Variational reconstruction using subdivision surfaces with continuous sharpness control

Xiaoqun Wu¹ (✉), Jianmin Zheng², Yiyu Cai², and Haisheng Li¹

© The Author(s) 2017. This article is published with open access at Springerlink.com

Abstract We present a variational method for subdivision surface reconstruction from a noisy dense mesh. A new set of subdivision rules with continuous sharpness control is introduced into Loop subdivision for better modeling subdivision surface features such as semi-sharp creases, creases, and corners. The key idea is to assign a sharpness value to each edge of the control mesh to continuously control the surface features. Based on the new subdivision rules, a variational model with L_1 norm is formulated to find the control mesh and the corresponding sharpness values of the subdivision surface that best fits the input mesh. An iterative solver based on the augmented Lagrangian method and particle swarm optimization is used to solve the resulting non-linear, non-differentiable optimization problem. Our experimental results show that our method can handle meshes well with sharp/semi-sharp features and noise.

Keywords variational model; subdivision surface; sharpness; surface reconstruction; L_1 norm

1 Introduction

Subdivision surfaces have many nice properties such as compactness, smoothness, arbitrary topology, and local control. Various subdivision schemes have been developed. Some of them have been

widely used in animation and the entertainment industry. For example, subdivision surfaces are included in the MPEG4 standard [1]. While most subdivision schemes are designed to create surfaces smooth everywhere, some special sharp subdivision rules have been introduced in order to allow modeling of sharp features such as corners, darts, and creases. Hoppe et al. [2] introduced a set of special subdivision rules to define a piecewise smooth subdivision surface. Noticing that in the real world objects are rarely perfectly sharp, DeRose et al. [3] introduced the notion of a semi-sharp feature and proposed performing a few steps of geometry preserving subdivision and then switching to classic smooth Catmull–Clark subdivision. An integer was used to determine how many iterations of geometry preserving subdivision were performed, which served to control the sharpness of the shape.

Research has been conducted into using subdivision surfaces for reconstruction. It has been observed that using smooth subdivision surfaces to reconstruct shapes containing sharp features may not recover these features very well. Therefore Hoppe et al. [2] proposed to use their piecewise smooth subdivision surfaces to fit models with sharp features, which achieved very convincing reconstruction results. However, their method has two drawbacks. Firstly, the method needs to detect the sharp features before fitting, and so the reconstruction results depend on the correctness of the pre-detected sharp features. Secondly, in Hoppe et al.'s subdivision scheme, the classification of edges is binary, as sharp or smooth. This is not very effective for modeling semi-sharp features. Although DeRose et al.'s approach [3] provides a simple way to model semi-sharp features, and has

1 Beijing Key Lab of Big Data Technology for Food Safety, School of Computer and Information Engineering, Beijing Technology and Business University, China. E-mail: X. Wu, xiaoqunwu@gmail.com (✉); H. Li, lihsh@th.btbu.edu.cn.

2 College of Engineering, Nanyang Technological University, Singapore. E-mail: J. Zheng, asjmzheng@ntu.edu.sg; Y. Cai, myycail@ntu.edu.sg.

Manuscript received: 2017-02-27; accepted: 2017-04-27

been extended from the cubic case [4] to arbitrary degree for semi-sharp creases in Ref. [5], these methods of controlling the iterations of geometry preserving subdivision are inconvenient for reverse engineering where such sharpness control must be determined. Ideally, such a sharpness parameter should be continuously controllable.

Unlike prior art, our idea is to use real numbers to control the sharpness of geometric features, treating both the sharpness parameters and the control points as unknowns and solving for them within a variational framework. In this paper, we first extend the Loop subdivision scheme and Hoppe et al.'s sharp subdivision rules by introducing real numbers as sharpness parameters to continuously control the geometric features of surfaces. Using the proposed subdivision scheme, we present a variational model for feature-aware subdivision surface reconstruction from dense noisy mesh models. This model uses the L_1 norm to handle different types of noise and outliers in a consistent way. The optimal values of the sharpness parameters and control points are obtained by solving a minimization problem. As a result, our algorithm can automatically recover sharp and semi-sharp features in the input data. Overall, the contribution of the paper includes:

- a new subdivision scheme with continuous sharpness control, which can conveniently model both sharp and semi-sharp features, and
- a variational, feature-aware subdivision reconstruction algorithm, which is robust to different types of noise and works for a variety of geometric features.

The rest of the paper is organized as follows. Section 2 reviews related work. Section 3 introduces a new set of subdivision rules which provide continuous sharpness control. Section 4 presents our variational surface reconstruction. Experimental results are reported in Section 5, and Section 6 concludes the paper.

2 Related work

Subdivision surfaces are currently one of the most powerful surface representations used to model smooth shapes. Compared to regular surface splines such as NURBS which need multiple patches to handle shapes with complex topology, subdivision

surfaces cover a range of representations from “pure” patches to “pure” meshes, and can easily handle shapes of arbitrary topology. Also, unlike polygonal meshes [6], subdivision surfaces generate smooth surfaces, which is important in designing aesthetically pleasing shapes.

