Feature Enhancement by Vertex Flow for 3D Shapes

Zhongping Ji[1], Ligang Liu[2], Bin Wang[3] and Wenping Wang[4]

[1]Hangzhou Dianzi University, jzp@hdu.edu.cn
[2]Zhejiang University, ligangliu@zju.edu.cn
[3]Tsinghua University, wangbins@tsinghua.edu.cn
[4]The University of Hong Kong, wenping@cs.hku.hk

ABSTRACT

We present a new method for enhancing shape features of a mesh surface by moving mesh vertices from low-curvature regions to high-curvature regions or feature regions. The movement of the vertices, also called vertex flow, is driven by minimizing an objective function defined to take into account several important considerations in mesh improvement. First, a new edge-based energy term is used to measure the uniformity of the approximation error of a target shape by a mesh surface. Clearly, given a fixed number of triangle faces of a mesh surface approximating an underlying target surfaces, the approximation is made more uniform by placing more faces are used in higher-curvature regions and fewer faces in lower-curvature regions. Therefore, the minimization of this edge-based energy term provides a strong force to move mesh vertices towards high-curvature regions. Second, to maintain faithful shape approximation during vertex flow, a distance-error term is included to penalize the displacement of mesh vertices along normal directions of the underlying surface, and a novel local quadratic model is employed to efficiently minimize this term. Third, a fairing term is used to ensure the smoothness of the resulting mesh surface. Our method enhances significantly shape features even at a low sampling rate and is useful to several feature-aware geometry processing operations, such as simplification and perceptual line drawing.

Keywords: feature enhancement, vertex flowing, mesh improvement, optimization.
DOI: 10.3722/cadaps.2011.649-664

1 INTRODUCTION

Triangular meshes are the most versatile surface representation in computer graphics. Triangular meshes are generated from a variety of sources including 3D scanners, modeling software, and computer vision algorithms. Different triangulations of the same surface are needed to suit for specific requirements of different applications. When an underlying, or original, shape is available, the error between this shape and its approximating mesh may be reduced by dense sampling. However, over-sampling will increase the number of vertices and thus the associated complexity, transmission

and processing costs. We propose a method that optimizes mesh approximating quality with feature preservation without increasing the number vertices.

We wish to achieve a uniform distribution of approximation errors of a mesh to a given underlying target surface with fixed mesh complexity. Clearly, to attain the same approximation error, more faces are needed in a higher curvature region or around features and fewer faces in flat regions, such as a flat part of a surface. In our approach, this desired distribution of mesh faces is achieved by moving mesh vertices on the surface via the minimization of an edge-based energy, which is shown to measure the uniformity of the approximation.

To ensure faithful and smooth shape approximation by the modified mesh surfaces, we prevent large displacements of the mesh vertices away from the underlying surface by incorporating an error term measuring the distance from mesh vertices from the underlying surface. Efficient minimization of this distance term, which is necessarily nonlinear, is achieved by employing a local quadratic model proposed in [16]. A distinct feature of this model is that it allows easy vertex flow with the tangent space of a surface, while inhibiting vertex flow in the normal direction. Finally, a standard faring term is included in the objective function to ensure the smoothness of the resulting mesh surface.

Fig.1 shows an example in which the features of a mesh surface are enhanced with our method. Our method works directly on a mesh surface of an arbitrary topological type without the need for mesh decomposition and mesh parametrization. In the present work we focus on feature enhancement with only vertex relocation in an optimization framework, assuming that the edge connectivity of the mesh surface remain unchanged. However, there is no essential difficulty in combining the idea presented here with an appropriate edge-flip scheme to further improve mesh approximation quality.

Fig.1: A feature enhancement example: (a) A mesh surface with 200k vertices, (b) A down-sampled mesh of the same surface with 6k vertices, which loses some features due to the low sampling rate, (c) An improved mesh generated from (b) by our method, without using (a). Some of features (such as eyes, nose, and feet) are restored or enhanced, as compared with (a).

The remainder of the paper is organized as follows. In Section 2, we briefly review related work. The problem is formulated and our optimization framework is discussed in Section 3. Section 4 addresses key implementation issues. Some experimental results are presented in Section 5. We conclude the paper in Section 6 with a summary and discussions on future work.

2 RELATED WORK

There are several topics related to our work, including mesh optimization [8,14], mesh smoothing [9], surface resampling and remeshing [1,21]. We will review briefly related works on these topics in the following.

Mesh improvement. Mesh improvement produces an improved mesh approximating a given original surface but with more reasonable distributed vertices. Mesh quality improvement methods include several operations: vertex insertion or collapse, edge swapping, local vertex relocation, and combinations of these.

Vertex insertion/collapse, vertex relocation and edge swapping are local operators that are frequently used in mesh improvement. Hoppe et al. [8] describe an energy minimization approach to mesh optimization. Vertices are inserted or deleted from the mesh based on a priority queue according to approximation errors. Turk [21] proposes a method for distributing a given number of points over a mesh surface uniformly. These points are connected to form a re-tiling of a surface that is faithful to both the geometry and the topology of the original mesh surface. Recently, mesh optimization using global *Laplacian* operator is proposed [13,14]. Vertices are relocated so that they approximate prescribed *Laplacians* and positions in a least-squares sense.

Remeshing of surfaces aims to create a new mesh with high quality. Generally, remeshing techniques are classified into five main categories: structured remeshing, high quality remeshing, compatible remeshing, feature remeshing, and error-driven remeshing [2]. Feature remeshing focused on preserving or even restoring sharp features when producing the resulting meshes. Vorsatz et al. [21] proposed another way for feature preserving remeshing where mesh vertices are extracted to feature edges using a hierarchical curvature field. Some approaches perform connectivity optimization with vertex relocation [23], assisted by global or local surface parameterization, so mesh vertices need to be mapped to the original surface after relocation in the parameter domain.

