Available online at www.sciencedirect.com
SCIENCE DIRECT®
Computer Aided Geometric Design 23 (2006) 125–145
COMPUTER AIDED GEOMETRIC DESIGN
www.elsevier.com/locate/cagd

Discrete surface modelling using partial differential equations

Guoliang Xu [a,1], Qing Pan [a], Chandrajit L. Bajaj [b,*,2]

[a] *State Key Laboratory of Scientific and Engineering Computing, Institute of Computational Mathematics, Academy of Mathematics and System Sciences, Chinese Academy of Sciences, Beijing, 100080, China*
[b] *Center for Computational Visualization and Institute for Computational Engineering & Sciences, Department of Computer Science, University of Texas, Austin, TX 78712, USA*

Received 29 July 2004; received in revised form 15 May 2005; accepted 23 May 2005

Available online 5 July 2005

Abstract

We use various nonlinear partial differential equations to efficiently solve several surface modelling problems, including surface blending, N-sided hole filling and free-form surface fitting. The nonlinear equations used include two second order flows, two fourth order flows and two sixth order flows. These nonlinear equations are discretized based on discrete differential geometry operators. The proposed approach is simple, efficient and gives very desirable results, for a range of surface models, possibly having sharp creases and corners.

Keywords: Free-form surface fitting; Blending; N-sided hole; Differential equations

1. Introduction

We use various partial differential equations (PDE) to solve several surface modelling problems. The PDEs we use include the mean curvature flow, the averaged mean curvature flow, two fourth order (surface diffusion flow and quasi surface diffusion flow) and even higher order flows. All these equations are

* Corresponding author.
E-mail address: bajaj@cs.utexas.edu (C.L. Bajaj).

[1] Supported in part by Natural Science Foundation of China (10371130) and National Key Basic Research Project of China (2004CB318000).

[2] Supported in part by NSF grant ACI-9982297, NSF-ITR grants ACI-0220037 and EIA-03-25550, and a grant from NIH R01 GM074258-021.

0167-8396/$ – see front matter
doi:10.1016/j.cagd.2005.05.004

nonlinear and the geometry is intrinsic, i.e., the PDEs do not depend upon any particular parameterization. The problems we solve include surface blending, N-sided hole filling and free-form surface fitting with high order boundary continuity.

For the problems of surface blending and N-sided hole filling, we are given triangular surface meshes of the surrounding area. Triangular surface patches need to be constructed to fill the openings enclosed by the surrounding surface mesh and interpolate the hole boundary with some specified order of continuity. For the free-form surface fitting problem, we are possibly given a set of points, or a wire frame of curves that defines an outline of the desired shape, or even some surface patches. We construct a surface which interpolates the points or curves or the boundaries of the patches with specified order of continuity. The free-form surface fitting problem is the most general, including the surface blending and N-sided hole filling problems, as its special cases.

Our twofold strategy for solving these problems is as follows: First we construct an initial triangular surface mesh ("filler") using any of a number of automatic or semi-automatic free-form modelling techniques (see (Bajaj and Ihm, 1992; Bajaj et al., 1993; Greiner, 1994; Peters and Wittman, 1996; Xu et al., 2001)). One may also interactively edit this "filler" to meet the weak assumptions for an initial solution shape. This "filler" may be bumpy or noisy, and in general this "filler" does not satisfy the smoothness boundary conditions, though it may roughly characterize the shape of the surface to be constructed. Second we deform the initial mesh by solving a suitable flow PDE. Unlike most of the previous free-form modelling techniques, our approach solves high-order boundary continuity constraints without any prior estimation of normals or derivative jets along the boundary. The solution of the PDE is time dependent. We consider two possibilities for the time span of the evolution. One is a short time evolution, where we require the solution to respect to the initial shape or geometry (see Fig. 7). The other is a long time evolution, where the initial filler provides a topological structure, and what we look for is a stable solution state of the flow (see Figs. 1 and 4). In this paper, we focus our attention on these twofold solutions of PDEs with boundary continuity constraints, rather than the construction of initial filler mesh. In Section 3.4, we present automatic approaches for constructing the initial filler mesh, and our preferred choice.

Previous work. Earlier research on using PDEs to handle surface modelling problems trace back to Bloor et al.'s papers at the end of the 1980s (Bloor and Wilson, 1989, 1990). The basic idea of these

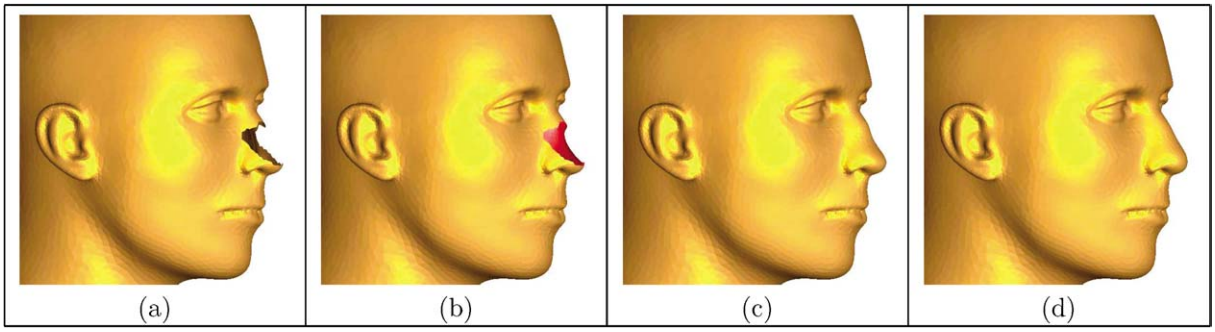

Fig. 1. (a) shows a head mesh with a hole around the nose. (b) shows an initial filler construction of the nose with a piece of minimal surface. (c) the filler surface, after 30 iteration, generated using fourth order flow ($k = 2$ in (2.9)) with time step size 0.0002. (d) the filler surface, after 20 iteration, generated using sixth order flow ($k = 3$ in (2.9)) with time step size 0.00002.

papers is the use of biharmonic equations on a rectangular domain to solve the blending and hole filling problems. One of the advantages of using the biharmonic equation is that it is linear, and therefore easier to solve. However, the equation is not geometry intrinsic and the solution of the equation (the geometry of the surface) depends on the concrete parameterization used. Furthermore, these methods are inappropriate to model surfaces with arbitrary shaped boundaries.

The evolution technique, based on the heat equation $\partial_t p - \Delta p = 0$, has been extensively used in the area of image processing (see (Preußer and Rumpf, 1999; Weickert, 1998). In (Weickert, 1998), there are 453 relevant references listed), where Δ is a $2D$ Laplace operator. This was extended lately to smoothing or fairing noisy surfaces (see (Clarenz et al., 2000; Desbrun et al., 1999; Meyer et al., 2002)). For a surface $\mathcal{M}$, the counterpart of the Laplacian Δ is the Laplace–Beltrami operator $\Delta_{\mathcal{M}}$ (see (do Carmo, 1992)). One then obtains the geometric diffusion equation $\partial_t p - \Delta_{\mathcal{M}} p = 0$ for a surface point $p(t)$ on the surface $\mathcal{M}(t)$. Taubin (1995) discussed the discretized operator of the Laplacian and related approaches in the context of generalized frequencies on meshes. Kobbelt (1996) considered discrete approximations of the Laplacian in the construction of fair interpolatory subdivision schemes. This work was extended in (Kobbelt et al., 1998) to arbitrary connectivity for purposes of multi-resolution interactive editing. Desbrun et al. (1999) used an implicit discretization of geometric diffusion to obtain a *strongly stable* numerical smoothing scheme. The same strategy of discretization is also adopted and analyzed by Deckelnick and Dziuk (2002) with the conclusion that this scheme is *unconditionally stable*. Clarenz et al. (2000) introduced anisotropic geometric diffusion to enhance features while smoothing. Ohtake et al. (2000) combined an inner fairness mechanism in their fairing process to increase the mesh regularity. Bajaj and Xu (2003) smooth both surfaces and functions on surfaces, in a C^2 smooth function space defined by the limit of triangular subdivision surfaces (quartic Box splines). Similar to the surface diffusion using the Laplacian, a more general class of PDE based methods called *flow surface techniques* have been developed which simulate different kinds of flows on surfaces (see (Westermann et al., 2000) for references) using the equation $\partial_t p - V(p,t) = 0$, where $V(p,t)$ represents the instantaneous stationary velocity field.

Level set methods were also used in surface fairing and surface reconstruction (see (Bajaj et al., 2003; Bertalmio et al., 2000; Chopp and Sethian, 1999; Museth et al., 2002; Osher and Fedkiw, 2000; Whitaker and Breen, 1998; Zhao et al., 2000)). In these methods, surfaces are formulated as iso-surfaces (level surfaces) of 3D functions, which are usually defined from the signed distance over Cartesian grids of a volume. An evolution PDE on the volume governs the behavior of the level surface. These level-set methods have several attractive features including, ease of implementation, arbitrary topology (see (Breen and Whitaker, 2001)) and a growing body of theoretical results. Often, fine surface structures are not captured by level sets, although it is possible to use adaptive (see (Preußer and Rumpf, 1999)) and triangulated grids as well as Hermite data (see (Kobbelt et al., 2001)). To reduce the computational complexity, Bertalmio et al. (2000) solve the PDE in a narrow band for deforming vectorial functions on surfaces (with a fixed surface represented by the level surface).

Recently, surface diffusion flow has been used to solve the surface blending problem and free-form surface fitting problem (Schneider and Kobbelt, 2000; Schneider and Kobbelt, 2001). In (Schneider and Kobbelt, 2000), fair meshes with G^1 conditions are created in the special case where the meshes are assumed to have subdivision connectivity. In this paper, local surface parameterization is still used to estimate the surface curvatures. The later paper (Schneider and Kobbelt, 2001) uses the same equation for smoothing meshes while satisfying G^1 boundary conditions. Outer fairness (the smoothness in the classical sense) and inner fairness (the regularity of the vertex distribution) criteria are used in their fairing

process. The finite element method is used by Clarenz et al. (2004) to solve the Willmore flow equation, based on a new variational formulation of the flow, for the aim of surface restoration. Willmore flow is also used to smooth triangular mesh in (Yoshizawa and Belyaev, 2002).