The first two subdivision surface schemes for arbitrary topology meshes were introduced in 1978 by Catmull and Clark [7], and Doo and Sabin [8]. They generalized tensor product B-splines of bidegree three and two respectively to arbitrary topologies, by extending the refinement rules to irregular parts of the control meshes. Loop subdivision was introduced for triangular meshes by Charles Loop [9] in 1987. Since then, many other subdivision schemes have been proposed. Examples include quad/triangle subdivision [10], mid-edge subdivision [11, 12], and $\sqrt{3}$ subdivision [13]. Stam [14] presented a method for exact evaluation of Catmull–Clark and Loop subdivision surfaces for arbitrary parameter values. Non-uniform Catmull–Clark and Doo–Sabin surfaces that generalize non-uniform tensor product B-spline surfaces to arbitrary topologies were proposed in Ref. [15]. Recently, a new matrix weighted rational subdivision surface scheme, an extension of traditional subdivision schemes, was proposed in Ref. [16]. In Ref. [17], surface subdivision and geometric partial differential equation methods were combined for freeform surface design. A matrix-based approach to modeling creases on arbitrary-degree subdivision curves was introduced in Ref. [18].

Subdivision surface reconstruction has been investigated. Most methods use smooth subdivision rules during surface fitting, which however has difficulty in capturing geometric detail. Lee et al. [19] proposed a new surface representation called the displaced subdivision surface, which uses a subdivision surface as a smooth domain, and a displacement function over this domain for detail. Panozzo et al. [20] presented an automatic method for construction of quad-based subdivision surfaces using fitmaps. To reconstruct the surface detail, Ref. [21] starts from an interactively specified initial control mesh and adaptively recovers the detail by inserting new vertices. The “MeshToSS” tool was introduced in Ref. [22] to convert a dense triangular mesh to progressive subdivision surfaces.

Litke et al. [23] proposed a new subdivision surface fitting algorithm with prescribed tolerance based on a method of “quasi-interpolation”. To preserve surface detail, the natural ridge-joined connectivity of umbilics and ridge-crossing was used to provide connectivity of the control mesh for subdivision in Ref. [24]. Some methods attempt to improve foot point finding and distance computation near features: different error terms such as point distance error, tangent distance error, and squared distance error are applied to improve efficiency and accuracy. A subdivision surface fitting method based on parameter correction was developed in Ref. [25]; it uses a combination of point error and tangent error for efficiency. Squared distance minimization has also been used in subdivision fitting [26]. This method considers the geometric error based on curvature to measure the distance from the subdivision surface to the target.

Another direction of research in subdivision surfaces focuses on designing special rules for sharp features. For example, to fit subdivision surfaces with sharp features, Ref. [2] proposed a new set of subdivision rules to model tangent plane discontinuities. In Ref. [27], a direct approach was proposed for constructing a subdivision surface from an irregular and dense triangle mesh of arbitrary topology. A new exact evaluation method for all types of sharp features was given in Ref. [28]. Lavoué and Dupont [29] generalized sharp creases to semi-sharp creases by setting the number of iterations of geometry preserving subdivision to control the sharpness of Catmull–Clark subdivision surfaces [3]. A semi-sharp subdivision surface fitting approach based on feature line approximation was developed; semi-sharp features were achieved by relaxation according to the curvature of the target surface. All these methods need to detect the sharp features first. The accuracy of feature detection affects the sharp features of the final reconstructed subdivision surface.

3 Loop subdivision with continuous sharpness control

Loop subdivision is a popular subdivision scheme for triangular meshes [9]. The refinement step proceeds by splitting each triangular face into four sub-triangles. The subdivision masks for generating new

edge points and updating old vertex points are given in Fig. 1; $a(n) = [40 - (3 + 2 \cos(2\pi/n))^2]/64$ where n is the vertex valence.

In order to introduce sharp features into the Loop subdivision surfaces, new subdivision rules were proposed in Ref. [2], in which tangent plane continuity across sharp edges was relaxed. The masks for a crease edge, a dart, and a corner are given in Fig. 2.

Inspired by these works, we introduce a new subdivision scheme with continuous sharpness control. In particular, each edge e in the control mesh is assigned a real number $h_e \in [0, 1]$ called the sharpness. When $h_e = 0$, subdivision generates a smooth edge just as in classic Loop subdivision. When $h_e = 1$, it generates a crease edge just as in Ref. [2]. When $0 < h_e < 1$, it generates a blend between a smooth edge and a crease, providing continuous control of sharpness. To do so, the new edge point e^v is computed by

$$e^v = (1 - h_e)e_{\text{smooth}}^v + h_e e_{\text{crease}}^v$$

where e_{smooth}^v and e_{crease}^v are respectively the edge points generated by the Loop subdivision rule and the crease rule proposed in Ref. [2].

For a vertex v , let n_v denote the number of incident edges with $h_e > 0$. Depending on n_v , the rule for computing the new vertex point falls into the

Fig. 1 Loop subdivision masks. Left: vertex mask. Right: edge mask.

Fig. 2 Sharp feature masks. Blue lines are sharp edges. The red rectangle is the new inserted vertex.

following different cases:

- When $n_v < 2$, the vertex is considered to be smooth ($n_v = 0$) or a dart ($n_v = 1$). The new vertex point is simply computed using the smooth vertex mask.
- When $n_v = 2$, the vertex is considered to be a crease vertex. If the two incident edges with nonzero sharpness are e_j and e_k , we define the sharpness of the crease vertex to be $s_v = (h_{e_j} + h_{e_k})/2$. Then

$$v_{\text{crease}}^{\text{new}} = (1 - s_v)v_{\text{smooth}} + s_v v_{\text{crease}}$$

where v_{smooth} and v_{crease} are the vertex points generated by the smooth and crease rules, respectively.