Feature enhancement. Guskov et al. proposed an algorithm for feature enhancement of mesh surfaces based on multiresolution signal processing [7]. This filtering technique is directly applied to mesh vertices locally, so triangle flipping frequently occurs [24]. Yagou et al. developed a 3D Shape enhancement method based on the high-boost filter in signal and image processing [24]. It consists of affecting the high-boost filter to face normals on a triangle mesh and updating mesh vertex positions to make them adapt to the boosted normals. However, this method will introduce the aliasing and the mesh irregularization which are suppressed by the *Laplacian* smoothing therein. The edge-sharpener algorithm [14] detects the chamfer edges and inserts new vertices which are forced to lie on intersections of planes at sharp features [14]. Lai et al. [11] proposed a classification and editing framework for triangular meshes. They used a feature sensitive metric to recognize features on multiple scales via integral invariants of local neighborhoods. The dilation and erosion of prong features are one of their applications. Recently, Eigensatz et al. [5] provided a shape editing tool based on curvature domain processing. Surface curvature is direct accessed to facilitate shape processing, including filtering, synthesis and feature enhancement algorithms. Feature enhancement is achieved by amplifying the largest absolute principal curvature by an amount proportional to the difference of the absolute principal curvature values. This change enhances convex ridges as well as concave valleys. However, their optimization entails solving a time-consuming nonlinear least-squares problem.

Other related works are methods for exaggerating the shading features. Without modifying the geometry of surfaces, these methods modify the normals (like in bump mapping, normal mapping) or the light position. Cignoni et al. [4] presented a simple technique to improve the perception features of 3D shapes. Based on a simple modification of the surface normals, the geometric features of the object are enhanced during the rendering. Rusinkiewicz et al. [18] proposed a non-photorealistic shading model, where the effective light positions for different areas of the surface are dynamically adjusted. It reveals detail regardless of surface orientation and computes the shading at multiple scales to bring out detail while conveying overall shape. Ritschel et al. [17] presented an approach for 3D scene enhancement by increasing the local contrast. Using unsharp masking, their method avoids temporal artifacts, better depicts shape details, provides clearer separation between objects and improves the overall dynamic range.

3 FEATURE ENHANCED VERTEX FLOWING

A triangular mesh M is a triple (V,E,T), where $V=\{v_1,v_2,\ldots,v_n\}, v_i \in R^3$ is the set of vertex positions defining the shape of the mesh in R^3; $E=\{e_i \mid i=1,2,\ldots,l\}$ is the set of edges in which each edge can be represented by a pair of vertex indices as $e_i=(i_1,i_2), 1\le i_1,i_2\le n$; and $T=\{t_i \mid i=1,2,\ldots,m\}$ is the set of triangles in which each triangle t_i can be represented by a triple of vertex indices as $t_i=(i_1,i_2,i_3), 1\le i_1,i_2,i_3\le n$. Denote $|X|$ as the cardinality of the set X.

3.1 Objective Function

Our goal is to make the vertices flow over the mesh surface from flat regions with low curvatures to featured regions with high curvatures, thus enhancing surface features and the perceptual appearance of the shape. The basic idea is that the flow of the vertices can be induced by minimizing appropriate energy function via quadratic minimization.

Starting from their positions in the input mesh M_0, the mesh vertices will be updated to minimize the following objective function F_{total} to form an updated mesh M:

$$\arg\min_{V^*} F_{total} = F_{em} + \lambda F_{sd} + \mu F_{smo}, \tag{1}$$

where F_{em} is the sum of edge-based energy terms for all the edges of M, which measure the uniformity of the approximation. Since in general the approximation errors are not uniformly distributed for a mesh surface with uniformly distributed mesh vertices, the minimization of this term F_{em} will induce the flow of mesh vertices from low-curvature regions to high-curvature regions. The second term F_{sd} is the sum of the squared distance from the vertices of M to the original surface M. We use the minimization of this term to prevent the mesh M from deviating much from the initial surface M_0. The last term F_{smo} is a term which makes the low-curvature regions smooth. The coefficients λ and μ are their associated weights that are used to balance their influences. We will elaborate on each of the above energy terms in the next section.

Hence, intuitively, by minimizing the above objective function, the mesh vertices are attracted to the nearby feature region (due to the term F_{em} term), and they are not are away from the initial surface M_0 (due to the term F_{sd}). As a result of this vertex flow, the features of the mesh are enhanced or sharpened.

3.2 Edge-metric Term - F_{em}

We first introduce an edge metric on a surface S and then use this edge metric to define the edge-based energy F_{em} of a mesh M. For each edge e, we define its energy $f_{em}(e)$ such that $f_{em}(e)$ returns a large value if e is a relatively long edge in a high-curvature region of the surface S; thus the minimization of $f_{em}(e)$ will make the mesh vertices have a more reasonable distribution.

Let e be an undirected edge, corresponding to two directed edge vectors e_{ij} and e_{ji}, incident to the vertices v_i and v_j. Consider the directed edge $e_{ij} = v_j - v_i$. Let T_{i1} and T_{i2} be unit vectors in the principal curvature directions of v_i. Let k_{i1} and k_{i2} be the principal curvatures at v_i. Then we define the edge metric at v_i for e_{ij} as

$$f_{v_i}(e_{ij}) = |k_{i1}|(e_{ij} \cdot T_{i1})^2 + |k_{i2}|(e_{ij} \cdot T_{i2})^2. \tag{2}$$

Similarly, define the edge metric at v_j for e_{ji} as

$$f_{v_j}(e_{ji}) = |k_{j1}|(e_{ji} \cdot T_{j1})^2 + |k_{j2}|(e_{ji} \cdot T_{j2})^2, \tag{3}$$

where T_{j1}, T_{j2}, k_{j1} and k_{j2} are evaluated at v_j. Then the energy $f_{em}(e)$ for the edge e is defined as

$$f_{em}(e) = \frac{1}{2}(f_{v_i}(e_{ij}) + f_{v_j}(e_{ji})). \tag{4}$$

Finally, the total edge-based energy for the mesh M is define as

$$F_{em} = \frac{1}{|E|}\sum_{e \in E} f_{em}(e), \tag{5}$$

where E is the set of all the edges of the mesh M.

It can be shown that the energy term $f_{em}(e)$ for edge e measures the deviation, or the maximum error, δ of the edge e from the surface S, see Fig.2. The proof is given in the Appendix. Clearly, the error δ will be large if the curvature of S is large (for an edge e of fixed length) or if the edge e is long (for a fixed curvature of S).