Main results. We use second order flows (mean curvature flow and averaged mean curvature flow) for G^0 continuity, fourth order flows for G^1 continuity and sixth order flows for G^2 continuity in each of several surface modelling problems. The proposed approach is simple and easy to implement. It is general, solves several surface modelling problems in the same manner, and gives very desirable results for a range of complicated free-form surface models, possibly having sharp features and corners. Furthermore, it avoids the estimation of normals or tangents or curvatures on the boundaries.

The rest of the paper is organized as follows: Section 2 describes several nonlinear PDEs used in this paper. In Section 3, we give details of the discretization and the numerical computation for the solutions of the PDEs. Examples to illustrate the different effects achievable from the solution of the PDEs are given in Section 4.

2. Partial differential equation models

Let $\mathcal{M}$ be a smooth surface and $p \in \mathcal{M}$ be the surface point. The general form of the geometric flows we consider is in the following form (see (Westermann et al., 2000))

$$\frac{\partial p}{\partial t} = V(p,t),$$

where $V(p,t) \in \mathbb{R}^3$ represents a velocity field. We shall focus our attention on using two classes velocity fields, one is curvature driven velocity field in the normal direction, the other is the higher order Laplace–Beltrami operators acting on surface point p.

2.1. Geometric partial differential equations

We now describe several geometric PDE models we use in this paper. More details on the existence and uniqueness of the solutions, the numerical computations of the solutions and evolution behaviors can be found in a series of papers by Mayer, Simonett, Escher (Escher et al., 1998; Escher and Simonett, 1998; Simonett, 2001) and Huiskens' (1987) paper. Let $\mathcal{M}_0$ be a compact closed immersed orientable surface in $\mathbb{R}^3$. A curvature driven geometric evolution consists of finding a family $\{\mathcal{M}(t)\colon\ t \geqslant 0\}$ of smooth closed immersed orientable surfaces in $\mathbb{R}^3$ which evolve according to the flow equation

$$\frac{\partial p}{\partial t} = N(p)V_n(k_1,k_2,p), \quad \mathcal{M}(0) = \mathcal{M}_0. \tag{2.1}$$

Here $p(t)$ is a surface point on $\mathcal{M}(t)$, $V_n(k_1,k_2,p)$ denotes the normal velocity of $\mathcal{M}(t)$, which depends on the principal curvatures k_1, k_2 of $\mathcal{M}(t)$, $N(p)$ stands for the unit normal of the surface at $p(t)$. In this paper we identify the surface point p and surface normal $N(p)$ as 3×1 matrices (column vectors). Hence, the arithmetic operations of these quantities are regarded matrix operations. The product of a scalar $a \in \mathbb{R}$ and a matrix M is written as either aM or Ma.

Let $A(t)$ denote the area of $\mathcal{M}(t)$, $V(t)$ denote the volume of the region enclosed by $\mathcal{M}(t)$. Then it has been shown that (see (Willmore, 1993, Theorem 4))

$$\frac{\mathrm{d}A(t)}{\mathrm{d}t} = \int\limits_{\mathcal{M}(t)} V_n H \,\mathrm{d}\sigma, \qquad \frac{\mathrm{d}V(t)}{\mathrm{d}t} = \int\limits_{\mathcal{M}(t)} V_n \,\mathrm{d}\sigma, \tag{2.2}$$

where $H = \frac{1}{2}(k_1 + k_2)$ is the mean curvature of $\mathcal{M}(t)$.

2.1.1. Mean curvature flow (*see* (Dziuk, 1991; White, 2002))

Taking $V_n = -H = -\frac{1}{2}(k_1 + k_2)$ in (2.1), we obtain the mean curvature flow PDE:

$$\frac{\partial p}{\partial t} = -N(p)H(p), \quad \mathcal{M}(0) = \mathcal{M}_0. \tag{2.3}$$

It follows from (2.2) that

$$\frac{\mathrm{d}A(t)}{\mathrm{d}t} = -\int\limits_{\mathcal{M}(t)} H^2 \,\mathrm{d}\sigma. \tag{2.4}$$

(2.4) implies that the mean curvature flow is area reducing.

2.1.2. Averaged mean curvature flow (*see* (Escher and Simonett, 1998; Huiskens, 1987; Sapiro, 2001))

In (2.1), if we take $V_n = h(t) - H(t)$, where $h(t) = \int_{\mathcal{M}(t)} H \,\mathrm{d}\sigma / \int_{\mathcal{M}(t)} \mathrm{d}\sigma$, then we have the averaged mean curvature flow PDE:

$$\frac{\partial p}{\partial t} = N(p)\big[h(t) - H(p)\big], \quad \mathcal{M}(0) = \mathcal{M}_0. \tag{2.5}$$

The existence proof of the global solutions to this flow can be found in Huiskens' (1987) paper. It follows from (2.2) that

$$\frac{\mathrm{d}A(t)}{\mathrm{d}t} = \int\limits_{\mathcal{M}(t)} (hH - H^2)\,\mathrm{d}\sigma = \int\limits_{\mathcal{M}(t)} \big[hH - H^2 - h(h-H)\big]\,\mathrm{d}\sigma = -\int\limits_{\mathcal{M}(t)} (h-H)^2\,\mathrm{d}\sigma \leqslant 0, \tag{2.6}$$

since obviously $\int_{M(t)} h(h-H) = h(h\int_{M(t)} \mathrm{d}\sigma - \int_{M(t)} H \,\mathrm{d}\sigma) = 0$. On the other hand, the second equation of (2.2) implies that

$$\frac{\mathrm{d}V(t)}{\mathrm{d}t} = h(t) \int\limits_{\mathcal{M}(t)} \mathrm{d}\sigma - \int\limits_{\mathcal{M}(t)} H \,\mathrm{d}\sigma = 0.$$

Hence the averaged mean curvature flow is volume preserving and area shrinking. The area shrinking stops if $H \equiv h$.

2.1.3. Surface diffusion flow (*see* (Schneider and Kobbelt, 2001))

If we take $V_n = \Delta H$, we get the so-called surface diffusion flow PDE:

$$\frac{\partial p}{\partial t} = N(p)\Delta H(p), \quad \mathcal{M}(0) = \mathcal{M}_0, \tag{2.7}$$

where $\Delta := \Delta_{\mathcal{M}}$ is Laplace–Beltrami operator which acts on functions defined on surface $\mathcal{M}(t)$. The existence and uniqueness of solutions for this flow is given in (Escher et al., 1998). From (2.2) and Green's formula we have

$$\frac{\mathrm{d}}{\mathrm{d}t}A(t) = \int\limits_{\mathcal{M}(t)} \Delta H H \,\mathrm{d}\sigma = -\int\limits_{\mathcal{M}(t)} |\nabla H|^2 \,\mathrm{d}\sigma \leqslant 0,$$

$$\frac{\mathrm{d}}{\mathrm{d}t}V(t) = \int\limits_{\mathcal{M}(t)} \operatorname{div}(\nabla H) \,\mathrm{d}\sigma = -\int\limits_{\mathcal{M}(t)} \nabla H \nabla(1) \,\mathrm{d}\sigma = 0,$$

where ∇ stands for the (tangential) gradient operator (see (do Carmo, 1976, pp. 101–102)) acting on differential functions defined on the surface $\mathcal{M}$. Hence, the surface diffusion flow is area shrinking, but volume preserving. The area stops shrinking when the gradient of H is zero. That is, $\mathcal{M}$ is a surface with constant mean curvature.

2.1.4. Higher order geometric flows

$$\frac{\partial p}{\partial t} = (-1)^{k+1} N(p) \Delta^k H(p), \quad \mathcal{M}(0) = \mathcal{M}_0. \tag{2.8}$$

Using Green formula, we have

$$\int\limits_{\mathcal{M}(t)} \Delta^k H \,\mathrm{d}\sigma = \int\limits_{\mathcal{M}(t)} \Delta(\Delta^{k-1} H) \,\mathrm{d}\sigma = \int\limits_{\mathcal{M}(t)} \nabla(\Delta^{k-1} H) \nabla(1) \,\mathrm{d}\sigma = 0.$$

Hence, the flow (2.8) is volume preserving if $k \geqslant 1$ from the second equation of (2.2).

Remark 2.1. We should note that the area/volume preserving/shrinking properties for the flows mentioned above are valid for closed surfaces. In our application of these flows, these properties may not be true since the surfaces always have fixed boundaries. For a open surface with fix boundary, the volume $V(t)$ could be defined as the directional volume between $\mathcal{M}(0)$ and $\mathcal{M}(t)$. It is easy to see that the volume preserving property for the averaged mean curvature flow is still valid. But for the higher order flow (2.8) ($k \geqslant 1$), this property is no longer valid, because a term related to the boundary does not vanish when Green's formula is used. For our modelling problems, volume preservation is not a desirable property (see Figs. 1 and 4).

Remark 2.2. In (Schneider and Kobbelt, 2001), Schneider and Kobbelt use elliptic equation $N(p)\Delta H(p) = 0$, while we use several time dependent parabolic type equations. In our approach, we have a progressive process starting from an initial value, so that a family of solutions is obtained. Such an approach is very desirable if the initial value is an approximation of the required solution.

2.2. Quasi geometric partial differential equations

Now we generalize the heat equation on a surface to the following higher order flows:

$$\frac{\partial p}{\partial t} = (-1)^{k+1} \Delta^k p, \quad \mathcal{M}(0) = \mathcal{M}_0, \quad k > 0. \tag{2.9}$$

Since $\Delta p = -2H(x)N(p)$, it is easy to see that (2.9) is the mean curvature flow when $k = 1$ (up to a factor 2). But since $(\Delta^k H)N \neq \Delta^k(HN)$ in general, (2.9) is different from the flow (2.8). To distinguish the difference between (2.8) and (2.9), we call (2.9) as a quasi geometric PDE.