- When $n_v > 2$, the vertex is considered to be a corner. Let A_s the set of incident edges with nonzero sharpness. Then the sharpness of the vertex is defined as $s_v = \sum_{e_i \in A_s} h_{e_i}/n_v$ and the new vertex point is computed by

$$v_{\text{corner}}^{\text{new}} = (1 - s_v)\bar{v}_{\text{crease}}^{\text{new}} + s_v v_{\text{corner}}$$

where $\bar{v}_{\text{crease}}^{\text{new}}$ is the weighted average of the crease vertices and v_{corner} is the corner vertex generated by the masks in Fig. 2.

After one refinement step, each old edge is replaced by two new edges. The sharpness of the new edges is inherited from the old edge.

Following Ref. [3], as well as using edge sharpness during the subdivision process, we also use a global integer k to control the number of iterations of refinement. After k iterations, the subdivision rules are switched to the modified Loop subdivision rules [2]. The advantage of this approach is that the nice property of the modified Loop subdivision surface is guaranteed mathematically while visually semi-sharp features are retained.

Using this feature-aware subdivision scheme, we can conveniently create fillets with various curvatures. Figure 3 shows a simple example where different shapes are generated from the same initial control mesh but have different values of edge sharpness.

4 Variational subdivision surface reconstruction

4.1 Problem

Subdivision surface reconstruction aims to construct a subdivision surface to recover the underlying surface, including its geometric features, from which a set of input points has been sampled. This is actually a very challenging problem, for several reasons. Firstly, the problem by its nature is an inverse problem. Both geometry and topology of the underlying surface are unknown. Secondly, to define a subdivision surface, we have to specify the control points and their connectivity. Their determination involves two different types of problems. The former is a continuous optimization problem, while the latter is a combinatorial problem. Thirdly, the underlying surface may contain sharp or semi-sharp features. Automatically incorporating those features in the optimization process is nontrivial. Fourthly, the input data usually contains noise and outliers. The co-existence of noise, outliers, and sharp features makes the problem even more difficult.

4.2 Solution overview

We now present an engineering solution to the problem. The pipeline is shown in Fig. 4; it consists of the four steps outlined below.

Fig. 3 Examples of different values of edge sharpness. The control mesh is drawn as a wireframe. Edges with different sharpness are drawn in different colors: gray, yellow, magenta, and red correspond to $h_e = 0, 0.6, 0.9, 1$, respectively.

Fig. 4 Pipeline of our approach.

1. If the input is a set of discrete points without connectivity information, we use some existing software package to convert them into a triangular mesh. If the input data is already a triangular mesh, this step is not needed.
2. Let the input mesh or generated mesh be S^0 , with point data $\mathbf{v} = \{v_1, \dots, v_n\}$. It may contain noise and outliers. This step removes the noise and outliers of S^0 while keeping the geometric features. The results is a smoothed mesh S_s . Existing methods such as bilateral mesh denoising can be used for this purpose.
3. The denoised triangular mesh is simplified to a coarser one while preserving the sharp features. A feature preserving variant of the QEM method [30] is used to generate the coarse mesh. We denote the coarse mesh by S_c^0 and assume that S_c^0 has point set $\mathbf{p} = \{p_1, \dots, p_m\}$.
4. We take S_c^0 as the initial control mesh for the subdivision surface to be constructed by the new subdivision rules given in Section 3. We fix the connectivity of S_c^0 and optimize the positions of the control points and the edge sharpnesses to achieve the best fit to mesh S^0 .

In the rest of this section, we focus on Step 4, which is the key component of surface reconstruction. Its approach can also be incorporated into existing subdivision surface reconstruction algorithms to enhance their performance.

4.3 Variational model

The control points $\mathbf{p}$ and edge sharpnesses $\mathbf{h} = \{h_1, \dots, h_{n_e}\}$, where n_e is the number of control mesh edges, are sought so as to satisfy:

$$\min_{\mathbf{p}, \mathbf{h}} E(\mathbf{p}, \mathbf{h}) = E_s(\mathbf{p}) + \lambda E_f(\mathbf{p}, \mathbf{h})$$

where the energy function includes two terms: a smoothing term and an error (fidelity) term,

balanced by the fidelity parameter λ . As the values of the sharpnesses $\mathbf{h}$ are determined by optimization, our method does not depend on feature detection.

4.3.1 Smoothing term

To prevent the control mesh from self-intersection and to avoid the influence of noise, we introduce the following smoothing term:

$$E_s(\mathbf{p}) = \sum_{i=1}^m \|L(p_i)\| = \mathbf{L}\mathbf{p}$$

where $L(p_i)$ is the discrete Laplacian at point p_i , $\mathbf{L}$ is the Laplacian matrix, and $\|\cdot\|$ is the L_1 norm. We call $\sum_{i=1}^m \|L(p_i)\|$ the total Laplacian of $\mathbf{p}$. The L_1 norm is used instead of L_2 to discourage smoothing of creases and corners, as shown in Fig. 5.

