Minimal Mean-Curvature-Variation Surfaces and Their Applications in Surface Modeling*

Guoliang Xu[1] and Qin Zhang[2]

[1,2] LSEC, Institute of Computational Mathematics, Academy of Mathematics and System Sciences, Chinese Academy of Sciences, Beijing 100080, China
[2] Department of Basic Courses, Beijing Information Science and Technology University, Beijing 100085, China

Abstract. Physical based and geometric based variational techniques for surface construction have been shown to be advanced methods for designing high quality surfaces in the fields of CAD and CAGD. In this paper, we derive a Euler-Lagrange equation from a geometric invariant curvature integral functional–the integral about the mean curvature gradient. Using this Euler-Lagrange equation, we construct a sixth-order geometric flow (named as minimal mean-curvature-variation flow), which is solved numerically by a divided-difference-like method. We apply our equation to solving several surface modeling problems, including surface blending, N-sided hole filling and point interpolating. The illustrative examples provided show that this sixth-order flow yields high quality surfaces.

Keywords: Euler-Lagrange equation, Minimal mean-curvature-variation flow, Energy functional, Discretization, Surface modeling.

1 Introduction

Problems such as surface fairing, free-from surface design, surface blending and N-sided hole filling have been important issues in the areas of CAD and CAGD. These problems can be efficiently solved by an energy-based variational approach (e. g. [2,6,14,15]). Roughly speaking, the variational approach is to pursue a curve or surface which minimizes certain type of energy simultaneously satisfying prerequisite boundary conditions. A problem one meets within this approach is the choice of energy models. Energy models previously used can be classified into the categories of physical based and geometric based. The class of physical

where $f(x, y)$ and Ω are surface parametrization and its domain, respectively. Recently, energy functionals based on geometric invariants begin to lead in this field. As is well-known, area functional and total curvature functional (see [7])

$$\mathscr{E}_3(\mathcal{M}) := \int_{\mathcal{M}} dA, \qquad \mathscr{E}_4(\mathcal{M}) := \int_{\mathcal{M}} (k_1^2 + k_2^2)\, dA$$

are the most frequently used energies, where k_1 and k_2 are the principal curvatures. The energy

$$\mathscr{E}_5(\mathcal{M}) := \int_{\mathcal{M}} \left[\left(\frac{dk_1}{d\mathbf{e}_1} \right)^2 + \left(\frac{dk_2}{d\mathbf{e}_2} \right)^2 \right] dA$$

proposed by Moreton et al. in [11] punishes the variation of the principal curvatures, where $\mathbf{e}_1$ and $\mathbf{e}_2$ are principal directions corresponding to the principal curvatures k_1 and k_2. The advantage of utilizing physical based models is that the resulting equations are linear and therefore easy to solve. The disadvantage is that the resulting equations are parameter dependent. Energy models based on geometric invariants can overcome this shortcoming.

Another critical problem of the variational approach is how to find out those surfaces which minimize these energy functionals. Two approaches have been employed to solve this problem. One method is using the optimization approach (see [6,11,15]). The minimization problem can be discretized to arrive at finite dimensional linear or nonlinear systems. Approximate solutions are then obtained by solving the constructed systems. Another widely accepted method is based on variational calculus. The first step of this method is to calculate the Euler-Lagrange equations for the energy functionals, then solve these equations for the ultimate surface. This method is superior to the optimization technique in general because optimization is lack of local shape control and computationally expensive.

To solve the Euler-Lagrange equations, gradient descent flow method has been introduced and widely accepted. For instance, from the Euler-Lagrange equation $H = 0$ of $\mathscr{E}_3(\mathcal{M})$, the well-known mean curvature flow $\frac{\partial \mathbf{r}}{\partial t} = H\mathbf{n}$ is constructed. Here $\mathbf{n}$ is the normal vector field of the surface. When the steady state of the flow is achieved, we obtain $H = 0$. Similarly, *Willmore surfaces* (see [16]), the solution of the Euler-Lagrange equation $\Delta H + 2H(H^2 - K) = 0$ of the energy

$$\mathscr{E}_6(\mathcal{M}) := \int_{\mathcal{M}} H^2 dA,$$

applications. For instance, in the shape design of the streamlined surfaces of aircraft, ships and cars, G^2 continuous surfaces are crucial. Therefore, higher order flow need to be considered. On this aspect, Xu et al. have utilized a sixth-order flow in [20] to achieve G^2 continuity and Zhang et al. have used another sixth-order PDE in [21,22] to obtain C^2 continuity. A sixth-order equation is also proposed in [3] by Botsch and Kobbelt to conduct real-time freeform medeling. But all these sixth-order flows and PDEs are neither physical based nor geometric based in the sense mentioned above.