The experiments conducted in this paper show that flows (2.9) sometimes behave better than the geometric flows mentioned above for our geometry modelling problems. However, the theoretical analysis on the existence and stability of their solutions is currently unavailable.

3. Solution of the PDEs

There are basically two classes of approaches for solving a PDE on any domain. One approach is based on finite divided differences, the other is based on finite elements (see (Bajaj and Xu, 2003; Clarenz et al., 2004; Deckelnick and Dziuk, 2002)). The approach we adopt in this paper is based on finite divided differences. Since we are dealing with differential equations over 2-manifolds in $\mathbb{R}^3$, the classical finite divided differences will be replaced by discretized differential geometric operators over surfaces. Section 3.1 deals with discretized geometric differential operators. Next in Section 3.2 we detail how the boundary conditions are respected. Discretizations of the PDEs in the spatial direction are described in Sections 3.3 and 3.4. Semi-implicit discretization in the time domain is considered in Section 3.5. Other issues, such as mesh regularization and initial mesh construction, are addressed in Section 3.6.

3.1. Discretized Laplace–Beltrami operator

One of the fundamental problems in solving our PDEs is the discretization of the Laplace–Beltrami operator. On a triangular surface mesh, several discretized approximations of the operator have been proposed (see (Desbrun et al., 1999; Guskov et al., 1999; Taubin, 2000; Xu, 2004b)). In this paper we adopt the discretization developed by Meyer et al. in (Meyer et al., 2002). A comparative research about the various discretized Laplace–Beltrami operators is conducted in (Xu, 2004a). It has been shown that the scheme of Meyer et al.'s is better for discretizing our PDEs. Let f be a smooth function on a surface, then Δf is approximated over a triangular mesh M by

$$\Delta f(p_i) \approx \frac{1}{A_M(p_i)} \sum_{j \in N_1(i)} \frac{\cot\alpha_{ij} + \cot\beta_{ij}}{2} \big[f(p_j) - f(p_i)\big], \tag{3.1}$$

where $N_1(i)$ is the index set of 1-ring of neighbor vertices of vertex p_i, α_{ij} and β_{ij} are the triangle angles shown in Fig. 2 (left). $A_M(p_i)$ is the area for vertex p_i as shown in Fig. 2 (right), where q_j is the circumcenter point for the triangle $[p_{j-1}p_j p_i]$ if the triangle is non-obtuse. If the triangle is obtuse, q_j is chosen to be the midpoint of the edge opposite to the obtuse angle. Since $\Delta p = -2H(p)N(p)$ (see (Willmore, 1993, p. 151)), we have

$$(\Delta p)_{p=p_i} = -2H(p_i)N(p_i) \approx \frac{1}{A_M(p_i)} \sum_{j \in N_1(i)} \frac{\cot\alpha_{ij} + \cot\beta_{ij}}{2} (p_j - p_i). \tag{3.2}$$

This gives an approximation of the mean curvature normal (see (Meyer et al., 2002)). The higher order Laplace–Beltrami operators are discretized recursively as

$$\Delta^k f(p_i) = \Delta(\Delta^{k-1} f)(p_i) = \frac{1}{A_M(p_i)} \sum_{j \in N_1(i)} \frac{\cot\alpha_{ij} + \cot\beta_{ij}}{2} \big[\Delta^{k-1} f(p_j) - \Delta^{k-1} f(p_i)\big] \tag{3.3}$$

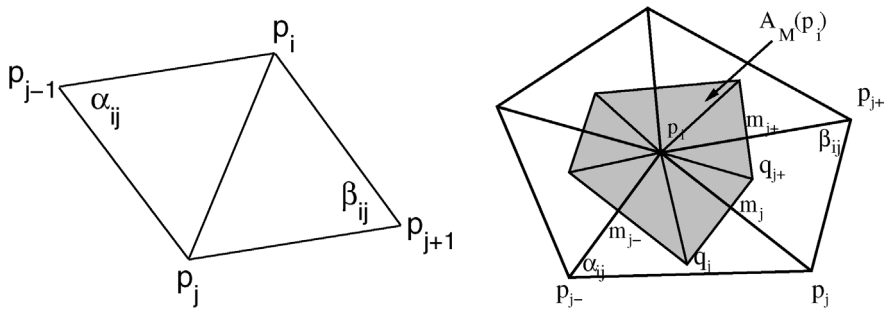

Fig. 2. Left: The definition of the angles α_{ij} and β_{ij}. Right: The definition of the area $A_M(p_i)$.

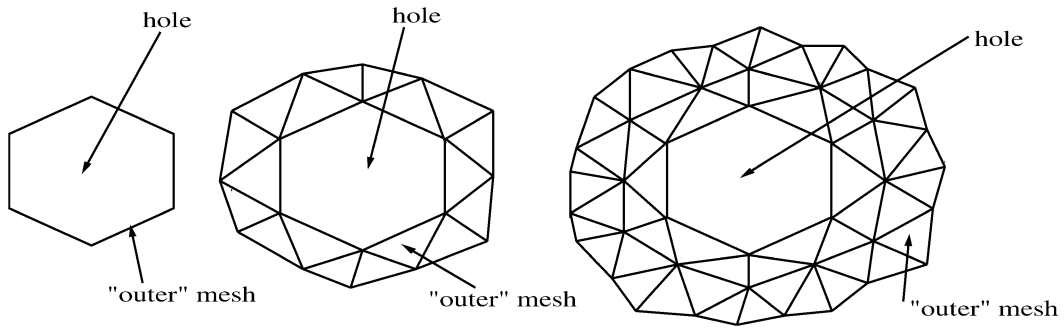

Fig. 3. Left: The involved vertices of the "outer" mesh for a G^0 boundary condition. The "outer" mesh is just the boundary of the hole. Middle: The involved vertices of the "outer" mesh for a G^1 boundary condition. Right: The involved vertices of the "outer" mesh for a G^2 boundary condition.

with $\Delta^0 f(p_i) = f(p_i)$. Note that $\Delta^k f(p_i)$ involves function values on a k-ring of neighboring vertices of p_i.

3.2. Handling of boundary conditions

3.2.1. Natural boundary conditions for blending and hole filling

By the natural boundary conditions, we mean that no continuity conditions are specified at the boundary points, but the continuity is implied by the "outer" mesh incident to the boundary of the hole (see Fig. 3). Such a treatment for boundary condition is suitable for both the blending problem and the N-sided hole filling problem, since the "outer" mesh always exists in such problems.

Let g_i be the order of continuity at a boundary point p_i, $g = \max g_i$. Then we can use the order $2g$ flow $\frac{\partial p}{\partial t} = (-1)^{g+1}\Delta^g H(p)N(p)$ for constructing the triangular surface patch with G^{g_i} continuity at the boundary vertex p_i. $\Delta^g H$ is discretized recursively: $\Delta^g H = \Delta(\Delta^{g-1}H)$. At a boundary vertex p_i, $\Delta^k H(p_i)$ is evaluated according to the following rule:

Evaluation rule at boundary. *$\Delta^k H(p_i)$ is evaluated recursively by formulas* (3.6) *and* (3.7) *if $k \leqslant g_i$, otherwise $\Delta^k H(p_i)$ is set to zero and the recursion stops.*

Note that even for an inner vertex p_j, the recursive definition may make $\Delta^k H(p_j)$ involve the evaluation of a lower order Laplace–Beltrami operator on the boundary. In general, the recursive evaluation of $\Delta^k H(p_i)$ at p_i (for p_i either being an inner or an outer vertex) involves $(k+1)$-ring neighbor vertices of p_i. Some of them may be inner vertices, and the remaining are outer vertices. The inner vertices are treated as unknowns in the discretized equations and the outers are incorporated into the right-hand side.

3.2.2. Natural boundary conditions for free-form surface filling

In the free-form surface filling problem, we are given a wireframe of curves (edges) and we wish to flesh the wireframe with surface patches that contain the curves as boundary with pre-specified order of continuity. At each of the intersection points of the patches, an order of continuity is pre-specified and the evaluation rule mentioned above is applied. For each inner point, a discretized linear equation is generated using the operator discretization (3.7). These linear equations for different patches are collected together and solved simultaneously. Note that one linear equation may involve inner vertices of several patches. However, if the continuity order at each boundary point is zero, any equation corresponding to an inner vertex does not involve inner vertices of other patches.

Remark 3.1. Schneider and Kobbelt (2001) use Moreton and Sequin's least square fitting of the second fundamental form relative to a local parameterization to estimate the required data on the boundary. These estimations of the boundary derivative data are based on incomplete information. Hence, the estimated data maybe not reliable. Our approach is based on the identity $\Delta_{\mathcal{M}} p = -2H(p)N(p)$. Hence, we do not need to estimate boundary derivative data, such as normals, tangents or curvatures. Furthermore, the boundary conditions are treated in the same way for equations with different orders.

3.3. Spatial discretization of quasi geometric flows

Let us consider first the discretization of (2.9) in the spatial direction for $k = 1, 2, 3$. Let $P = [p_1, \ldots, p_m]^T \in \mathbb{R}^{m\times 3}$, $\Delta P = [\Delta p_1, \ldots, \Delta p_m]^T \in \mathbb{R}^{m\times 3}$, where $p_1, \ldots, p_m$ are all the unknown vertices to be determined in each of our modelling problems. Then (3.2) could be written in matrix form:

$$\Delta P = -(\mathcal{D}\mathcal{W})P + B^{(1)}, \tag{3.4}$$

where $\mathcal{D} = \operatorname{diag}[\frac{1}{2A(p_1)}, \ldots, \frac{1}{2A(p_m)}]$ is a diagonal matrix, $\mathcal{W} = \{w_{ij}\}_{i,j=1}^m$ with

$$w_{ij} = \begin{cases} \sum_{k\in N_1(i)} \cot\alpha_{ik} + \cot\beta_{ik}, & i = j, \\ -(\cot\alpha_{ij} + \cot\beta_{ij}), & i \neq j,\ i \in N_1(j),\ j \in N_1(i), \\ 0, & \text{otherwise.} \end{cases}$$

Furthermore, $\mathcal{W}$ is a sparse, symmetric and positive definite matrix (see (Schneider and Kobbelt, 2001)). The constant term $B^{(1)} \in \mathbb{R}^{m\times 3}$ is obtained from the boundary conditions. It follows from (3.4) that

$$\Delta^2 P = (\mathcal{D}\mathcal{W}\mathcal{D}\mathcal{W})P + B^{(2)}, \tag{3.5}$$

where $B^{(2)} \in \mathbb{R}^{m\times 3}$ is obtained from the boundary conditions. Again, $\mathcal{W}\mathcal{D}\mathcal{W}$ is a sparse, symmetric and positive definite matrix. In general,

$$\Delta^k P = (-1)^k(\mathcal{D}\mathcal{W})^k P + B^{(k)},$$

and the matrix for $\mathcal{D}^{-1}(\mathcal{D}\mathcal{W})^k$ is also sparse, symmetric and positive definite.