4.3.2 Error term

To compute the error, we may project the data points q_i onto the subdivision surface S . This is complicated in practice as the surface S is defined as the limit of a set of refined mesh sequences using our feature-aware subdivision scheme. Hence, we use a piecewise linear approximation $\tilde{S}$ for S instead. $\tilde{S}$ is obtained by subdividing the initial control mesh k times to produce a refined mesh S^k using our feature-aware subdivision masks, and then pulling all the vertices of S^k to the limit positions using the limit position masks of Ref. [2]. In this way, $\tilde{S}$ can be written as an affine combination of the initial positions $\mathbf{p}$.

To make our fitting algorithm independent of the results of the smoothing process, we use the points of the input noisy mesh S^0 instead of the smoothed clean mesh obtained from Step 2. To handle the possible existence of different types of noise, we use the L_1 norm to measure the distance in this error term. For each data point $v_i \in S^0$, let the closest point on the approximated surface $\tilde{S}$ be w_i . It can be represented by a linear combination of vertices of

Fig. 5 Smoothing alternatives. Left: use of L_2 norm. Right: use of total Laplacian. Close-up views are shown at lower right.

$\tilde{S}$, so the error term can be represented as

$$E_f(\mathbf{p}, \mathbf{h}) = \|\mathbf{v} - M(\mathbf{h})\mathbf{p}\|$$

where $M(\mathbf{h})$ is a matrix depending on $\mathbf{h}$. For simplicity, we denote $M(\mathbf{h})$ by $\mathbf{M}$. $\mathbf{Mp}$ is the foot point vector of $\mathbf{v}$ on $\tilde{S}$.

4.4 Numerical solver

Combining both terms gives our variational optimization model:

$$\min_{\mathbf{p}, \mathbf{h}} \{\|\mathbf{Lp}\| + \lambda\|\mathbf{v} - \mathbf{Mp}\|\}$$

$\mathbf{p}$ and $\mathbf{h}$ are values to be determined.

This variational optimization problem is highly non-linear, and also non-differentiable due to the use of the L_1 norm. To solve it, we let $\mathbf{q} = \mathbf{Lp}$ and $\mathbf{z} = \mathbf{v} - \mathbf{Mp}$. The optimization model becomes a constrained problem:

$$\begin{aligned} \min_{\mathbf{p}, \mathbf{h}} \quad & \|\mathbf{q}\| + \lambda\|\mathbf{z}\| \\ \text{subject to} \quad & \mathbf{q} = \mathbf{Lp}, \quad \mathbf{z} = \mathbf{v} - \mathbf{Mp} \end{aligned}$$

We construct the following augmented Lagrangian functional:

$$\begin{aligned} L(\mathbf{p}, \mathbf{h}, \mathbf{q}, \mathbf{z}; \lambda_q, \lambda_z) = & \|\mathbf{q}\| + \lambda\|\mathbf{z}\| + \langle \lambda_q, \mathbf{q} - \mathbf{Lp} \rangle \\ & + \frac{a_q}{2} \|\mathbf{q} - \mathbf{Lp}\|^2 + \langle \lambda_z, \mathbf{z} - (\mathbf{v} - \mathbf{Mp}) \rangle \\ & + \frac{a_z}{2} \|\mathbf{z} - (\mathbf{v} - \mathbf{Mp})\|^2 \end{aligned}$$

where λ_q and λ_z are the Lagrangian multipliers, $\langle, \rangle$ is the vector dot product operator, and a_q and a_z are two positive numbers. We now seek a solution to the saddle-point problem:

$$\min_{\mathbf{p}, \mathbf{h}, \mathbf{q}, \mathbf{z}} \max_{\lambda_q, \lambda_z} L(\mathbf{p}, \mathbf{h}, \mathbf{q}, \mathbf{z}; \lambda_q, \lambda_z)$$

by using Algorithm 1. Algorithm 1 involves solving several sub-problems, whose details are now described.

Initialization of $\mathbf{h}$. The initial sharpness is set based on coarse detection of sharp features. Specifically, for each edge e of the initial control mesh S_c^0 , we use a threshold θ_0 to detect whether e is a candidate crease edge. Let θ_e be the angle between the normals of the two faces containing e . If $\theta_e > \theta_0$, $h_e = 1$; otherwise, $h_e = 0$.

$\mathbf{p}$ subproblem. Fixing $\mathbf{h}, \mathbf{q}, \mathbf{z}$, we solve the following quadratic equation:

$$\begin{aligned} \min_{\mathbf{p}} \quad & \langle \lambda_q, \mathbf{q} - \mathbf{Lp} \rangle + \frac{a_q}{2} \|\mathbf{q} - \mathbf{Lp}\|^2 \\ & + \langle \lambda_z, \mathbf{z} - (\mathbf{v} - \mathbf{Mp}) \rangle + \frac{a_z}{2} \|\mathbf{z} - (\mathbf{v} - \mathbf{Mp})\|^2 \end{aligned}$$

This can be converted into a linear system.

Algorithm 1 Algorithm for subdivision surface fitting

The specified error threshold is ϵ_0 .