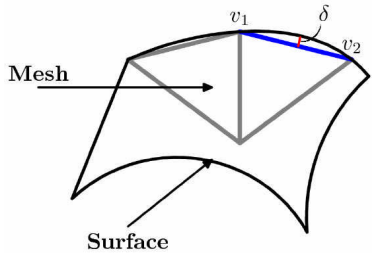

Fig. 2: The maximum error δ between the edge $e = v_1v_2$ and the surface S.

Note that, when used for optimization, only the endpoints v_i and v_j are allowed to vary in the expression of Fem and all the other quantities are treated as constants. Therefore F_{em} is a positive semi-definite (PSD) form in terms of the mesh vertices. The minimization of F_{em} amounts to minimizing the L_1 error between the mesh M and the surface S. Suppose that an input mesh M_0 approximating an underlying smooth surface S has its vertices uniformly distributed with respect to the surface area. Then the approximation error (M_0 to S) is larger in high curvature region than in low curvature region. In this case, the minimization tends to drive the vertices of M_0 from low-curvature regions to high-curvature regions.

3.3 Squared-distance Error - F_{sd}

To ensure that the updated mesh M remains an acceptable approximation to the underlying surface S, we include a distance error term F_{sd} to measure the distance between M and the initial mesh M_0 (since S is usually unavailable and M_0 can be taken as an proxy of S). The minimization of F_{sd} will penalize a large deviation of M from M_0. To encourage vertex flow in tangential directions, we introduce the SD error term which is defined by a quadratic approximate of the squared distance from a point to a surface. We will discuss the formulation of this quadratic model briefly in the following.

Consider a smooth surface S in 3D and a point $p \in S$. Let T_1 and T_2 be unit vectors in the principal curvature directions of S at p. Then the normal vector of S at p is $N = T_1 \times T_2$. Let $\rho_1 > 0$ and $\rho_2 > 0$ be the principal curvature radii of S at p. Let v_0 be a fixed point on the normal line of S through the point p. Let d be the minimum of the principal curvature radius of all vertices. Consider a variable point $v = (x, y, z)^T$ in the neighborhood of the point v_0. It is shown in [16] that a quadratic approximation of the squared distance function from the variable v to the surface S is

$$h(v) = \frac{d}{d + \rho_1}[(v - p) \cdot T_1]^2 + \frac{d}{d + \rho_2}[(v - p) \cdot T_2]^2 + [(v - p) \cdot N]^2, \tag{6}$$

which is a positive semi-definite quadratic form in terms of the point $v = (x, y, z)^T$. An iso-distance surface $h(v) = c$ for a constant c, is an ellipsoid centered at p. The orientation and shape of this

ellipsoid are determined by d and curvature information of S at p, i.e. T_1, T_2, ρ_1, and ρ_2. If p is in a low-curvature region an d is relatively small, the coefficients $d/(d+\rho_1)$ and $d/(d+\rho_2)$ are much smaller than 1. Then in this case the iso-distance surface $h(v)=c$ is a flat ellipsoid (see Fig.3).

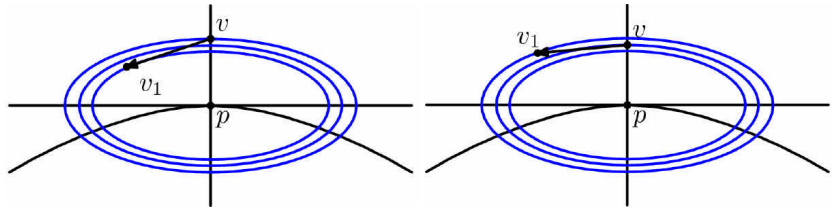

Fig. 3: (a) The *SD* error decrease when the point v moves to v_1; (b) The *SD* error increase slightly when the point v moves to v_1.

We use $h(v)$ given by Equation 6 as the error term to measure the error from a point v near the vertex p on the mesh M. During the process of optimization, the point may have to move sideway with respect to the vertex p. Such a movement may still result in a smaller error value when using the *SD* error term (see Fig.3(a)). Another possibility is that, a point may move away from p along a tangential direction of the surface S (see Fig.3(b)). In this case, the value of $h(v)$ will increase only slightly as compared with the (squared) Euclidean error term $\|v-p\|^2$. Hence, we may say that the *SD* error term has little resistance in a low-curvature region (or along a low-curvature direction if only one of the principal curvatures is small) for the tangential movement of mesh vertices; in contrast, such a movement or, for this matter, any movement of v away from p, is penalized heavily by the Euclidean error term $\|v-p\|^2$. This favorable property of the *SD* error term ensures that, with little resistance, the edge-based energy term F_{em} can effectively transport mesh vertices to arrive at a good distribution.

The total *SD* error is then defined as

$$F_{sd} = \frac{1}{|V|}\sum_{v\in V} h(v), \tag{7}$$

Clearly, F_{sd} is also a positive semi-definite quadratic form in terms of the mesh vertices.

3.4 Fairness Term - F_{smo}

The squared-distance error punishes the movement in the normal direction, which may influence the fairness of the triangles. To alleviate this effect, we introduce a term which defines the fairness and smoothness similar to [13,14]. That is, for each vertex v_i, we define its fairness as its *Laplacian* operator:

$$L(v_i) = v_i - \frac{1}{d_i}\sum_{e_{ij}\in E} v_j, \tag{8}$$

where d_i is the valence of the vertex v_i. It is seen that the vertex v_i lies in the center of gravity of its 1-ring neighbors if $L(v_i)=0$. The fairness energy of the mesh is defined as

$$F_{smo} = \sum_{v_i\in V} \mu_i \|L(v_i)\|^2, \tag{9}$$

where μ_i is the weight of v_i.

To preserve sharp features of the surface, the vertices in the high curvature regions could have small weights in the fairness energy. Therefore, the weights μ_i is defined as

$$\mu_i = e^{-1/k}, \tag{10}$$

where $k = |k_{i1}| + |k_{i2}|$.

4 IMPLEMENTATION ISSUES

In this section we discuss some implementation issues which concern about the minimization of the objective function.