3.4. Spatial direction discretization of geometric flows

Let

$$\omega_{ij} = \begin{cases} \sum_{k\in N_1(i)} \frac{\cot\alpha_{ik}+\cot\beta_{ik}}{2A_M(p_i)}, & i = j, \\ -\frac{\cot\alpha_{ij}+\cot\beta_{ij}}{2A_M(p_i)}, & i \neq j,\ i \in N_1(j),\ j \in N_1(i), \\ 0, & \text{otherwise}, \end{cases}$$

and $N(i) = N_1(i) \cup \{i\}$. Then we have

$$N(p_i)H(p_i) \approx \frac{1}{2} \sum_{j\in N(i)} \omega_{ij} p_j. \tag{3.6}$$

The higher order Laplace–Beltrami operators acting on H are discretized recursively as

$$\Delta^k H(p_i) = \Delta(\Delta^{k-1}H)(p_i) \approx -\sum_{j\in N(i)} \omega_{ij} \Delta^{k-1} H(p_j) \tag{3.7}$$

with

$$\Delta^0 H(p_i) = H(p_i) \approx \frac{1}{2} \sum_{j\in N(i)} \omega_{ij} N(p_i)^T p_j. \tag{3.8}$$

Note that $\Delta^k H(p_i)$ involves values of the mean curvature on a k-ring of neighboring vertices of p_i.

Using (3.6)–(3.8) and the evaluation rule at the boundary, we can write $N(p_i)\Delta^k H(p_i)$ as the following form:

$$N(p_i)\Delta^k H(p_i) \approx (-1)^k \sum_{j\in J_0} \omega_{ij}^{(k)} p_j + B_i^{(k)}, \quad \omega_{ij}^{(k)} \in \mathbb{R}^{3\times 3},\ B_i^{(k)} \in \mathbb{R}^3,$$

where J_0 is the index set of the (unknown) vertices to be determined, $B_i^{(k)}$ comes from boundary condition. To be more specific, let J denote the index set of the mesh M, J_k be the union of J_0 and the index set of the boundary vertices where C^k condition is specified. Then

$$\begin{aligned} N(p_i)H(p_i) &\approx \frac{1}{2} \sum_{j\in N(i)} \omega_{ij} p_j = \frac{1}{2} \sum_{j\in N(i)\cap J_0} \omega_{ij} p_j + \frac{1}{2} \sum_{j\in N(i)\cap\{J\setminus J_0\}} \omega_{ij} p_j \\ &= \sum_{j\in J_0} \omega_{ij}^{(0)} p_j + B_i^{(0)}, \end{aligned} \tag{3.9}$$

where $\omega_{ij}^{(0)} = \frac{1}{2}\omega_{ij} I_3$ for $j \in N(i) \cap J_0$, $\omega_{ij}^{(0)} = 0$ otherwise, $B_i^{(0)} = \frac{1}{2}\sum_{j\in N(i)\cap\{J\setminus J_0\}} \omega_{ij} p_j$. Similarly,

$$\begin{aligned} N(p_i)\Delta H(p_i) &\approx -N(p_i) \sum_{j\in N(i)} \omega_{ij} H(p_j) = -N(p_i) \sum_{j\in N(i)\cap J_1} \omega_{ij} H(p_j) \\ &= -N(p_i) \sum_{j\in N(i)\cap J_1} \omega_{ij} N(p_j)^T N(p_j) H(p_j) \\ &\approx -\sum_{j\in N(i)\cap J_1} \omega_{ij} N(p_i) N(p_j)^T \left[\sum_{k\in J_0} m_{jk}^{(0)} p_k + B_j^{(0)} \right] \end{aligned}$$

$$= -\sum_{j\in J_0} \omega_{ij}^{(1)} p_j + B_i^{(1)}, \tag{3.10}$$

$$\begin{aligned} N(p_i)\Delta^2 H(p_i) &\approx -N(p_i) \sum_{j\in N(i)} \omega_{ij} \Delta H(p_j) = -N(p_i) \sum_{j\in N(i)\cap J_2} \omega_{ij} \Delta H(p_j) \\ &\approx N(p_i) \sum_{j\in N(i)\cap J_2} \omega_{ij} \sum_{k\in N(j)\cap J_1} \omega_{jk} H(p_k) \\ &\approx \sum_{j\in N(i)\cap J_2} \sum_{k\in N(j)\cap J_1} \omega_{ij}\omega_{jk} N(p_i) N(p_k)^T \left[\sum_{l\in J_0} m_{kl}^{(0)} p_l + B_k^{(0)}\right] \\ &= \sum_{j\in J_0} \omega_{ij}^{(2)} p_j + B_i^{(2)}. \end{aligned} \tag{3.11}$$

(3.9)–(3.11) are used to discretize the right-handed side of (2.8) for $k = 0, 1, 2$. The discretization of $N(p_i)\Delta^k H(p_i)$ for $k > 2$ is recursively calculated using (3.7) and boundary conditions.

3.5. Time discretization

Given an approximate solution $\{p_i^{(n)}\}_{i=1}^m$ of the order $2k$ PDE at t_n for all the inner vertices, we construct an approximate solution $\{p_i^{(n+1)}\}_{i=1}^m$ for the next time step $t_{n+1} = t_n + \tau^{(n)}$ by using a semi-implicit Euler scheme. That is, we replace the derivative $\frac{\partial p}{\partial t}$ with $[p(t_{n+1}) - p(t_n)]/\tau^{(n)}$, and the quantities w_{ij} in (3.4), ω_{ij} and $N(p_i)$ in (3.6)–(3.8), $h(t)$ in (2.5) are computed using the previous result at t_n. Normals $N(p_i)$ are computed from Loop's subdivision surface (see (Bajaj and Xu, 2003) for detail). Such a treatment yields a linear system of equations with the inner vertices as unknowns. Let $\mathcal{P}^{(n+1)} = [(p_1^{(n+1)})^T, \ldots, (p_m^{(n+1)})^T]^T \in \mathbb{R}^{3m}$. The linear system for the geometric flows can be written as the matrix form

$$\left[I + \tau^{(n)}\mathcal{W}^{(k)}\right]\mathcal{P}^{(n+1)} = \mathcal{B}^{(k)}, \qquad \mathcal{W}^{(k)} = \left\{\omega_{ij}^{(k)}\right\}, \qquad \mathcal{B}^{(k)} \in \mathbb{R}^{3m}. \tag{3.12}$$

The matrix $\mathcal{W}^{(k)} \in \mathbb{R}^{3m\times 3m}$ is highly sparse, hence an iterative method for solving such a linear system is desirable. We use Saad's iterative method (Saad, 2000), named GMRES, to solve the system. The experiment shows that this iterative method works very well.

Let $P^{(n+1)} = [p_1^{(n+1)}, \ldots, p_m^{(n+1)}]^T \in \mathbb{R}^{m\times 3}$. The linear system for the flows (2.9) can be written as the matrix form

$$\left[I + \tau^{(n)}(\mathcal{D}\mathcal{W})^{k+1}\right]P^{(n+1)} = B^{(k)}, \quad \text{or} \quad W^{(k)}P^{(n+1)} = \mathcal{D}^{-1}B^{(k)} \tag{3.13}$$

where $B^{(k)} \in \mathbb{R}^{m\times 3}$, $W^{(k)} = \mathcal{D}^{-1} + \tau^{(n)}\mathcal{W}(\mathcal{D}\mathcal{W})^k \in \mathbb{R}^{m\times m}$ is a highly sparse, symmetric and positive definite matrix, and hence we use a conjugate gradient iterative method with diagonal preconditioning to solve the system.

Note that for the same size problem, the size of coefficient matrix in (3.12) is three times larger than that of coefficient matrix in (3.13). Furthermore, the matrix $W^{(k)}$ in (3.13) is symmetric and positive definite. The matrix in (3.12) is not. We also note that the discretization of (2.9) does not involve the computation of the surface normals.

Remark 3.2. It is well known that the condition of the linear system arising from the proposed semi-implicit discretization behaves like $O(1 + \tau^{(n)}h^{-2k})$, where h is the minimal edge length of the mesh.

Hence, if the mesh to be evolved is very irregular, the resulting system will be ill-conditioned. In such a case, a small time step size is required to make an iterative solver converge. Such a problem is relieved by the mesh regularization treatment (see Section 3.6). On the other hand, more advanced iterative method, such as multi-grid techniques based on a hierarchical mesh representation (see (Lang, 2001)) or algebraic multi-grid techniques, could be used to accelerate the iteration process. In the current implementation, these techniques are not incorporated.