Initialize $\mathbf{h}^{(0)}$ by preprocessing and $\mathbf{p}^{(0)}$ using S_c^0 ;

$\mathbf{z}^{(0)} = \mathbf{q}^{(0)} = 0$;

$\lambda_q^{(0)} = \lambda_z^{(0)} = 0$;

$l = 0, \epsilon = \infty$;

while $\epsilon > \epsilon_0$ and $l < \text{max-iterations}$ **do**

    Compute $(\mathbf{p}^{(l+1)}, \mathbf{h}^{(l+1)}, \mathbf{q}^{(l+1)}, \mathbf{z}^{(l+1)})$ as an approximate minimizer of the augmented Lagrangian functional with the Lagrangian multipliers $\lambda_q^{(l)}, \lambda_z^{(l)}$

$$(\mathbf{p}^{(l+1)}, \mathbf{h}^{(l+1)}, \mathbf{q}^{(l+1)}, \mathbf{z}^{(l+1)}) \approx$$

$$\arg \min_{\mathbf{p}, \mathbf{h}, \mathbf{q}, \mathbf{z}} L(\mathbf{p}, \mathbf{h}, \mathbf{q}, \mathbf{z}; \lambda_q^{(l)}, \lambda_z^{(l)})$$

    using the following iterations:

$$\mathbf{p}^{(l+1)} = \mathbf{p}^{(l)}, \mathbf{h}^{(l+1)} = \mathbf{h}^{(l)}, \mathbf{q}^{(l+1)} = \mathbf{q}^{(l)}, \mathbf{z}^{(l+1)} = \mathbf{z}^{(l)}$$

for $ll = 1, \dots, K$ **do**

        compute

$$\mathbf{p}^{(l+1)} = \arg \min_{\mathbf{p}} L(\mathbf{p}, \mathbf{h}^{(l+1)}, \mathbf{q}^{(l+1)}, \mathbf{z}^{(l+1)}; \lambda_q^{(l)}, \lambda_z^{(l)})$$

$$\mathbf{h}^{(l+1)} = \arg \min_{\mathbf{h}} L(\mathbf{p}^{(l+1)}, \mathbf{h}, \mathbf{q}^{(l+1)}, \mathbf{z}^{(l+1)}; \lambda_q^{(l)}, \lambda_z^{(l)})$$

$$\mathbf{q}^{(l+1)} = \arg \min_{\mathbf{q}} L(\mathbf{p}^{(l+1)}, \mathbf{h}^{(l+1)}, \mathbf{q}, \mathbf{z}^{(l+1)}; \lambda_q^{(l)}, \lambda_z^{(l)})$$

$$\mathbf{z}^{(l+1)} = \arg \min_{\mathbf{z}} L(\mathbf{p}^{(l+1)}, \mathbf{h}^{(l+1)}, \mathbf{q}^{(l+1)}, \mathbf{z}; \lambda_q^{(l)}, \lambda_z^{(l)})$$

end for

    Now update the Lagrangian multipliers:

$$\lambda_q^{(l+1)} = \lambda_q^{(l)} + a_q(\mathbf{q} - \mathbf{Lp})$$

$$\lambda_z^{(l+1)} = \lambda_z^{(l)} + a_z(\mathbf{z} - (\mathbf{v} - \mathbf{Mp}))$$

$$\epsilon = \|\mathbf{p}^{(l+1)} - \mathbf{p}^{(l)}\|^2;$$

$l++$;

end while

$\mathbf{h}$ subproblem. Fixing $\mathbf{p}, \mathbf{q}, \mathbf{z}$, we find the sharpness $\mathbf{h}$ which only affects the subdivision rule. Hence, the $\mathbf{h}$ -sub problem is

$$\min_{\mathbf{h}} \langle \lambda_z, \mathbf{z} - (\mathbf{v} - M(\mathbf{h})\mathbf{p}) \rangle + \frac{a_z}{2} \|\mathbf{z} - (\mathbf{v} - M(\mathbf{h})\mathbf{p})\|^2$$

To solve this minimization problem, we adopt particle swarm optimization (PSO) [31], an evolutionary procedure. It starts from many random initialization seeds. At each iteration a set of candidate solutions is sought, the score of each solution is calculated, and the solutions are updated in the search space by shifting them towards the best current solution.

$\mathbf{q}$ subproblem. Fixing $\mathbf{p}, \mathbf{h}, \mathbf{z}$, we find $\mathbf{q}$ by solving:

$$\min_{\mathbf{q}} \{\|\mathbf{q}\| + \langle \lambda_q, \mathbf{q} - \mathbf{Lp} \rangle + \frac{a_q}{2} \|\mathbf{q} - \mathbf{Lp}\|^2\}$$

After reformulation, we have

$$\min_{\mathbf{q}} \left\{ \sum_{i=0}^m (\|q_i\| + \frac{a_q}{2} \|q_i - (\mathbf{Lp} - \frac{\lambda_q}{a_q})_i\|^2 + c) \right\}$$

This problem is decomposable and has the closed

form solution:

$$\mathbf{q} = \max\{0, 1 - \frac{1}{a_q \|\mathbf{w}_q\|}\} \mathbf{w}_q$$

where $\mathbf{w}_q = \mathbf{L}\mathbf{p} - \lambda_q/a_q$.

$\mathbf{z}$ subproblem. Fixing $\mathbf{p}, \mathbf{h}, \mathbf{q}$, we find $\mathbf{z}$ by solving:
 $\min_{\mathbf{z}} \{\lambda \|\mathbf{z}\| + \langle \lambda_{\mathbf{z}}, \mathbf{z} - (\mathbf{v} - \mathbf{M}\mathbf{p}) \rangle + \frac{a_z}{2} \|\mathbf{z} - (\mathbf{v} - \mathbf{M}\mathbf{p})\|^2\}$
 which also has a closed form solution:

$$\mathbf{z} = \max\{0, 1 - \frac{\lambda}{a_z \|\mathbf{w}_z\|}\} \mathbf{w}_z$$

where $\mathbf{w}_z = \mathbf{v} - \mathbf{M}\mathbf{p} - \lambda_{\mathbf{z}}/a_z$.