4.1 Minimization of Edge-based Energy Term F_{em}

An edge-based energy terms F_{em} has been introduced so far. As a matter of fact, a global minimum of this term is attained when the mesh contracts to a single point, which is clearly not a feasible solution since in this case the mesh vertices do not lie on the mesh M. To circumvent this problem, we reformulate the edge-based energy term as follows. Consider the term of edge e

$$f_{em}(e) = \frac{1}{2}(f_{v_i}(e_{ij}) + f_{v_j}(e_{ji})). \tag{11}$$

$f_{em}(e)$ becomes minimum if and only if its gradient vectors with respect to both v_i and v_j are zero, i.e.

$$\nabla_{v_i} f_{em}(e) = \nabla_{v_j} f_{em}(e) = 0. \tag{12}$$

However, since we are only interested in how $f_{em}(e)$ can be reduced with the tangential movement of the vertices v_i and v_j, so we choose not to consider the normal components of the gradient vectors; that is, we change the conditions Equation 12 to

$$\nabla_{v_i} f_{em}(e) \cdot T_{i1} = \nabla_{v_i} f_{em}(e) \cdot T_{i2} = 0, \tag{13}$$

and

$$\nabla_{v_j} f_{em}(e) \cdot T_{j1} = \nabla_{v_j} f_{em}(e) \cdot T_{j2} = 0. \tag{14}$$

That is, we only require the tangential components of the gradient vectors of $f_{em}(e)$ to vanish. Thus we consider only the minimization of the quadratic function

$$\left|\nabla_{v_i} f_{em}(e) \cdot T_{i1}\right|^2 + \left|\nabla_{v_i} f_{em}(e) \cdot T_{i2}\right|^2 + \left|\nabla_{v_j} f_{em}(e) \cdot T_{j1}\right|^2 + \left|\nabla_{v_j} f_{em}(e) \cdot T_{j2}\right|^2. \tag{15}$$

Computationally, the above reformulation can easily be implemented as follow. Suppose that the total sum of all edge-based energy terms is expressed as the quadratic form $f_{em} = V^T A V$, where $V = (v_1^T, v_2^T, \ldots, v_{|V|}^T)$ is a $3|V|$ dimensional vector, and $A = (A_1, A_2, \ldots, A_{|V|})$ with each A_i being a $3|V| \times 3$ matrix. Then, with respect to an arbitrary mesh vertex v_i, we have

$$\nabla_{v_i} f_{em}(e) = 2A_i^T V. \tag{16}$$

Therefore ,

$$\nabla_{v_i} f_{em}(e) \cdot T_{i1} = 2T_{i1} A_i^T V, \tag{17}$$

and

$$\nabla_{v_i} f_{em}(e) \cdot T_{i2} = 2T_{i2} A_i^T V. \tag{18}$$

Hence, we just need to consider the minimization of the following sum

$$\sum_{v_i \in V} [(T_{i1} A_i^T V)^2 + (T_{i2} A_i^T V)^2]. \tag{19}$$

The coefficient matrix of this least-squares problem can easily derived from the original matrix A by matrix manipulation.

4.2 Minimization of Vertex-based Energy Terms F_{sd} and F_{smo}

The energy terms F_{sd} and F_{smo} are both vertex-based energy terms. Fortunately, these terms are all quadratic form. It leads to a linear system to minimize the quadratic function. We denote the matrices of F_{sd} and F_{smo} by B and C respectively. It is easy to know that B is a $3|V| \times 3|V|$ matrix and C is $3|V| \times 3|V|$.

4.3 Minimization of the Objective Function F_{total}

Minimizing the objective function F_{total} results in solving a linear system of equations

$$LX = b, \tag{20}$$

where L is a $8n \times 3n$ matrix for a mesh M which contains $n = |V|$ vertices. Here $X = (v_0, v_1, \cdots, v_{n-1})^T$ is a $3n$ dimensional vector, where $v_i = (x_i, y_i, z_i)$. For the over-determined system Equation 20, we solve it in a least-squares sense. Finally, we compute the Cholesky factorization for a $3n \times 3n$ sparse matrix, and a direct solver of [20] is used in our implementation.

4.4 Remarks

We need to use the curvature at the mesh vertices in our energy terms. There are various approaches for estimating curvature on discrete meshes. From our experience, the approach proposed by Rusinkiewicz [19] gives a nice performance on curvature evaluation. We adopt this approach to compute the curvatures on the vertices in all our experiments.

Any extra linear constraints on vertex coordinates can be easily integrated into our system. That is, we can add any other linear constraints on the vertices in Equation 20 and these linear constraints can be satisfied in a least-squares sense, as in [13]. For example, users could want to fix the positions of some important feature points. This can be done by introducing some position constraints in the optimization.

Note that the connectivity of the mesh is unchanged during the optimization. Due to the fairness term in the objective function, the triangulation quality after the optimization generally is good in most of our experiments. We also give users the option to perform the mesh improvement techniques like edge-flips and edge splits to improve the connectivity of the triangulation [2].

5 EXPERIMENTAL RESULTS

We have implemented our vertex flowing method on a Pentium IV 3GHz computer with 1G memory. We show some examples in this section to illustrate the applicability and flexibility of our approach.

There are three feature-aware energy terms in the objective function we use. We will first show how each error term affects the results. Fig.4 shows some examples using different weights λ, for the SD error term F_{sd}. The features become more prominent as the weight λ decreases. This means that more deviation from the initial mesh is allowed for enhancing the features.

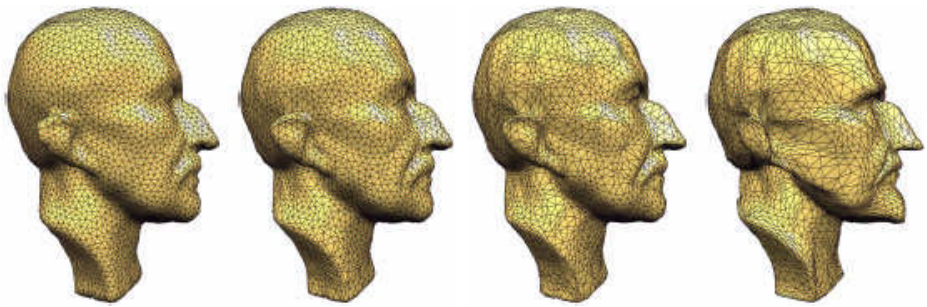

Fig. 4: Comparisons with different weights for F_{sd}. (a) a resampling mesh of Max-Plank model; (b),(c),(d) are the results by the vertex flowing algorithm with $\lambda = 50, 10, 0.2$ respectively.