Upper-bound of time step. It is known that several surface evolutions (e.g., the mean curvature flow (see (Dziuk, 1991; White, 2002)) and the surface diffusion flow (see (Bansch et al., 2002))) may develop singularities. For our geometric modelling problems, suppose we have a topologically correct initial surface mesh construction and we look for solutions that have the same topology as the initial mesh. Hence, we require that our solution is within the time period in that no singularity occurs. Therefore, we shall determine the time step $\tau^{(n)}$ so that t_n should not go beyond the time moment when the singularity occurs. Let $L(p_i^{(n)}, M(t_n))$ be the spatial discretization of $V(p,t)$ at vertex $p_i^{(n)}$ over the mesh $M(t_n)$. Then from the approximate equality

$$\left\| p_i^{(n+1)} - p_i^{(n)} \right\| = \tau^{(n)} \left\| L(p_i^{(n)}, M(t_n)) \right\|$$

and the requirement

$$\left\| p_i^{(n+1)} - p_i^{(n)} \right\| \leqslant \frac{1}{2} \min_{j \in N_1(i)} \left\| p_j^{(n)} - p_i^{(n)} \right\| \tag{3.14}$$

we determine an upper-bound for $\tau^{(n)}$ as follows

$$\tau^{(n)} \leqslant B_n := \frac{1}{2} \min_{1 \leqslant i \leqslant m} \left\{ \frac{\min_{j \in N_1(i)} \| p_j^{(n)} - p_i^{(n)} \|}{\| L(p_i^{(n)}, M(t_n)) \|} \right\}.$$

Requirement (3.14) guarantees that no vertex-collision happens. When the singularity is nearly to occur, the upper-bound B_n will approach to zero. Hence the evolution cannot move beyond the singular point for time.

Remark 3.3. When the singularity is nearly to occur, the upper-bound B_n will approach to zero. This will be a very low efficiency process. So a threshold value ϵ_0 should be put on the minimal B_n. If the determined B_n is smaller than the threshold value, we terminate the evolution process (see (3.17)–(3.18)).

3.6. Other important issues

3.6.1. Mesh regularization

The surface motion by the geometric PDEs described in Section 2 may cause a very irregular (nonuniform) distribution of the mesh vertices. Hence, introducing a regularization mechanism in the evolution process is necessary. Since *the tangential displacement does not influence the geometry of the deformation, just its parameterization* (see (Epstein and Gage, 1987)), we also add a tangential displacement to the motion. Hence, the general form of our geometric evolution problem could be written as

$$\frac{\partial p}{\partial t} = V(p,t) + V_t(p) T(p), \quad \mathcal{M}(0) = \mathcal{M}_0, \tag{3.15}$$

where $T(p)$ is a tangent direction at the surface point p, $V_t(p)$ is the tangential velocity. In the process of numerical solution of Eq. (3.15), $V_t(p)T(p)$ is chosen as

$$\mathcal{U}_0(p_i^{(n)}) - \big(\mathcal{U}_0(p_i^{(n)}), N(p_i^{(n)})\big)N(p_i^{(n)}) \tag{3.16}$$

where $\mathcal{U}_0(p_i^{(n)}) = \frac{1}{\text{card}(N_1(i))}\sum_{j\in N_1(i)}(p_j^{(n)} - p_i^{(n)})$, N is the surface normal computed from the limit surface of Loop's subdivision. This discretization of $V_t(p)T(p)$ is very similar to the one given by Ohtake et al. (2000), which is $\mathcal{U}_0(p_i^{(n)}) - (\mathcal{U}_0(p_i^{(n)}), N(p_i^{(n)}))\mathcal{U}_0(p_i^{(n)})$. The difference is that our displacement is in the tangent plane. In (3.16), $\mathcal{U}_0(p_i^{(n)})$ could be replaced by $\mathcal{U}_0(p_i^{(n+1)})$ to use as many of the new values as possible, and still yield a linear system. However, such a treatment destroys the symmetric property of the coefficient matrix. The tangential motion (3.16) is also used by Wood et al. (2000) and Ohtake et al. (2001).

3.6.2. Stopping criteria

We need to determine the minimal iteration number n, so that the evolution procedure stops at $t = t_n$. The following two criteria are used

$$\big\|M(t_n) - M(0)\big\| \geqslant \epsilon_1 \quad \text{or} \quad B_n < \epsilon_0 \tag{3.17}$$

$$\big\|M(t_{n+1}) - M(t_n)\big\|/\tau^{(n)} \leqslant \epsilon_2 \quad \text{or} \quad B_n < \epsilon_0 \tag{3.18}$$

where ϵ_i are given control constants, B_n is the determined upper-bound for $\tau^{(n)}$. Criterion (3.17) is for short time evolution, where we require $M(n\tau^{(n)})$ near $M(0)$. Criterion (3.18) is for long time evolution, where we are looking for a stable status of the solution. Condition $B_n < \epsilon_0$ is imposed for avoiding dead-loop around the singular point of time.

3.6.3. Construction of initial surface mesh

To provide an initial solution to the geometric evolution problem, we need to construct an initial triangular surface mesh ("filler") for each opening using any of a number of automatic or semi-automatic free-form surface construction techniques (Bajaj and Ihm, 1992; Bajaj et al., 1993; Davis et al., 2002; Greiner, 1994; Peters and Wittman, 1996; Xu et al., 2001). One can also interactively edit this "filler" to meet the weak assumptions for an initial solution shape.

Since the opening to be filled could be topologically complicated, we solve the problem in two steps. In the first step we fit each opening by an implicit algebraic surface or spline which interpolates or approximates the boundary data (Bajaj et al., 1993; Bajaj and Xu, 1994; Peters and Wittman, 1996). The approach we used is the one developed by Bajaj et al. (1992, 1993, 1994). In this approach, the data to be interpolated or approximated could be points or curves (even with normals). For ours, the boundary data are always points. Of course, this approach may not guarantee to produce topologically correct surfaces. If this happens, we break the opening into several parts by inserting a few curves (polygons) and then repeat the surface fitting for each part until we achieve a reasonable shape for the "filler".

After the algebraic surface is obtained, a triangulation step is employed. Since this triangulation should be consistent with the boundary polygon of the opening, we adopted the expansion technique developed in (Bajaj and Xu, 1994). Using this approach, we triangulate the surfaces starting from the boundary of the opening.

Remark 3.4. Comparing with finite element approach, the finite difference approach described above is easy to implement and it treats the equations with different orders in a uniform fashion. In the finite

element approach, one has to make efforts to derive a variational form for each of the PDEs. For higher order flows, hybrid method is used in general, such an approach will introduce much more unknowns, and therefore the resulted linear system is much larger. For example, in order to use finite element method (linear element) for the surface diffusion flow, Bänsch et al. (2002) split the PDE into a system of four equations.

4. Comparative examples

In this section, we give several examples to show how the PDEs are used to solve different problems in a uniform fashion. We also compare the effects of flows (2.8) and (2.9). All the figures produced by the fourth and sixth flows are generated using (2.9), except for the figures of the second row of Fig. 4 and third row of Fig. 6. These figures are produced using the flow (2.8). When we compare the effects of (2.8) and (2.9), we use the same number of iterations but double time step size for (2.8) because the factor 2 in the relation $\Delta p = -2HN$.

4.1. Comparison of the flows

The first three figures of the first row of Fig. 4 show the long time evolution solutions of the mean curvature flow, the fourth order flow, and the sixth order flow (2.9) for the input semi-sphere with an initial construction of the opening, a triangulated disk. The mean curvature flow does not change the disk. (b) and (c) are the results after 10 iterations with $\tau^{(n)} = 0.1$ and $\tau^{(n)} = 0.001$, respectively. Further iterations do not have a significant change on the shape of the solution surface. The fourth and sixth order flows yield convex surfaces and the smoothness is clearly observed. Also notice that the sixth order flow

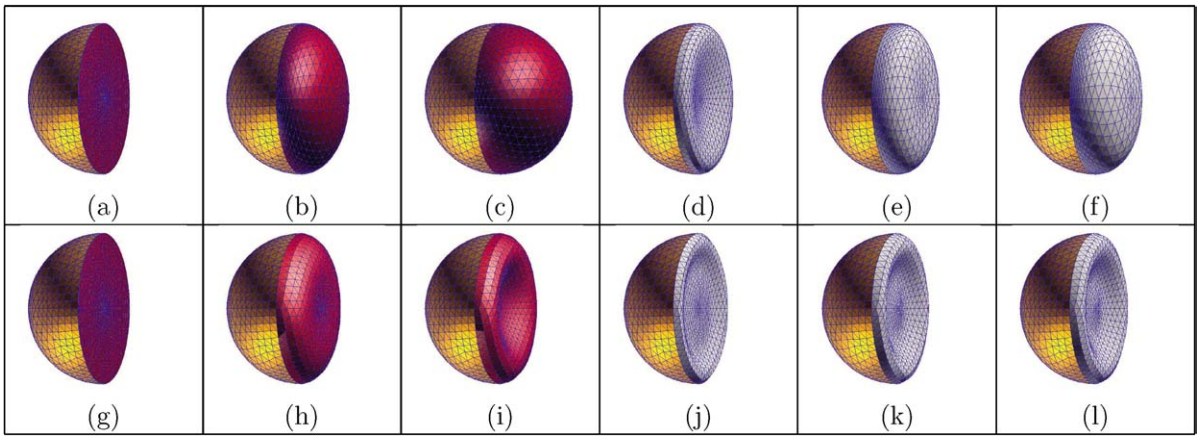

Fig. 4. The first and second row show the results of (2.9) and (2.8), respectively. (a) (same as (g)) The input semi-sphere (left part) with an initial planar triangulation of the disk opening. The mean curvature flow does not change the disk (initial mesh). (b) The result of fourth order flow after 10 iteration with $\tau^{(n)} = 0.1$. (c) The result of the sixth order flow after 10 iteration with $\tau^{(n)} = 0.01$. (d), (e) and (f) show three intermediate results of the sixth order flow with $\tau^{(n)} = 0.001$, and 1, 6 and 10 iterations, respectively. (h) The result of the surface diffusion flow after 10 iteration with $\tau^{(n)} = 0.2$. (i) The result of the sixth order flow (2.8) after 10 iteration with $\tau^{(n)} = 0.02$. (j), (k) and (l) show three intermediate results of the sixth order flow (2.8) with $\tau^{(n)} = 0.002$, and 1, 6 and 10 iterations, respectively.