5 Experiments

This section reports our experimental results on a variety of models. Our implementation uses Intel's MKL sparse solver to solve the sparse linear system. The experiments were conducted on an

Intel Xeon E5-1650 CPU. Our method contains a few parameters. They were set empirically but consistently for all the examples. Specifically, let the average edge lengths of the initial control mesh and the input model be $\bar{l}_c$ and $\bar{l}_d$, respectively. Then $\lambda = \alpha m \bar{l}_c / (n \bar{l}_d)$, $\alpha = 100$, $a_q = 1$, and $a_z = 25$. Our method is not very sensitive to the value of the fidelity parameter λ due to the use of L_1 norm.

We first evaluate the effect of the sharpness in reconstruction. For this purpose, we fix the positions of the control mesh and only optimize the sharpness. Figure 5 shows an example, where the sharpness is obtained by solving the optimization problem. Edges whose sharpness is non-zero are marked in yellow. We can see that the reconstruction with the optimized sharpness is better; sharpness provides additional degrees of freedom for reconstruction.

Fig. 6 Rocker arm model: (a) noisy dense mesh, (b) mesh after denoising, (c) initial mesh obtained by simplification, (d) subdivision surface reconstructed by Loop subdivision, (e) subdivision surface using the rules in Ref. [2], and (f) reconstructed subdivision surface using our method. The bottom row shows close-up views of the subdivision surfaces in the row above.

We next present an example in Fig. 6 illustrating the whole pipeline of our method; it also provides a comparison with two other two methods. Two types of noise, outliers with variance $0.5\bar{l}_e$ and Gaussian noise with variance $0.1\bar{l}_e$, where $\bar{l}_e$ is the mean edge length of the input mesh, have been added to generate the noisy mesh shown in Fig. 6(a). Figure 6(b) is the mesh after denoising. Figure 6(c) is an initial mesh obtained by simplification from (b). Using this initial mesh, we find the optimal positions and sharpness to generate a subdivision surface shown in Fig. 6(f). Figure 6(d) is the subdivision surface reconstructed by classical Loop subdivision, which is much smoother and does not represent the high curvature areas well without increasing the density of the control mesh. Figure 6(e) is the subdivision surface reconstructed by the approach of Ref. [2], where crease edge needs to be detected first by computing the dihedral angles between the faces incident to edges. That approach can recover the sharp edges well if they are detected correctly, but may have difficulty in recovering semi-sharp edges. Since our method tunes the sharpness continuously, it can handle both sharp and semi-sharp features well.

Figure 7 shows two further examples: a coarse CAD-like model, and a graphics model with fine details. It can be seen that our method works consistently well for both types of 3D model.

Finally, two comparisons are given in Figs. 8 and 9, where the two input noisy meshes contain two types of noise. Figures 8(b)–8(e) and 9(b)–9(e) are the reconstructed surfaces generated by smooth rules [9], piecewise smooth rules [2], the approach of Ref. [25], and our method. Figures 8(g)–8(j) and 9(g)–9(j) visualize the distributions of errors between the ground truth models and the reconstructed surfaces respectively. It can be seen that both the smooth subdivision scheme and the sharp subdivision scheme [2] do not recover geometric features very well using the same number of control points as our approach. In particular, Fig. 8(c) shows that when sharp edges are detected incorrectly, the piecewise smooth subdivision rules of Ref. [2] may yield sharp edges near smooth areas of the original model. In Ref. [28], methods were proposed for exact evaluation of Loop subdivision surfaces with various types of sharp features. These methods are important for high quality surface fitting. To recover

Fig. 7 Reconstruction results. Left: input models. Above: Fandisk with 750 vertices. Below: head of David with 195,760 vertices. Right: reconstructed subdivision surfaces, using 152 and 1583 control points respectively.

sharp features in reconstruction, sharp features are detected and then the corresponding subdivision mask is used. Thus, correctness of the detection step is crucial. In order to capture surface features, the method of Ref. [25] needs to adaptively add control points to achieve the same level of accuracy as our method. Even so, some artifacts may still be observed near sharp features as shown in Fig. 8(d). Our method automatically controls the sharpness of the control mesh and works well for both sharp and semi-sharp features, without the need to increase the number of control points or accurately detect sharp edges visually and quantitatively.

Table 1 summarises the numbers of vertices, the numbers of control points (NCP), the maximum errors (ME), and the root mean errors (RME) for these examples. Our method produces the reconstruction results with the smallest errors. While the method of Ref. [25] can generate reconstruction results with comparable errors, it requires many more control points. Timings for our algorithm are reported in Table 2.

Fig. 8 Vase. (a) Input noisy model corrupted by Gaussian noise ($\sigma = 0.1 \bar{l}_e$) and 10% impulsive noise with scale $0.5 \bar{l}_e$, (b–e) subdivision surfaces reconstructed using Loop subdivision, the rules in Ref. [2], the method of Ref. [25], and our method, (f) ground truth model, and (g–j) error distributions for (b–e) respectively.

