i = \frac{1}{2} \left(\|\mathbf{Q}_i - \mathbf{Q}_{i-}\| + \|\mathbf{Q}_i - \mathbf{Q}_{i+}\| \right) . \tag{14}$$

Substituting Equations (12), (13) and (14) into Equations (10) and (11), 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 $\{x_i\}$, which makes minimization of the target function a very expensive numerical process. We will 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 (8) 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 $\{x_i\}$. Now the target function becomes

$$F = \omega_1 q_{\text{fair}} + \omega_2 q_{\text{diff}} . \tag{15}$$

In addition, we put the following constraint on the components of the displacement

$$|x_i| <= \frac{1}{2} \min_{j \in s(i)} e_j, \text{ for all } i \in I_M ,$$

where $s(i)$ is the set of indices of edges adjacent to vertex $\mathbf{V}_i$, and e_j is the length of the edge with index j. 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 can be formulated as

$$\begin{cases} \text{minimize } F, \\ \text{subject to } |x_i| <= \frac{1}{2} \min_{j \in s(i)} e_j, \ i \in I_M , \end{cases} \tag{16}$$

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

3.7 Iteration

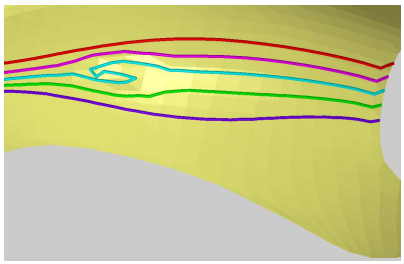
We use an iterative procedure to gradually improve the quality of the irregular region. In each iteration step, a quadratic programming problem is formed using current geometric information of the mesh. Then we solve the minimization problem, and adjust the vertices according to the solution to obtain 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.,

$$\frac{\max_{i \in I_M} |x_i|}{e} < \varepsilon , \tag{17}$$

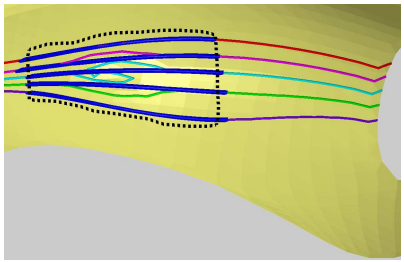
where e is the average edge length inside the irregular region $\mathbf{R}$, and ε is a positive threshold value specified by the user. The iterative procedure is summarized in Algorithm 2.

4 Implementation and examples

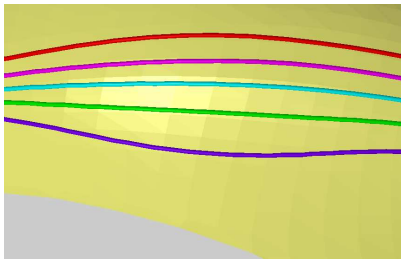
Here we show implementation results of the presented method on some mesh models. In these examples, we set $\omega_1 = \omega_2 = 1$ for the target function in Equation (15), and set $\varepsilon = 0.001$ for the termination condition specified in Formula (17). Figure 7 shows the fairing of the mesh 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



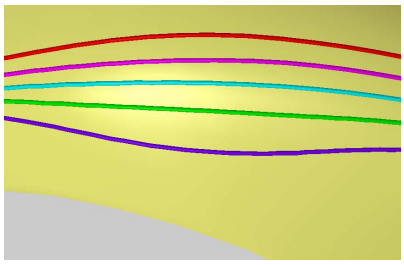
(a)



(b)



(c)



(d)

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

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 $\mathbf{R}$

Identify the set of movable vertices $\{\mathbf{V}_i | i \in I_M\}$ of $\mathbf{R}$;
Set the number of iterations $n = 0$;
repeat
    Construct the target function F of displacements $\{x_i | i \in I_M\}$;
    Solve the minimization problem (16) to obtain the values of $\{x_i | i \in I_M\}$;
    for *each* $i \in I_M$ **do**
        Adjust vertex $\mathbf{V}_i$ according to x_i;
    end
    Update the highlight line model of the mesh using Algorithm 1;
    Set $n = n + 1$;
until $n > N_{max}$ *OR* $\max_{i \in I_M} |x_i|/e < \varepsilon$;