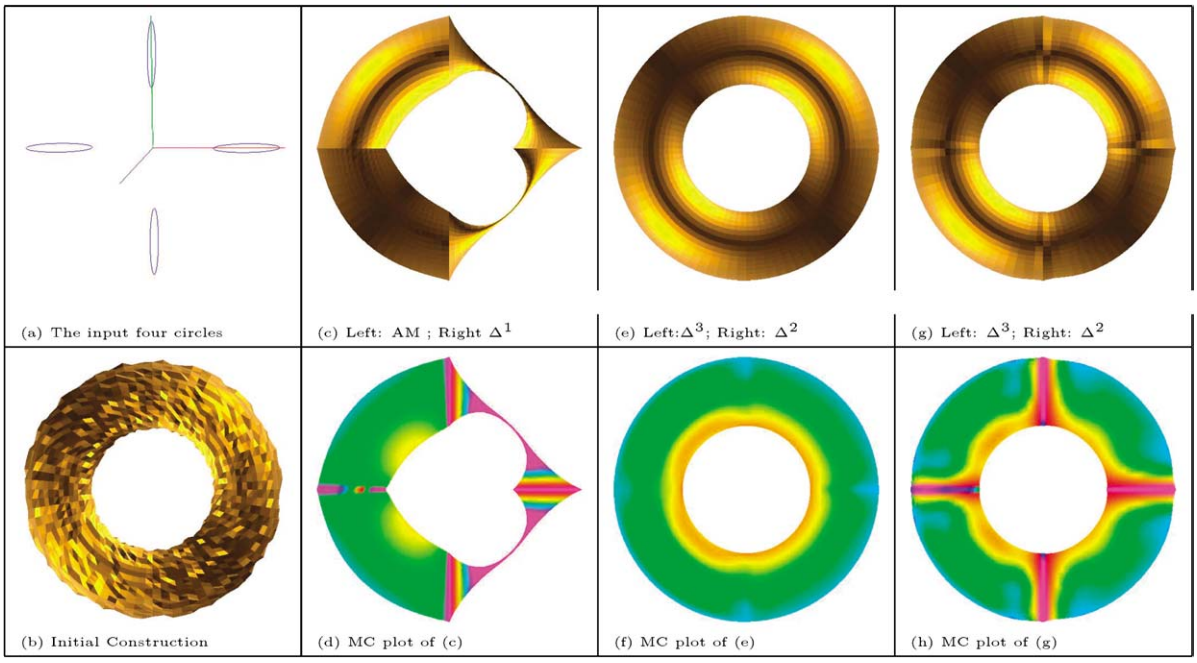

Fig. 5. Comparison of different flows. Δ^k represents $2k$ order flow (2.9) is used. AM denote the averaged mean curvature flow. The time step sizes for the second, fourth and sixth order flows are chosen to be 0.1, 0.0025, and 0.0000625, respectively. (c), (e), (g) are the faired interpolating surface meshes after 6 iterations, where the continuities at the boundary curves are set to 0, 2 and 0, respectively. (d), (f), (h) are the mean curvature (MC) plots of (c), (e), (g), respectively.

recovers the sphere accurately. The last three figures show three intermediate results of the sixth order flow. The second and third figures of the second row of Fig. 4 show the evolution solutions of the surface diffusion flow and sixth order flows (2.8) for the input semi-sphere with an initial construction of the opening. (h) and (i) are produced using the same number of iterations as (b) and (c), respectively, and double time step sizes. Again, the last three figures show three intermediate results of the sixth order flow. Comparing with the figures of the first row, the geometric flows change the surface shape in a much slower rate.

Remark 4.1. We have pointed that the geometric flows (2.8) have volume preserving properties for a closed surface. However, for an open surface with fixed boundary, the volume preserving properties are not guaranteed. (h) and (i) show that the volume preserving property is not valid.

Fig. 5 shows the combined use of different flows. The aim of this toy example is to illustrate the difference of these flows, especially the continuity on the patch boundaries. (a) shows four circles to be interpolated. Two of the circles are in the xz-plane, the other two are in the yz-plane. (b) shows an initial G^0 surface mesh constructed using (Bajaj and Ihm, 1992) with some additional noise added. (c), (e) and (g) are the faired interpolating surfaces after 6 iterations using different combinations of the flows. The time step sizes for the second, fourth and sixth order flows are chosen to be 0.1, 0.0025, and 0.0000625, respectively. Since the higher order flows evolve faster than the lower order flows, we use smaller time step sizes for higher order flows to obtain nearly the same surface evolution speed. Each of the meshes

consists of four surface patches. The left two patches are in the regions $R^{-+} := \{(x, y, z)\colon\ x \leqslant 0,\ y \geqslant 0\}$ and $R^{--} := \{(x, y, z)\colon\ x \leqslant 0,\ y \leqslant 0\}$, respectively, and generated by one type of flow. The right two patches are in the regions $R^{++} := \{(x, y, z)\colon\ x \geqslant 0,\ y \geqslant 0\}$ and $R^{+-} := \{(x, y, z)\colon\ x \geqslant 0,\ y \leqslant 0\}$, respectively, and generated by a different flow. (d), (f) and (h) are the mean curvature plots of (c), (e) and (g), respectively. The mean curvature at each vertex is computed by (3.2).

The aim of (c) is to show the difference between the mean curvature flow and the averaged mean curvature flow, where the left part is generated by the averaged mean curvature flow and the right part is produced by the mean curvature flow. The mean curvature flow shrinks the surface very fast while the averaged mean curvature flow does not. Further evolution using the mean curvature flow will yield a pinch-off of the surface. Therefore, if we model a surface patch using second order flows with G^0 boundary condition, the averaged mean curvature flow is more desirable than the mean curvature flow.

The patches in R^{-+} and R^{--} of (e) are produced by the sixth order flow (2.9) (with $k = 3$), while the patches in R^{++} and R^{+-} are produced by the fourth order flow (2.9). As a whole, the surface looks smooth, our curvature plot reveals the smoothness difference at the intersection curves, the sixth order flow gives a smoother result than the fourth order flow.

(g) is produced as (e), but the continuity order at the four circles are set to zero. Hence G^0 continuity is achieved there.

4.2. Surface blending

Given a collection surface mesh with boundaries, we construct a fair surface to blend the meshes at the boundaries with specified geometric continuity. Fig. 6 shows the case, where three cylinders to be blended are given (a) with an initial G^0 construction (b) using (Bajaj and Ihm, 1992) with some additional noise added. The blending surfaces (c), (e) and (g) are the faired blending meshes generated using the flow (2.9) with $k = 1, 2, 3$, respectively. These figures show the results after 32, 32 and 60 iterations with time step sizes 0.01, 0.001, and 0.0001, respectively. (d), (f) and (h) show the mean curvature plots correspondingly. These figures clearly show the difference of smoothness achieved at blending boundaries. The mean curvature flow gives G^0 continuity results. The fourth order flow produces smooth surfaces at boundaries. The sixth order flow produces even smoother surfaces as expected.

(i) and (k) are the faired blending meshes generated using the flow (2.8) with $k = 1, 2$, respectively. These figures show the results after 32 and 60 iterations with time step sizes 0.002 and 0.0002, respectively. (j) and (l) show the mean curvature plots of (i) and (k), respectively. It should be noted that the flows (2.9) generate little fatter surface than the flows (2.8).

4.3. N-sided hole filling

Given a surface mesh with a hole, we construct a fair surface to fill the hole with specified geometric continuity on the boundary. Fig. 1 shows such an example, where a head mesh with a hole in the nose subregion is given as input (a). An initial G^0 reconstruction of the nose is shown in (b) using (Bajaj and Ihm, 1992) and then evolved with the mean curvature flow. The blending surfaces ((c) and (d)) are generated using the flow (2.9) with $k = 2$ and 3, respectively. It should be observed that the sixth order flow yields a better restoration surface. The head mesh with the hole in the nose subregion is available from http://lsec.cc.ac.cn/~xuguo/xuguo2.htm.