The fairness error term F_{smo} improves the quality of the triangles in the vertex flowing. Fig. 5 shows some examples using different weights μ for F_{smo} and compares the qualities of the result meshes. Here we measure the triangle quality by the radius ratio $2r/R$, mapped to [0,1], where R and r are the radii of the circumscribed and inscribed circles respectively [15]. The quality of the triangles in (c) with larger μ is better than that in (a) and (b). It shows that the fairness term could improve the quality of the result mesh.

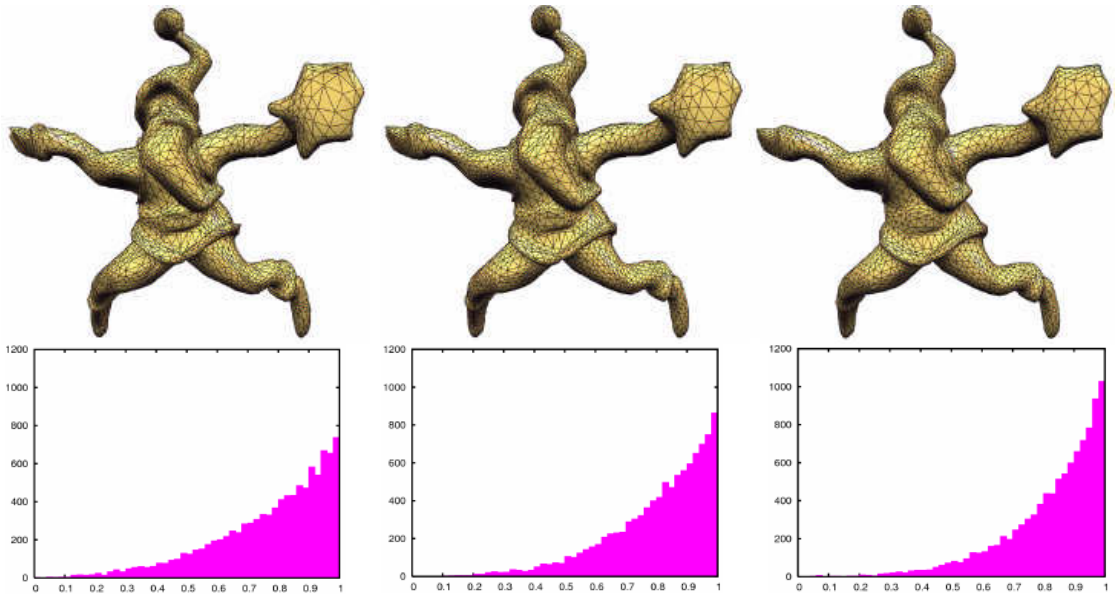

Fig. 5: Comparisons with different weights for F_{smo}. (from left to right) (a,b,c) are the results by vertex flowing algorithm with $\mu = 0, 0.5, 1$ respectively. The upper row shows the result meshes and the lower row shows the histogram of the triangle quality.

It can be seen that the weight parameters λ and μ measure the importance of the corresponding energy terms respectively. From our experimental experience, the weights $\lambda = 1$ and $\mu = 0.5$ could work well for most cases. We use these values in all the following examples.

Fig.6 shows an example of performing the vertex flowing minimization on a torus mesh surface. Note that the vertex density changes after the minimization as many vertices have been transported to high-curvature regions from low-curvature regions. Another example about the horse mesh is shown in Fig.7.

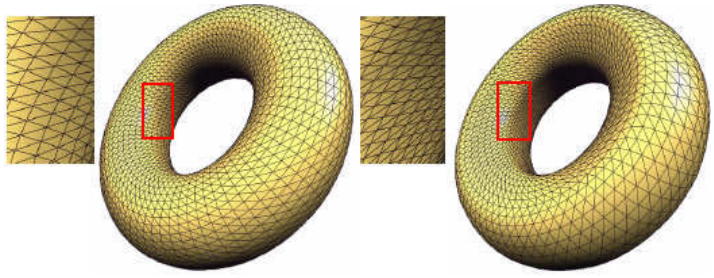

Fig. 6: Results by our approach. Left: the original torus mesh; Right: the result mesh using vertex flowing optimization.

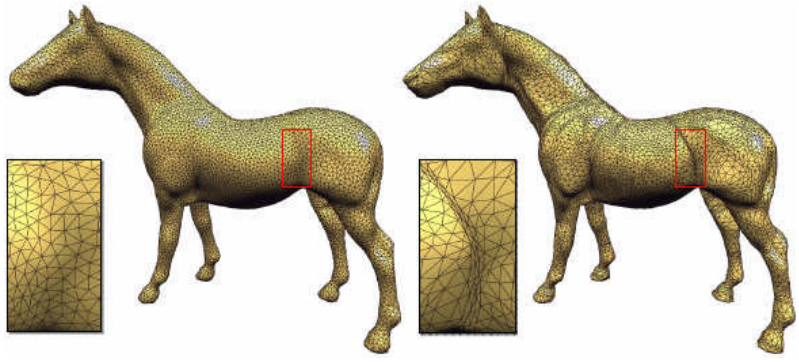

Fig. 7: Results by our approach. Left: a horse mesh; Right: the result mesh using vertex flowing optimization.

Comparisons. In Fig.8, we compare our approach with Guskov et al.'s approach [7] which can also be used to enhance mesh features. Its main idea is to extrapolate the difference between the original mesh and a single resolution relaxed mesh. As seen in the comparisons, features of the original surfaces could be well improved using our vertex flowing approach. When features on the mesh are not obvious or the sampling rate is low, the previous method may fail to enhance the features. As shown in the examples of this paper, our method can restore or enhance the features of low sampling rate models.

Fig. 8: Comparison with Guskov et al.'s approach [7]. Left: a resampling mesh of Armadillo model; Middle: result generated by Guskov's approach; Right: result by vertex flowing.