Fig. 9 Blade. (a) Input noisy model corrupted by Gaussian noise ($\sigma = 0.1 \bar{l}_e$) and 10% impulsive noise with scale $0.5 \bar{l}_e$, (b–e) subdivision surfaces reconstructed using Loop subdivision, the rules in Ref. [2], the method of Ref. [25], and our method, (f) ground truth model, and (g–j) error distributions for (b–e) respectively.

Table 1 Summary of models: number of vertices, number of control points (CP), maximum error (ME), and root mean error (RME)

Model		CP	ME	RME
Rocker arm (70,400 vertices)	(d)	275	0.0290	0.00545
	(e)	275	0.0191	0.00217
	(f)	275	0.0100	0.00104
Vase (64,000 vertices)	(b)	252	0.0543	0.00664
	(c)	252	0.0211	0.00235
	(d)	1120	0.0076	0.000476
	(e)	252	0.00757	0.000376
Blade (51,200 vertices)	(b)	202	0.0561	0.00842
	(c)	202	0.0158	0.00221
	(d)	502	0.00669	0.000868
	(e)	202	0.00668	0.000768

Table 2 Timings for our approach

Model	Time
Rocker arm	20.3 min
Vase	18.6 min
Blade	13.8 min

6 Conclusions and future work

We have described a feature-aware subdivision surface reconstruction algorithm. It contains three major technical components. Firstly, we introduce a set of new subdivision rules which allows us to continuously control the sharpness of the resulting subdivision surface shape. Secondly, a variational model based on the L_1 norm is presented to reconstruct subdivision surfaces from noisy dense meshes. Thirdly, a numerical solver based on the augmented Lagrangian approach is used to solve the proposed non-linear, non-differentiable optimization problem. Our experiments demonstrate that our algorithm works well, and can automatically reconstruct overall smooth shapes with sharp and semi-sharp features.

Our current approach has two limitations. Firstly, the connectivity of the mesh is fixed during the optimization process, and only the vertex positions and edge sharpnesses are optimized. Thus the approach may not work well for cases where none of the edges of the initial control mesh are aligned with the creases in the target model. Secondly, our approach needs about 10–20 minutes to handle models with 50k–70k vertices. Most of the computational time is taken in solving the sharpness optimization subproblem due to its nonlinearity.

In future we will investigate how to optimize the connectivity of the mesh and how to speed up the solver.

Acknowledgements

The main idea of this paper was presented in the Computational Visual Media Conference 2017. This research was partially supported by the National Natural Science Foundation of China (No. 61602015), an MOE AcRF Tier 1 Grant of Singapore (RG26/15), Beijing Natural Science Foundation (No. 4162019), open funding project of State Key Lab of Virtual Reality Technology and Systems at Beihang University (No. BUAA-VR-16KF-06), and the Research Foundation for Young Scholars of Beijing Technology and Business University.

References

- [1] MPEG-4. ISO-IEC 14496-16. Coding of audio-visual objects: Animation framework extension (AFX). Available at <http://mpeg.chiariglione.org/standards/mpeg-4/animation-framework-extension-afx>.
- [2] Hoppe, H.; DeRose, T.; Duchamp, T.; Halstead, M.; Jin, H.; McDonald, J.; Schweitzer, J.; Stuetzle, W. Piecewise smooth surface reconstruction. In: Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques, 295–302, 1994.
- [3] DeRose, T.; Kass, M.; Truon, T. Subdivision surfaces in character animation. In: Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques, 85–94, 1998.
- [4] Stam, J. On subdivision schemes generalizing uniform B-spline surfaces of arbitrary degree. *Computer Aided Geometric Design* Vol. 18, No. 5, 383–396, 2001.
- [5] Kosinka, J.; Sabin, M. A.; Dodgson, N. A. Semi-sharp creases on subdivision curves and surfaces. *Computer Graphics Forum* Vol. 33, No. 5, 217–226, 2014.
- [6] Zhang, L.; Liu, L.; Gotsman, C.; Huang, H. Mesh reconstruction by meshless denoising and parameterization. *Computer & Graphics* Vol. 34, No. 3, 198–208, 2010.
- [7] Catmull, E.; Clark, J. Recursively generated B-spline surfaces on arbitrary topological meshes. *Computer-Aided Design* Vol. 10, No. 6, 350–355, 1978.
- [8] Doo, D.; Sabin, M. Behaviour of recursive division surfaces near extraordinary points. *Computer-Aided Design* Vol. 10, No. 6, 356–360, 1978.
- [9] Loop, C. T. Smooth subdivision surfaces based on triangles. Master Thesis. University of Utah, 1987.
- [10] Stam, J.; Loop, C. Quad/triangle subdivision. *Computer Graphics Forum* Vol. 22, No. 1, 79–85, 2003.
- [11] Habib, A.; Warren, J. Edge and vertex insertion for a