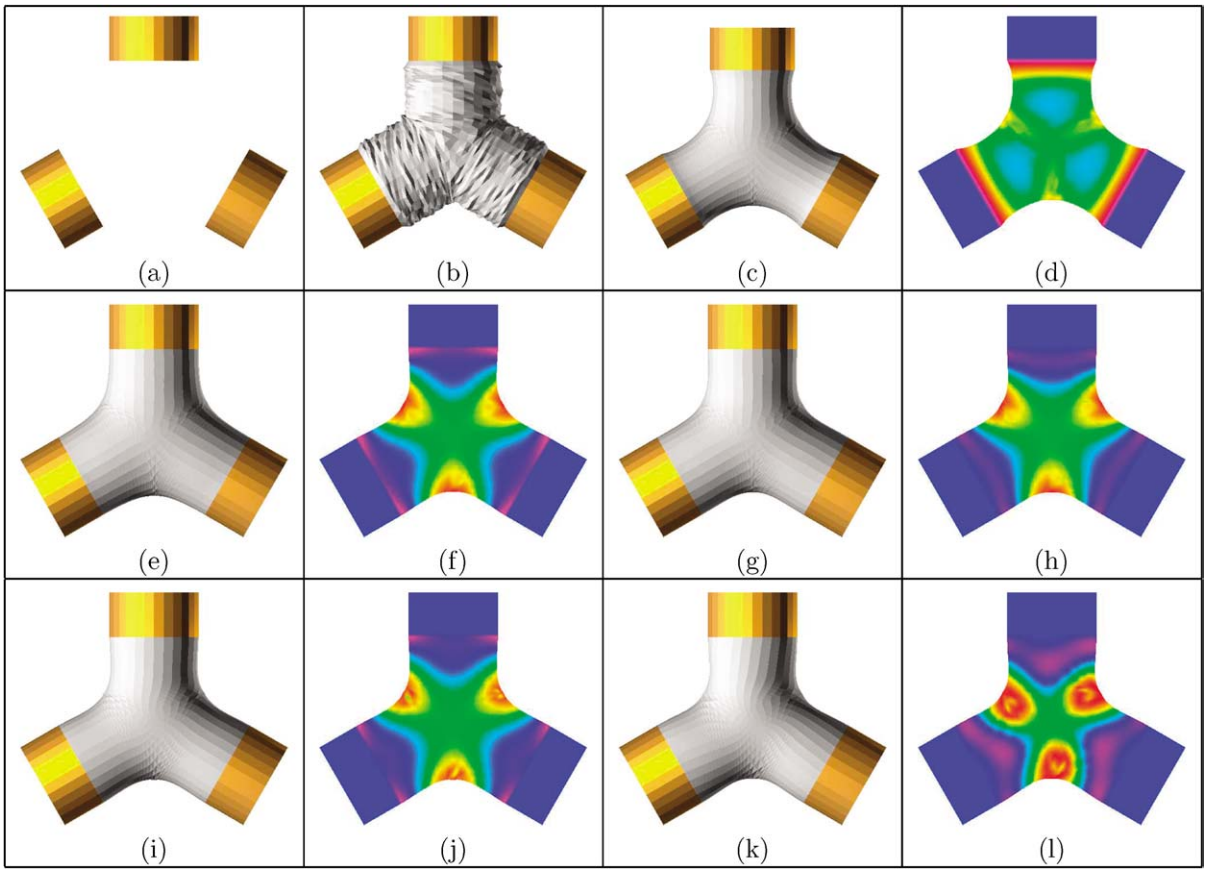

Fig. 6. (a) shows three cylinders to be blended. (b) shows the initial construction. (c), (e) and (g) are the faired blending meshes generated using the flow (2.9) with $k = 1, 2, 3$, respectively. These figures show the results after 32, 32 and 60 iterations with time step sizes 0.01, 0.001, and 0.0001, respectively. (d), (f) and (h) show the mean curvature plots correspondingly. (i) and (k) are the blending meshes generated using the flow (2.8) with $k = 1, 2$, respectively. These figures show the results after 32 and 60 iterations with time step sizes 0.002 and 0.0002, respectively. (j) and (l) show the mean curvature plots of (i) and (k), respectively.

4.4. Free-form surface construction

For the free-form surface fitting problem, we are given some curves, or partial patches, or points as input, and we wish to construct a fair surface mesh to interpolate this multi-dimensional data. Fig. 7 shows the approach of free-form surface construction, where some input curves with G^0 continuity requirement are given to preserve the sharp edges, and also given are some surface bands with a G^1 continuity requirement (see (a)). (b) shows an initial construction of the G^0 surface mesh using the patch filling scheme (Xu et al., 2001) with added noise. (c) is the faired surfaces, after 12 iterations, generated using the flow (2.9) with $k = 2$. The time step size is chosen to be 0.001. (d), (e) and (f) are zoomed in views of (a), (b) and (c), respectively.

Fig. 8 shows the free-form fitting approach from an input triangular mesh, where (a) shows the input surface triangular mesh with a G^1 continuity requirement at the vertices (see (a)). (b) shows an initial

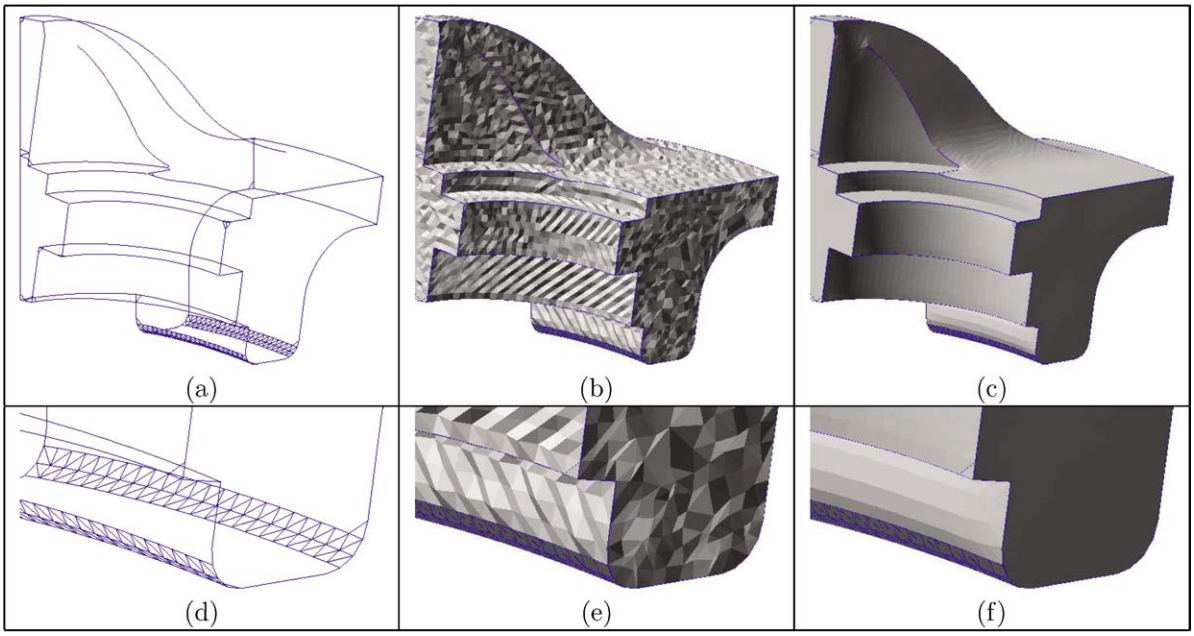

Fig. 7. Interpolating curves and patches: (a) shows some input curves with G^0 continuity requirement and some bands of mesh with G^1 continuity requirement. (b) shows an initial construction of the surface mesh. (c) is the faired surfaces, after 12 iterations, generated using the flow (2.9) with $k = 2$. The time step size is chosen to be 0.001. (d), (e) and (f) are the zoom in results of (a), (b) and (c), respectively.

construction of the surface mesh, where each input triangle is approximated with 16 sub-triangles. The newly introduced vertices are treated as unknowns and the input vertices are fixed in the fairing process. (c) and (d) are the faired meshes, after 2 iterations with $\tau^{(n)} = 0.01$, generated using the mean curvature flow and the averaged mean curvature flow, respectively. (e) is the faired mesh by fourth order flow, after 2 iterations with $\tau^{(n)} = 0.001$. (f) is the mean curvature plot of (e). The area shrinking of the mean curvature flow makes the input vertices to be interpolated become thorns (see (c)), while the area shrinking and the volume preservation of the averaged mean curvature flow make some of input vertices become thorns and some others become pits (see (d)). However, the fourth order flow does not suffer from this problem (see (e)). The obtained surface interpolates the input points and exhibits G^1 smoothness everywhere as well.

5. Conclusions

We have presented a general scheme for using PDEs to solve several surface modelling problems and with high order boundary continuity conditions. Our scheme has the following features: It produces very fair and desirable solution surfaces. It is simple and easy to implement. Specifically, it solves the free-form blending problem, the N-sided hole filling problem and free-form surface fitting problem in a uniform fashion, and solves the high order boundary continuity problem in an easy and natural way and avoids prior estimation of normals or derivative jets on the boundaries. The implementation results

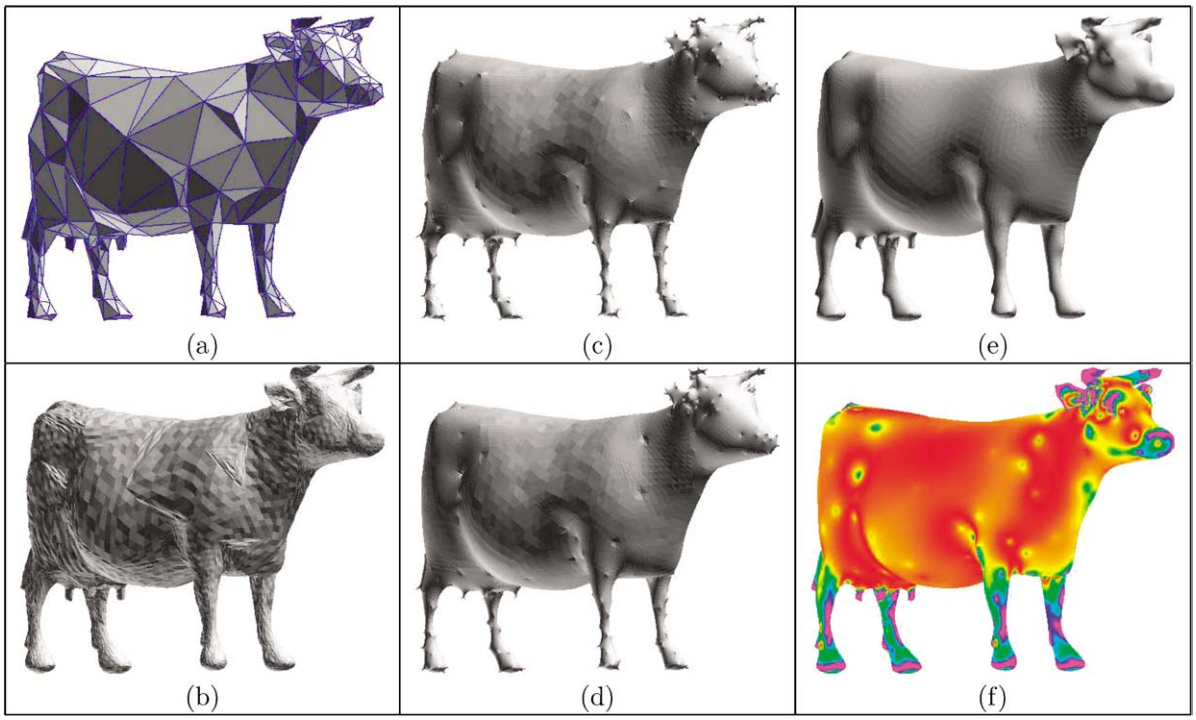

Fig. 8. Interpolating points: (a) shows some input points and their triangulation. (b) shows an initial construction of the surface mesh. (c) and (d) are the faired surfaces, after 2 iterations with $\tau^{(n)} = 0.01$, using the mean curvature flow and the averaged mean curvature flow, respectively. (e) is faired surfaces, after 2 iterations with $\tau^{(n)} = 0.001$, using the fourth order flow (2.9). (f) is the mean curvature plot of (e).

show that our solution works well for a wide range of surface models. Note that the C^1 or higher order continuity interpolatory surface blending solution produced by, e.g., (Bajaj and Ihm, 1992; Peters and Wittman, 1996) for complicated corners, or holes with many boundary curve segments, are usually of very high algebraic degree and thereby prone to be with unsuitable for certain applications. The current solution of starting with G^0 low degree blends, coupled with higher order flow evolution, yields in general a much better alternative for very smooth surface solutions.