In Fig.9, we compare our approach with exaggerated shading generated in [18]. Instead of changing the geometry, the exaggerated shading method is based on a non-photorealistic shading model. It exaggerates the shading effect by adjusting the effective light position for different areas of the surface. This example shows that our method is capable of exaggerates different details.

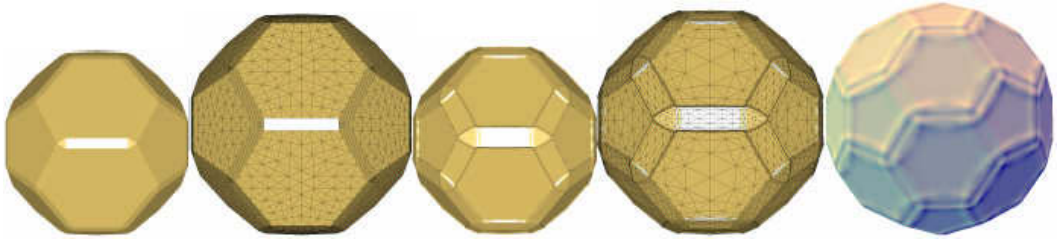

Fig. 9: Comparison with exaggerated shading [22]. (a) a football like model; (b) result by vertex flowing; (c) result generated by exaggerated shading.

Perceptual observation enhancement. Our vertex flowing method can enhance the perceptual observation for shapes, as shown in Fig.11. We used two state-of-the-art line drawing approaches to test the meshes. We use the same parameters for each pair models. The feature enhanced meshes always have better line drawing results than the original meshes. Therefore, our approach can help enhance the perceptual observation for mesh surfaces. Fig.10 shows the mean curvature maps of Max-Planck head models. Our method amplifies the high curvatures after vertex flowing.

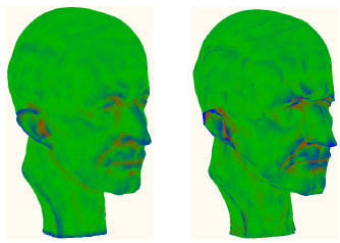

Fig. 10: Mean curvature maps. Color coded mean curvature: red (large, positive mean curvature) and blue (small, negative mean curvature). Left: a resampling mesh of Max-Plank model; Right: the result by vertex flowing.

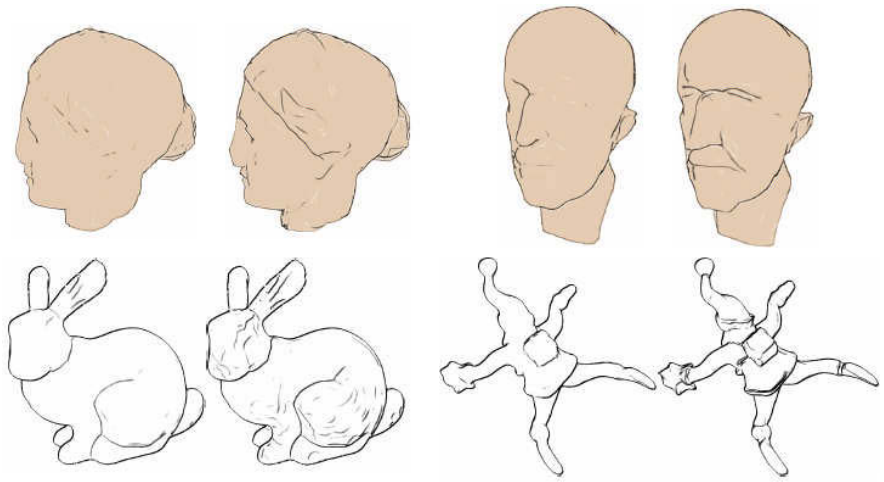

Fig. 11: Line drawing results of the meshes. (a),(b) line drawing results using the approach of [12]; (c),(d) results of apparent lines using the approach of [10]. For each pair in (a)-(d), the left mesh is the original mesh surface, the right one is the feature enhanced mesh by vertex flowing. It is obvious that the enhanced meshes by our approach can extract more perceptual information of the shapes.

Feature preserving LoD. Our framework of vertex flowing can be used to improve the quality of a simplified mesh as shown in Fig. 12. The original mesh shown in Fig. 12(a) is simplified using quadric error metrics method [6] shown in Fig. 12(b),(c). We generate a new mesh in Fig. 12(d) using the vertex flowing approach from Fig. 12(a). The feature enhanced mesh is simplified using the QEM method as well, obtaining Fig. 12(e),(f). We can see that the LoD sequence in the lower row preserve better features of the shape.

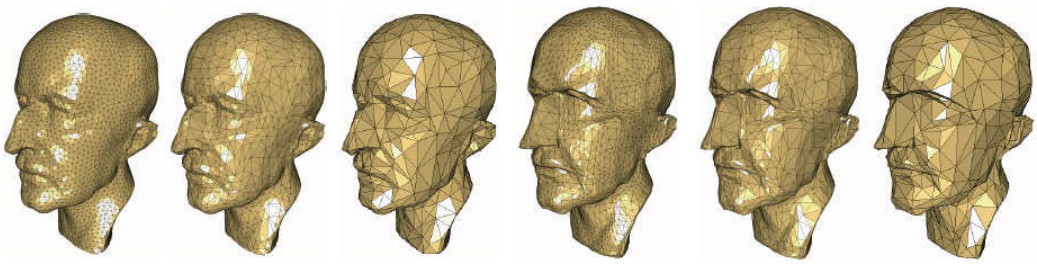

Fig. 12: Feature preserving LoD. (a) a resampling mesh of Max-Plank model; (d) the feature enhanced mesh using our approach; (a),(b),(c) are the LoD sequence using QEM approach [6]; (d),(e),(f) are the LoD sequence using QEM approach based on model pre-processing with vertex flowing method.

Table 1 lists the running time of the vertex flowing examples shown in this paper. As we can see, our approach achieves a good combination of speed, mesh quality, and feature improvement.