region specified for faring, 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, repectively. The new mesh surface in the faired region is of high quality; shapes of 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[6]. In Figure 8, we fair another irregular region on the roof of the Beetle model. To compare our method with existing mesh fairing techniques, we also implemented the surface diffusion flow technique proposed by Xu et al.[25] which is able to satisfy G^1 boundary conditions. Their method moves each vertex $\mathbf{V}_i$ inside the region to be faired along their normal direction, using equation $\partial_t \mathbf{V}_i = \mathbf{N}_i \Delta_B H_i$, where Δ_B is the Laplace-Beltrami operator, and $\mathbf{N}_i$ and H_i are the surface normal and mean curvature at $\mathbf{V}_i$, respectively. The resulting mesh of this method satisfies equation $\Delta_B H_i = 0$ for each vertex inside the fairing region. This fourth-order PDE is also used by Schneider and Kobbelt [18][19] in G^1 mesh fairing. 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, causing abrupt changes of highlight line shapes on

the left and right boundary of the region. One the other hand, our method produces a mesh surface with new highlight lines close to the desired ones, making the new highlight lines naturally blend the shapes of the two highlight lines outside the selected region. This example shows that although existing mesh 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 shapes. Therefore, our method can improve the shape quality of both the mesh surface and the highlight line model.

5 Conclusions and future work

A method of generating highlight line model for a given triangular mesh is provided in this paper. With a highlight line model, the job of identifying irregular regions of a mesh is reduced to that of identifying irregular portions of highlight lines. Subsequently, a method for removing local irregularities of a given triangular mesh is presented. The modification process is intuitive. A user only needs to identify the irregular regions, the modification is carried out automatically. Test cases show that the new method is capable of making the highlight line models converge to the desired shapes. 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. As far as limitations are concerned, the new method does not guarantee the new highlight lines would coincide with the desired ones, and it may fail when the irregular region is relatively large. How to eliminate these limitations will be a subject of future research.

Acknowledgements

We would like to thank Leif Kobbelt for providing us with the Volkswagen Beetle model. The research was supported by Chinese 973 Program(2004CB719400), and the National Science Foundation of China (60403047, 60533070). 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 by KSTC under grant COMM-Fund-712.

References

[1] M. Alexa. Wiener filtering of meshes. In *Proceedings of Shape Modeling International 2002*, pages 51–60, 2002.

[2] C. L. Bajaj and G. Xu. Anisotropic diffusion of surfaces and functions on surfaces. *ACM Transactions on Graphics*, 22(1):4–32, 2003.

[3] K.-P. Beier and Y. Chen. Highlight-line algorithm for real-time surface-quality assessment. *Computer-Aided Design*, 26(4):268–277, 1994.

[4] A. I. Bobenko and P. Schröder. Discrete willmore flow. In *Symposium of Geometry Processing*, pages 101–110, 2005.

[5] M. Botsch, M. Pauly, C. Rössl, S. Bischoff, and L. Kobbelt. Geometric moeling based on triangle meshes. 2006. SIGGRAPH 2006 Course Notes.

[6] Y. Chen, K.-P. Beier, and D. Papageorgiou. Direct highlight line modification on NURBS surfaces. *Computer Aided Geometric Design*, 14(6):583–601, 1997.

[7] U. Clarenz, U. Diewald, and M. Rumpf. Anisotropic geometric diffusion in surface processing. In *Proceedings of IEEE Visualization 2000*, pages 397–406, 2000.

[8] M. Desbrun, M. Meyer, P. Schröder, and A. Barr. Implicit fairing of irregular meshes using diffusion and curvature flow. In *Proceedings of SIGGRAPH 99*, pages 317–324, 1999.

[9] S. Fleishman, I. Drori, and D. Cohen-Or. Bilateral mesh denoising. *ACM Transactions on Graphics*, 22(3):950–953, 2003.

[10] K. Hildebrandt and K. Polthier. Anisotropic filtering of non-linear surface features. *Computer Graphics Forum*, 23(3):391–400, 2004.

[11] T. R. Jones, F. Durand, and M. Desbrun. Non-iterative, feature-preserving mesh smoothing. *ACM Transactions on Graphics*, 22(3):943–949, 2003.

[12] L. Kobbelt. Discrete fairing. In *Proceedings of the 7th IMA Conference on the Mathematics of Surfaces*, pages 101–131, 1997.

[13] L. Kobbelt, S. Campagna, J. Vorsatz, and H.-P. Seidel. Interactve multi-resolution modeling on arbitrary meshes. In *Proceedings of the SIGGRAPH 98*, pages 105–114, 1998.

[14] X. Liu, H. Bao, P. A. Heng, T. T. Wong, and Q. Peng. Constrained fairing for meshes. *Computer Graphics Forum*, 20(2):115–123, 2001.