In this paper, a sixth-order geometric based PDE is introduced. It is derived from the Euler-Lagrange equation of the energy functional

$$\mathscr{F}(\mathcal{M}) := \int_{\mathcal{M}} \|\nabla H\|^2 dA, \tag{1.1}$$

which punishes the variation of mean curvature. A surface which minimizes functional (1.1) is called *minimal mean-curvature-variation surface*. We expect that G^2 continuity can be achieved using this sixth-order flow in solving the surface modeling problems, such as surface blending, N-sided hole filling and scattered points interpolation. A semi-implicit divided-difference-like discretization scheme is proposed to solve the highly nonlinear PDE. The experimental and comparative results show that high quality surfaces are obtained.

The rest of this paper is organized as follows. In section 2, some used notations and preliminaries are introduced. One sixth-order flow is derived in section 3. The numerical solving of the flow is discussed in section 4. The application and examples are provided in section 5. Section 6 concludes this paper.

2 Notations and Preliminaries

In this section, we introduce some notations and several differential operators defined on surface used throughout this paper. Let $\mathcal{M}$ be a regular parametric surface represented as $\mathbf{r}(u,v) \in \mathbb{R}^3, \quad (u,v) \in \Omega \subset \mathbb{R}^2$, whose unit normal vector is $\mathbf{n} = \frac{\mathbf{r}_u \times \mathbf{r}_v}{\|\mathbf{r}_u \times \mathbf{r}_v\|}$ after suitable orientation being chosen, where the subscript of $\mathbf{r}$ denotes partial derivative and $\|\mathbf{x}\| := \langle \mathbf{x}, \mathbf{x} \rangle^{\frac{1}{2}} := (\mathbf{x}^T \mathbf{x})^{\frac{1}{2}}$ is the usual Euclidean norm. Superscript T stands for the transpose operation. We assume at least $\mathbf{r} \in C^6(\overline{\Omega}, \mathbb{R}^3)$. The coefficients of the first fundamental form and the second fundamental form are

surface $\mathcal{M}$ is a self-adjoint linear map on the tangent space $T_{\mathbf{r}}\mathcal{M} := \text{span}\{\mathbf{r}_u, \mathbf{r}_v\}$ defined by (see [5])

$$S : T_{\mathbf{r}}\mathcal{M} \to T_{\mathbf{r}}\mathcal{M}, \quad S(\mathbf{v}_{\mathbf{r}}) = -D_{\mathbf{v}}\mathbf{n},$$

where $\mathbf{v}_{\mathbf{r}}$ is an arbitrary tangent vector of $\mathcal{M}$ at point $\mathbf{r}$ and $\mathbf{v}$ is a tangent vector field satisfying $\mathbf{v}(\mathbf{r}) = \mathbf{v}_{\mathbf{r}}$, and $D_{\mathbf{v}}$ is directional derivative operator along direction $\mathbf{v}$. We can represent this linear map by a matrix as $S = [b_{\alpha\beta}][g^{\alpha\beta}]$. The trace divided by 2 and determinant of S, $H = \text{tr}(S)/2$, $K = \det(S)$ are the *mean curvature* and *Gaussian curvature*, respectively. Now let us introduce some used differential operators defined on surface $\mathcal{M}$.

Tangential gradient operator. Let f be a smooth function on $\mathcal{M}$. Then the *tangential gradient operator* ∇ acting on f is given by

$$\nabla f = [\mathbf{r}_u,\ \mathbf{r}_v][\, g^{\alpha\beta}\,][f_u,\ f_v]^T \in \mathbb{R}^3.$$

Second tangent operator. Let f be a smooth function on $\mathcal{M}$. Then the *second tangent operator* $\diamond$ acting on f is given by

$$\diamond f = [\mathbf{r}_u,\ \mathbf{r}_v][\, K\, b^{\alpha\beta}\,][f_u,\ f_v]^T \in \mathbb{R}^3.$$

Divergence operator. Let $\mathbf{v}$ be a C^1 smooth vector field on $\mathcal{M}$. Then the *divergence* of $\mathbf{v}$ is defined by

$$\text{div}(\mathbf{v}) = \frac{1}{\sqrt{g}}\left[\frac{\partial}{\partial u}, \frac{\partial}{\partial v}\right]\left[\,\sqrt{g}\,[\,g^{\alpha\beta}\,]\,[\mathbf{r}_u,\ \mathbf{r}_v]^T\mathbf{v}\,\right].$$

Note that if $\mathbf{v}$ is a normal vector field of $\mathcal{M}$, $\text{div}(\mathbf{v}) = 0$.

Laplace-Beltrami operator. Let $f \in C^2(\mathcal{M})$. Then ∇f is a smooth vector field on $\mathcal{M}$. The Laplace-Beltrami operator (LBO) Δ applying to f is defined by $\Delta f = \text{div}(\nabla f)$. From the definitions of ∇ and div, we can easily derive that

$$\Delta f = \frac{1}{\sqrt{g}}\left[\frac{\partial}{\partial u}, \frac{\partial}{\partial v}\right]\left[\,\sqrt{g}\,[\,g^{\alpha\beta}\,]\,[f_u,\ f_v]^T\,\right].$$