Model	*Vertex Number*	*Running Time (s)*
Torus(Fig. 6)	*4000*	*1.065*
Santa(Fig. 5)	*5000*	*1.305*
Venus head(Fig. 11(a))	*5000*	*1.328*
Max-Planck Head(Fig. 4)	*5000*	*1.347*

Dog(Fig. 1)	*6000*	*1.623*
Bunny(Fig. 11(c))	*6500*	*1.782*
Horse(Fig. 7)	*10000*	*2.812*
Armadillo(Fig. 8)	*10028*	*2.805*

Tab. 1: Running time for vertex flowing optimization shown in the paper. Except Fig. 4 and Fig. 5, we set the weights $\lambda = 1, \mu = 0.5$ for other examples.

Our optimization framework does not guarantee to preserve the orientation of the triangles. We rectify the topological errors in the post-processing if triangle flips occur. Fortunately, we rarely encounter this situation in our experiments due to the use of the *Laplacian* fairness term F_{smo}.

6 CONCLUSION

Mesh surfaces obtained by 3D scanners always lose their features to some degree because of low sampling rate. Our goal is to resume or enhance the features by the means of vertex flowing. We have presented an effective vertex flowing method that uses tangential flow of mesh vertices to achieve an optimal distribution of mesh vertices in a feature aware manner. The method is based on efficient quadratic optimization and works directly on a mesh of arbitrary topology type, i.e., without invoking complicated methods for mesh parameterization and mesh segmentation. The tangent flow of mesh vertices is achieved by integrating edge-based curvature-dependent energy term F_{em}, squared distance error term F_{sd}, and *Laplacian* fairing energy term F_{smo} in the objective function. The term F_{sd} constrains the improved mesh to the initial mesh and at the same time has little resistance to the tangential movement of mesh vertices and the term Fem induces a strong force to move mesh vertices from low-curvature regions to high-curvature regions.

We assume in the algorithm that the model surface is a mesh surface for which one estimate tangential and curvature information at each mesh vertex. However, the method works for a model surface given in any form, as long as the curvature information can be evaluated at each point of the surface; these include piecewise parametric surfaces or implicitly defined algebraic surfaces. For surfaces with boundaries, one needs to fix mesh vertices on the boundaries and only subject other internal vertices on optimization. Our further research is to combine the vertex flowing scheme presented here and other techniques of surface editing to yield a feature aware surface manipulation method.

Acknowledgement. We would like to thank Seungyong Lee and Tilke Judd for their kind help in applying their respective line drawing methods on the models in Fig. 11. This work is supported by the National Natural Science Foundation of China(61070071,90818011), Natural Science Foundation of Zhejiang Province (Y1111101) and the 973 National Key Basic Research Foundation of China (No. 2009CB320801).

REFERENCES

[1] Alliez, P.; Meyer, M.; Desbrun, M.: Interactive geometry remeshing, in: Proc. of SIGGRAPH, 2002, 347-354. DOI: 10.1145/566570.566588

[2] Alliez, P.; Ucelli, G.; Gotsman, C.; Attene, M.: Recent advances in remeshing of surfaces, State-of-the-art report of the AIM@SHAPE EU network, 2005.

[3] Attene, M.; Facidieno, B.; Spagnuolo, M.; Rossignac, J.: Edge-sharpener: Recovering sharp features in triangulations of non-adaptively re-meshed surfaces, in: Proc. of the Eurographics/ACM SIGGRAPH symposium on Geometry processing, 2003, 62-71.

[4] Cignoni, P.; Scopigno, R.; Tarini, M.: A simple normal enhancement technique for interactive non-photorealistic renderings, Computers and Graphics, 29(1), 2005, 125-133. DOI: 10.1016/j.cag.2004.11.012

[5] Eigensatz, M.; Sumner, R.W.; Pauly, M.: Curvature-domain shape processing, Comput. Graph. Forum, 27(2), 2008, 241-250. DOI: 10.1145/1477926.1477937
[6] Garland, M.; Heckbert, P.: Surface simplification using quadric error metrics, in: Proc. of SIGGRAPH, 1997, 209-216. DOI: 10.1145/258734.258849
[7] Guskov, I.; Sweldens, W.; SchrÄoder, P.: Multiresolution signal processing for meshes, in: Proc. of SIGGRAPH, 1999, 325-334. DOI: 10.1145/311535.311577
[8] Hoppe, H.; DeRose, T,; Duchamp, T,; McDonald, J.; Stuetzle, W.: Mesh optimization, in: Proc. of SIGGRAPH, 1993, 19-26. DOI: 10.1145/166117.166119
[9] Jones, T.; Durand, F.; Desbrun, M.: Non-iterative, feature preserving mesh smoothing, in: Proc. of SIGGRAPH, 2003, 943-949. DOI: 10.1145/1201775.882367
[10] Judd, T.; Durand, F.; Adelson, E.: Apparent ridges for line drawing, in: Proc. of SIGGRAPH, 2007. DOI: 10.1145/1275808.1276401
[11] Lai, Y.; Zhou, Q.; Hu, S.; Wallner, J.; Pottmann, H.: Robust feature classification and editing, IEEE Trans. Vis. Comput. Graph, 13 (1), 2007, 34-45. DOI: 10.1109/TVCG.2007.19
[12] Lee, Y.; Markosian, L.; Lee, S.; J. Hughes, J.: Line drawings via abstracted shading, in: Proc. of SIGGRAPH, 2007. DOI: 10.1145/1275808.1276400
[13] Liu, L.; Tai, C.-L.; Ji, Z.; Wang, G.: Non-iterative approach for global mesh optimization, Computer-Aided Design, 39 (9), 2007, 772-782. DOI: 10.1016/j.cad.2007.03.004
[14] Nealen, A.; Igarashi, T.; Sorkine, O.; Alexa, M.: Laplacian mesh optimization, in: Proc. of ACM GRAPHITE, 2006, 381-389. DOI:10.1145/1174429.1174494
[15] Pebay, P.; Baker, T.: Analysis of triangle quality measures, Math. Comput. 72 (244), 2003, 1817-1839. DOI: 10.1090/S0025-5718-03-01485-6
[16] Pottmann, H.; Hofer, M.: Geometry of the squared distance function to curves and surfaces, Visualization and Mathematics III, 2003, 223-244.
[17] Ritschel, T.; Smith, K.; Ihrke, M.; Grosch, T.; Myszkowski, K.; Seidel, H.; 3D unsharp masking for scene coherent enhancement, ACM Trans. Graph, 27 (3). DOI: 10.1145/1360612.1360689
[18] Rusinkiewicz, S.; Burns, M.; DeCarlo, D.: Exaggerated shading for depicting shape and detail, ACM Transactions on Graphics, 25 (3), 2006, 1199-1205. DOI: 10.1145/1141911.1142015
[19] Rusinkiewicz, S.: Estimating curvatures and their derivatives on triangle meshes, in: Symposium on 3D Data Processing, Visualization, and Transmission, 2004, 486-493. DOI: 10.1109/3DPVT.2004.54
[20] Toledo, S.: Taucs: A library of sparse linear solvers, http://www.tau.ac.il/stoledo/taucs, 2003.
[21] Turk, G.: Re-tiling polygonal surfaces, in: Proc. of SIGGRAPH, 1992, 55-64. DOI: 10.1145/133994.134008
[22] Vorsatz, J.; Rossl, C.; Kobbelt, L.; Seidel, H.: Feature sensitive remeshing, in: Proc. of EUROGRAPHICS, 2001, pp. 393-401. DOI: 10.1111/1467-8659.00532
[23] Vorsatz, J.; Rossl, C.; Seidel, H.; Dynamic remeshing and applications, in: Proc. of symposium on Solid modeling and applications, 2003, 167-175. DOI: 10.1145/781606.781633
[24] Yagou, H.; Belyaevy, A.; Weiz, D.: High-boost mesh filtering for 3D shape enhancement, Journal of Three Dimensional Images, 17 (1), 2003, 170-175.