Both the geometric flows and quasi geometric flows yield smooth surfaces at the boundaries. However, quasi geometric flows (2.9) have some attractive features, including ease of implementation, smaller and better behaved coefficient matrices and no requirement of derivatives (normal) estimation.

Acknowledgement

The authors are grateful to the referees for their carefully reading of the manuscript and for their helpful comments on improving the final version of this paper.

References

Bajaj, C., Ihm, I., 1992. Algebraic surface design with Hermite interpolation. ACM Trans. Graph. 19 (1), 61–91.

Bajaj, C., Xu, G., 1994. Rational spline approximations of real algebraic curves and surfaces. In: Dikshit, H.P., Michelli, C. (Eds.), Approximations and Decomposition Series. In: Advances in Computational Mathematics. World Scientific, Singapore, pp. 73–85.

Bajaj, C., Xu, G., 2003. Anisotropic diffusion of surface and functions on surfaces. ACM Trans. Graph. 22 (1), 4–32.

Bajaj, C., Ihm, I., Warren, J., 1993. Higher order interpolation and least squares approximation using implicit algebraic surfaces. ACM Trans. Graph. 12 (4), 327–347.

Bajaj, C., Wu, Q., Xu, G., 2003. Level-set based volumetric anisotropic diffusion for 3D image denoising. ICES Technical Report 03-10, University of Texas at Austin.

Bansch, E., Morin, P., Nochetto, R.H., 2002. Finite element methods for surface diffusion. In: Proc. of the International Conference on Free Boundary Problems, Theory and Applications, Trento, Italy.

Bertalmio, M., Sapiro, G., Cheng, L.T., Osher, S., 2000. A framework for solving surface partial differential equations for computer graphics applications. CAM Report 00-43, UCLA, Mathematics Department.

Bloor, M.I.G., Wilson, M.J., 1989. Generating blend surfaces using partial differential equations. Computer-Aided Design 21 (3), 165–171.

Bloor, M.I.G., Wilson, M.J., 1990. Using partial differential equations to generate free-form surfaces. Computer-Aided Design 22 (4), 221–234.

Breen, D., Whitaker, R., 2001. A level-set approach for the metamorphosis of solid models. IEEE Trans. Vis. Comput. Graph. 7 (2), 173–192.

do Carmo, M.P., 1976. Differential Geometry of Curves and Surfaces. Prentice-Hall, Englewood Cliffs, NJ.

Chopp, D.L., Sethian, J.A., 1999. Motion by intrinsic Laplacian of curvature. Interfaces and Free Boundaries 1, 1–18.

Clarenz, U., Diewald, U., Dziuk, G., Rumpf, M., Rusu, R., 2004. A finite element method for surface restoration with boundary conditions. Computer Aided Geometric Design 21 (5), 427–445.

Clarenz, U., Diewald, U., Rumpf, M., 2000. Anisotropic geometric diffusion in surface processing. In: Proceedings of Viz2000, IEEE Visualization, Salt Lake City, Utah, pp. 397–505.

Davis, J., Marschner, S.R., Garr, M., Levoy, M., 2002. Filling holes in complex surfaces using volumetric diffusion. In: Proc. First International Symposium on 3D Data Processing, Visualization, Transmission.

Deckelnick, K., Dziuk, G., 2002. A fully discrete numerical scheme for weighted mean curvature flow. Numer. Math. 91, 423–452.

Desbrun, M., Meyer, M., Schröder, P., Barr, A.H., 1999. Implicit fairing of irregular meshes using diffusion and curvature flow. In: SIGGRAPH99, Los Angeles, USA, pp. 317–324.

do Carmo, M., 1992. Riemannian Geometry. Boston.

Dziuk, G., 1991. An algorithm for evolutionary surfaces. Numer. Math. 58, 603–611.

Epstein, C.L., Gage, M., 1987. The curve shortening flow. In: Chorin, A., Majda, A. (Eds.), Wave Motion: Theory, Modeling, and Computation. Springer-Verlag, New York.

Escher, J., Simonett, G., 1998. The volume preserving mean curvature flow near spheres. Proc. Amer. Math. Soc. 126 (9), 2789–2796.

Escher, J., Mayer, U.F., Simonett, G., 1998. The surface diffusion flow for immersed hypersurfaces. SIAM J. Math. Anal. 29 (6), 1419–1433.

Greiner, G., 1994. Variational design and fairing of spline surface. Computer Graphics Forum 13, 143–154.

Guskov, I., Sweldens, W., Schröder, P., 1999. Mutiresolution signal processing for meshes. In: SIGGRAPH '99 Proceedings, pp. 325–334.

Huiskens, G., 1987. The volume preserving mean curvature flow. J. Reine Angew. Math. 382, 35–48.

Kobbelt, L., 1996. Discrete fairing. In: Goodman, T., Martin, R. (Eds.), The Mathematics of Surfaces VII. Information Geometers, pp. 101–129.

Kobbelt, L., Campagna, S., Vorsatz, J., Seidel, H.-P., 1998. Interactive muti-resolution modeling on arbitrary meshes. In: SIGGRAPH98, pp. 105–114.

Kobbelt, L., Botsch, M., Schwanecke, U., Seidel, H.P., 2001. Feature sensitive surface extraction from volume data. In: SIGGRAPH2001, pp. 51–66.

Lang, J., 2001. Adaptive Multilevel Solution of Nonlinear Parabolic PDE Systems: Theory, Algorithm, and Applications. Springer, Berlin.

Meyer, M., Desbrun, M., Schröder, P., Barr, A., 2002. Discrete differential-geometry operator for triangulated 2-manifolds. In: Proceedings of Visual Mathematics'02, Berlin, Germany.

Museth, K., Breen, D., Whitaker, R., Barr, A., 2002. Level set surface editing operators. In: SIGGRAPH02, pp. 330–338.

Ohtake, Y., Belyaev, A.G., Bogaevski, I.A., 2000. Polyhedral surface smoothing with simultaneous mesh regularization. In: Geometric Modeling and Processing Proceedings, pp. 229–237.

Ohtake, Y., Belyaev, A.G., Bogaevski, I.A., 2001. Mesh regularization and adaptive smoothing. Computer-Aided Design 33, 789–800.

Osher, S.J., Fedkiw, R.P., 2000. Level set methods. CAM Report 00-07, UCLA, Mathematics Department.

Peters, J., Wittman, M., 1996. Smooth blending of basic surfaces using trivariate box spline. In: IMA 96, The Mathematics of Surfaces. Dundee, UK.

Preußer, T., Rumpf, M., 1999. An adaptive finite element method for large scale image processing. In: Scale-Space Theories in Computer Vision, pp. 232–234.

Saad, Y., 2000. Iterative Methods for Sparse Linear Systems, second edition with corrections.

Sapiro, G., 2001. Geometric Partial Differential Equations and Image Analysis. Cambridge University Press, Cambridge.

Schneider, R., Kobbelt, L., 2000. Generating fair meshes with G^1 boundary conditions. In: Geometric Modeling and Processing, Hong Kong, China, pp. 251–261.

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

Simonett, G., 2001. The Willmore flow for near spheres. Differential and Integral Equations 14 (8), 1005–1014.

Taubin, G., 1995. A signal processing approach to fair surface design. In: SIGGRAPH'95 Proceedings, pp. 351–358.

Taubin, G., 2000. Signal processing on polygonal meshes. In: EUROGRAPHICS.

Weickert, J., 1998. Anisotropic Diffusion in Image Processing. B.G. Teubner, Stuttgart.

Westermann, R., Johnson, C., Ertl, T., 2000. A level-set method for flow visualization. In: Proceedings of Viz2000, IEEE Visualization, Salt Lake City, Utah, pp. 147–154.

Whitaker, R., Breen, D., 1998. Level set models for the deformation of solid objects. In: Proceedings of the 3rd International Workshop on Implicit Surfaces, Eurographics Association, pp. 19–35.

White, B., 2002. Evolution of curves and surfaces by mean curvature. In: Proceedings of the International Congress of Mathematicians, vol. I, Beijing, pp. 525–538.

Willmore, T.J., 1993. Riemannian Geometry. Clarendon Press, Oxford.

Wood, Z.J., Breen, D., Desbrun, M., 2000. Semi-regular mesh extraction from volumes. In: IEEE Visualization Proceedings of the conference on Visualization'00, Salt Lake City, Utah, United States, pp. 275–282.

Xu, G., 2004a. Convergence of discrete Laplace–Beltrami operators over surfaces. Comput. Math. Appl. 48, 347–360.

Xu, G., 2004b. Discrete Laplace–Beltrami operators and their convergence. Computer Aided Geometric Design 21, 767–784.

Xu, G., Bajaj, C., Huang, H., 2001. C^1 modeling with A-patches from rational trivariate functions. Computer Aided Geometric Design 18 (3), 221–243.

Yoshizawa, S., Belyaev, A.G., 2002. Fair triangle mesh generation with discrete elastica. In: Geometric Modeling and Processing, Saitama, Japan, pp. 119–123.

Zhao, H.K., Osher, S., Merriman, B., Kang, M., 2000. Implicit and non-parametric shape reconstruction from unorgainzed points using variational level set method. Computer Vision and Image Understanding 80 (3), 295–319.