- class of C^1 subdivision surfaces. *Computer Aided Geometric Design* Vol. 16, No. 4, 223–247, 1999.
- [12] Peters, J.; Reif, U. The simplest subdivision scheme for smoothing polyhedra. *ACM Transactions on Graphics* Vol. 16, No. 4, 420–431, 1997.
- [13] Kobbelt, L. $\sqrt{3}$ -subdivision. In: Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques, 103–112, 2000.
- [14] Stam, J. Exact evaluation of Catmull–Clark subdivision surfaces at arbitrary parameter values. In: Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques, 395–404, 1998.
- [15] Sederberg, T. W.; Zheng, J.; Sewell, D.; Sabin, M. Non-uniform recursive subdivision surfaces. In: Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques, 387–394, 1998.
- [16] Yang, X. Matrix weighted rational curves and surfaces. *Computer Aided Geometric Design* Vol. 42, 40–53, 2016.
- [17] Pan, Q.; Xu, G.; Zhang, Y. A unified method for hybrid subdivision surface design using geometric partial differential equations. *Computer-Aided Design* Vol. 46, 110–119, 2014.
- [18] Kosinka, J.; Sabin, M. A.; Dodgson, N. A. Creases and boundary conditions for subdivision curves. *Graphical Models* Vol. 76, No. 5, 240–251, 2014.
- [19] Lee, A.; Moreton, H.; Hoppe, H. Displaced subdivision surfaces. In: Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques, 85–94, 2000.
- [20] Panozzo, D.; Puppo, E.; Tarini, M.; Pietroni, N.; Cignoni, P. Automatic construction of quad-based subdivision surfaces using fitmaps. *IEEE Transactions on Visualization and Computer Graphics* Vol. 17, No. 10, 1510–1520, 2011.
- [21] Suzuki, H.; Takeuchi, S.; Kimura, F.; Kanai, T. Subdivision surface fitting to a range of points. In: Proceedings of the 7th Pacific Conference on Computer Graphics and Applications, 158–167, 1999.
- [22] Kanai, T. MeshToSS: Converting subdivision surfaces from dense meshes. In: Proceedings of the Vision Modeling and Visualization Conference, 325–332, 2001.
- [23] Litke, N.; Levin, A.; Schröder, P. Fitting subdivision surfaces. In: Proceedings of the Conference on Visualization, 319–324, 2001.
- [24] Ma, X.; Keates, S.; Jiang, Y.; Kosinka, J. Subdivision surface fitting to a dense mesh using ridges and umbilics. *Computer Aided Geometric Design* Vol. 32, 5–21, 2015.
- [25] Marinov, M.; Kobbelt, L. Optimization methods for scattered data approximation with subdivision surfaces. *Graphical Models* Vol. 67, No. 5, 452–473, 2005.
- [26] Cheng, K. S.; Wang, W.; Qin, H.; Wong, K. Y.; Yang, H.; Liu, Y. Design and analysis of optimization methods for subdivision surface fitting. *IEEE Transactions on Visualization and Computer Graphics* Vol. 13, No. 5, 878–890, 2007.
- [27] Ma, W.; Ma, X.; Tso, S.-K.; Pan, Z. A direct approach for subdivision surface fitting from a dense triangle mesh. *Computer-Aided Design* Vol. 36, No. 6, 525–536, 2004.
- [28] Ling, R.; Wang, W.; Yan, D. Fitting sharp features with Loop subdivision surfaces. *Computer Graphics Forum* Vol. 27, No. 5, 1383–1391, 2008.
- [29] Lavoué, G.; Dupont, F. Semi-sharp subdivision surface fitting based on feature lines approximation. *Computers & Graphics* Vol. 33, No. 2, 151–161, 2009.
- [30] Garland, M.; Heckbert, P. S. Surface simplification using quadric error metrics. In: Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques, 209–216, 1997.
- [31] Kennedy, J.; Eberhart, R. Particle swarm optimization. In: Proceedings of the IEEE International Conference on Neural Networks, Vol. 4, 1942–1948, 1995.

Xiaoqun Wu is now an assistant professor in the School of Computer and Information Engineering, Beijing Technology and Business University, China. She received her B.S. and M.S. degrees from Zhejiang University, China, in 2007 and 2009, respectively, and her Ph.D. degree from Nanyang Technological University, Singapore, in 2014. Her research focuses on digital geometry processing and applications.

Jianmin Zheng is an associate professor in the School of Computer Science and Engineering, Nanyang Technological University, Singapore. He received his B.S. and Ph.D. degrees from Zhejiang University, China. His recent research focuses on T-spline technologies, digital geometry processing, virtual reality, visualization, interactive digital media, and applications. He has published more than 150 technical papers in international conferences and journals. He was the conference co-chair of Geometric Modeling and Processing 2014 and has served on the program committees of many international conferences.

Yiyu Cai is an associate professor in the School of Mechanical and Aerospace Engineering, Nanyang Technological University, Singapore. He received his B.S. and M.S. degrees from Zhejiang University, China, and his Ph.D. degree from the National University of Singapore. His research interests include 3D-based design, simulation, serious games, virtual reality,

etc. He is also active in IDM application research for engineering, bio- and medical sciences, education, arts, etc.

Haisheng Li is a professor in the School of Computer and Information Engineering, Beijing Technology and Business University, China. He received his Ph.D. degree from Beihang University, China, in 2002. He is the director of the Network Center and the discipline leader in Computer

Science and Technology in Beijing Technology and Business University, China. His current research interests include computer graphics, scientific visualization, 3D model retrieval, etc.

Open Access The articles published in this journal are distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

Other papers from this open access journal are available free of charge from <http://www.springer.com/journal/41095>. To submit a manuscript, please go to <https://www.editorialmanager.com/cvmj>.