7 APPENDIX. PROOF OF A PROPERTY OF THE EDGE METRIC IN SECTION 3.2

Let $e = v_1 v_2$ be an edge of the mesh M, with the end points v_1 and v_2 assumed to be on the model surface S. We are going to show that $f_{em}(e) \approx 8\delta$, where δ is the maximum error between the model surface S and the edge e. Let the vector $v_{12} = v_2 - v_1$ has its tail at $v_1 \in S$. Denote $\ell = \|v_{12}\|$, then $v_{12} = \ell \overline{v}_{12}$, $\overline{v}_{12}$ is a unit vector. On the other hand, we project v_{12} into the tangent plane of S at v_1 along the normal direction of S at v_1 to obtain a tangent vector $\tilde{v}_{12}$. Denote $T = \tilde{v}_{12} / \|\tilde{v}_{12}\|$, which is a unit tangent vector of S at v_1. It is easy to show that $T \approx \overline{v}_{12}$, as a first order approximation. Let T_1 and T_2 be unit vectors in the principal curvature directions of the surface S at v_1. Let k_1 and k_2 be the principal curvatures at v_1 (in the directions of T_1 and T_2). Let k be the normal curvature of S at v_1 in

the direction of the tangent vector T. Let $\theta = \arccos(T \cdot T_1)$, i.e. the angle between the edge vector v_{12} and the principal curvature direction T_1. Suppose $k_1 k_2 \geq 0$. Without loss of generality, assume that $k_1 \geq 0$ and $k_2 \geq 0$. Then

$$\begin{aligned} f_{v_1} &= k_1 (v_{12} \cdot T_1)^2 + k_2 (v_{12} \cdot T_2)^2 \\ &= \ell^2 [k_1 (\bar{v}_{12} \cdot T_1)^2 + k_2 (\bar{v}_{12} \cdot T_2)^2] \\ &\approx \ell^2 [k_1 (T \cdot T_1)^2 + k_2 (T \cdot T_2)^2] \\ &= \ell^2 [k_1 \cos^2 \theta + k_2 \sin^2 \theta] \\ &= \ell^2 k \end{aligned}$$

where the last equality follows from Euler's formula.

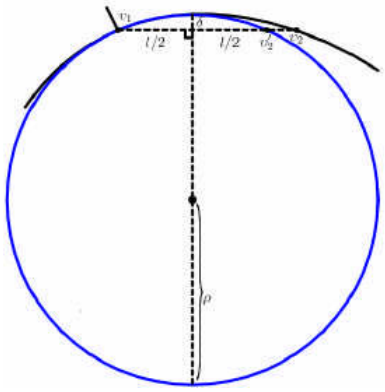

Fig. 13: The osculating circle of the sectional curve of the surface.

For the moment, assume $k > 0$. Let $\rho = 1/k$ be the normal curvature radius in the direction of v_{12}. Consider the osculating circle C of S in the normal section in the direction of T (see Fig. 13). Let v_2' be the second intersection of the edge $e = v_1 v_2$ (or its extended line) with the circle C. Let $\bar{\delta}$ be the error between the minor arc of the circle C and the line segment $v_1 v_2'$. As a second order approximation, we have $\bar{\delta} = \delta$, the maximum error between the edge $e = v_1 v_2$ and the surface S. Obviously,

$$(\ell/2)^2 = \bar{\delta}(2\rho - \bar{\delta}) = 2\bar{\delta}\rho - \bar{\delta}^2 = 2\bar{\delta}/k - \bar{\delta}^2$$

Dropping the second-order term, we obtain

$$\ell^2 k \approx 8\bar{\delta} \approx 8\delta$$

It follows that

$$f_{v_1}(v_{12}) = \ell^2 k \approx 8\delta.$$

Obviously, this approximation holds trivially when $k = 0$. Similarly, we can show

$$f_{v_2}(v_{21}) \approx 8\delta.$$

Hence,

$$f_{em}(e) = \frac{1}{2}(f_{v_1}(v_{12}) + f_{v_2}(v_{21})) \approx 8\delta.$$

as a first-order approximation.

When $k_1 k_2 < 0$, it can be shown by the same argument above that

$$\left| k_1 (v_{12} \cdot T_1)^2 + k_2 (v_{12} \cdot T_2)^2 \right| \approx 8\delta.$$

However, in this case, because of the need to maintain $f_{v_1}(v_{12})$ as a positive semi-definite quadratic form, we do not have an equality but only

$$f_{v_1}(v_{12}) \geq \left| k_1 (v_{12} \cdot T_1)^2 + k_2 (v_{12} \cdot T_2)^2 \right| \approx 8\delta.$$

That is, $f_{em}(e)$ is an over-estimate of the error δ if $k_1 k_2 < 0$.