[15] Y. Ohtake, A. Belyaev, and I. A. Bogaevski. Polyhedral surface smoothing with simultaneous mesh regularization. In *Proceedings of Geometric Modeling and Processing 2000*, pages 229–237, 2000.

[16] Y. Ohtake, A. Belyaev, and H.-P. Seidel. Mesh smoothing by adaptive and anisotropic gaussian filter applied to mesh normals. In *Proceedings of Vision, Modeling and Visualization 2002*, pages 203–210, 2002.

[17] J. Peng, V. Strela, and D. Zorin. A simple algorithm for surface denoising. In *Proceedings of IEEE Visualization 2001*, pages 107–112, 2001.

[18] R. Schneider and L. Kobbelt. Generating fair meshes with G^1 boundary conditions. In *Proceedings of Geometric modeling and Processing 2000*, pages 251–261, 2000.

[19] R. Schneider and L. Kobbelt. Geometric fairing of irregular meshes for free-form surface design. *Computer Aided Geometric Design*, 18(4):359–379, 2001.

[20] T. Tasdizen, R. Whitaker, P. Burchard, and S. Osher. Geometric surface smoothing via anisotropic diffusion of normals. In *Proceedings of IEEE Visualization 2002*, pages 125–132, 2002.

[21] G. Taubin. A signal processing approach to fair surface design. In *Proceedings of SIGGRAPH 95*, pages 351–358, 1995.

[22] J. Vollmer, R. Mencl, and H. Müller. Improved laplacian smoothing of noisy surface meshes. *Computer Graphics Forum*, 18(3):131–138, 1999.

[23] W. Welch and A. Witkin. Free-form shape design using triangulated surfaces. In *Proceedings of SIGGRAPH 94*, pages 157–166, 1994.

[24] T. J. Willmore. *Riemannian Geometry*. Oxford University Press, 1993.

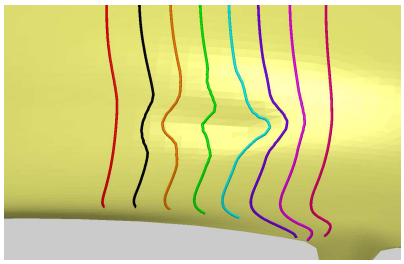
[25] G. Xu, Q. Pan, and C. Bajaj. Discrete surface modelling using partial differential equations. *Computer Aided Geometric Design*, 23(2):125–145, 2006.

[26] H. Yagou, Y. Ohtake, and A. Belyaev. Mesh smoothing via mean and median filtering applied to face normals. In *Proceedings of Geometric modeling and Processing 2002*, pages 124–131, 2002.

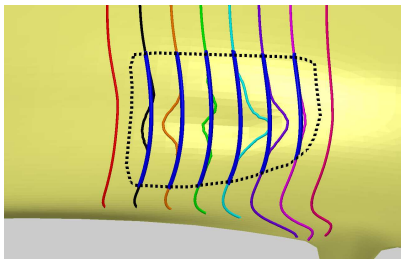
[27] J.-H. Yong and F. Cheng. Geometric hermite curves with minimum strain energy. *Computer Aided Geometric Design*, 21(3):281–301, 2004.

[28] J.-H. Yong, F. Cheng, Y. Chen, P. Stewart, and K. T. Miura. Dynamic highlight line generation for locally deforming nurbs surfaces. *Computer-Aided Design*, 35(10):881–892, 2003.

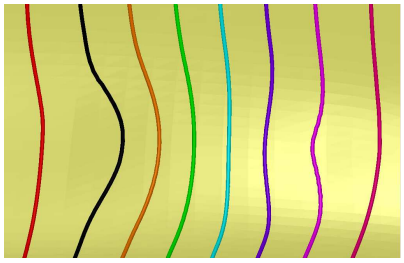
[29] S. Yoshizawa and A. G. Belyaev. Fair triangle mesh generations via discrete elastica. In *Proceedings of Geometric Modling and Processing 2002*, pages 119–123, 2002.



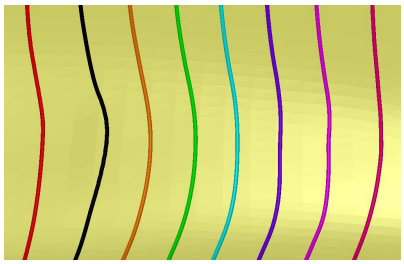
(a)



(b)



(c)



(d)

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.