It is easy to see that Δ is a second-order differential operator which relates closely to the *mean curvature normal* $\mathbf{H} := H\mathbf{n}$ by the relation

$$\Delta\mathbf{r} = 2\mathbf{H}. \tag{2.1}$$

Theorem 3.1. *Let $\mathscr{F}(\mathcal{M})$ be defined as* (1.1)*. Then the Euler-Lagrange equation of $\mathscr{F}(\mathcal{M})$ is*

$$\Delta^2 H + 2(2H^2 - K)\Delta H + 2\langle \nabla H, \diamond H\rangle - 2H\|\nabla H\|^2 = 0. \tag{3.1}$$

Proof. At first, we can rewrite the functional (1.1) as

$$\mathscr{F}(\mathcal{M}) = \int_\Omega \|\nabla H\|^2 \sqrt{g} du^1 du^2, \tag{3.2}$$

which is parameter-invariant. Consider now an extremal $\mathcal{M}$ of functional (3.2) and a family of normal variation $\underline{\mathbf{r}}(w,\varepsilon)$ of $\mathcal{M}$ defined by

$$\underline{\mathbf{r}}(w,\varepsilon) = \mathbf{r}(w) + \varepsilon\varphi(w)\mathbf{n}(w), \quad w \in \overline{\Omega}, \quad |\varepsilon| \ll 1,$$

where $\varphi \in C_c^\infty(\Omega) := \{\phi \in C^\infty(\Omega, \mathbb{R}); \mathrm{supp}\phi \subset \Omega\}$. Then we obtain

$$0 = \frac{d}{d\varepsilon}\mathscr{F}(\underline{\mathcal{M}}(\cdot,\varepsilon))\big|_{\varepsilon=0} =: \delta\mathscr{F}(\mathcal{M},\varphi), \tag{3.3}$$

where

$$\delta\mathscr{F}(\mathcal{M},\varphi) = \int_\Omega \left[\delta(\|\nabla H\|^2) + \|\nabla H\|^2(\delta\sqrt{g})/\sqrt{g}\right]\sqrt{g} du^1 du^2. \tag{3.4}$$

From

$$\begin{aligned} &\delta(g_{\alpha\beta}) = -2\varphi b_{\alpha\beta}, && \delta(g) = -4gH\varphi, \\ &\delta(\sqrt{g}) = -2H\varphi\sqrt{g}, && \delta(H) = (2H^2 - K)\varphi + \tfrac{1}{2}\Delta\varphi, \end{aligned}$$

we can deduce that

$$\delta(\|\nabla H\|^2) = 4H\varphi\|\nabla H\|^2 - 2\langle\nabla H, \diamond H\rangle\varphi + 2\langle\nabla H, \nabla[(2H^2 - K)\varphi + \frac{1}{2}\Delta\varphi]\rangle. \tag{3.5}$$

Substituting (3.5) into (3.4), we arrive at

$$\begin{aligned} \delta\mathscr{F}(\mathcal{M},\varphi) &= \int_\Omega \Big[2H\|\nabla H\|^2\varphi - 2\langle\nabla H, \diamond H\rangle\varphi \\ &\quad + 2\langle\nabla H, \nabla[(2H^2 - K)\varphi + \frac{1}{2}\Delta\varphi]\rangle\Big]\sqrt{g} du^1 du^2 \\ &= \int_\Omega \Big[2H\|\nabla H\|^2\varphi - 2\langle\nabla H, \diamond H\rangle\varphi \end{aligned}$$

Obviously, equation (3.1) is of sixth-order. It is easy to see that surfaces with constant mean curvature, such as Delaunay surfaces (see [12], pp. 144-148) (include unduloid and nodoid), sphere, cylinder, and minimal surfaces, are the solutions of the equation. But tori and cone are not the solution surfaces of the equation. It is not difficult to derive that

Theorem 3.2. *Equation* (3.1) *is invariant under the transforms of rotation, translation and scaling.*

Here the *invariant* means that a solution surface of (3.1) is still a solution under the three transforms mentioned.

Now let us introduce the sixth-order flow used in this paper. Let $\mathcal{M}_0$ be a compact immersed orientable surface in $\mathbb{R}^3$. A curvature driven geometric evolution consists of finding a family $\{\mathcal{M}(t) : t \geq 0\}$ of smooth immersed orientable surfaces in $\mathbb{R}^3$ which evolve according to the flow equation

$$\frac{\partial \mathbf{r}(t)}{\partial t} = \mathbf{n}V, \quad \mathcal{M}(0) = \mathcal{M}_0. \tag{3.6}$$

Here $\mathbf{r}(t)$ is a surface point on $\mathcal{M}(t)$, V denotes the normal velocity of $\mathcal{M}(t)$. Let $\mathcal{M}(t)$ be a closed surface with outward normal. Then it has been shown that (see [9], Theorem 4)

$$\frac{dA(t)}{dt} = -2\int_{\mathcal{M}(t)} VH\, dA, \quad \frac{dV(t)}{dt} = \int_{\mathcal{M}(t)} V\, dA, \tag{3.7}$$

where $A(t)$ denotes the area of surface $\mathcal{M}(t)$ and $V(t)$ denotes the volume of the region enclosed by $\mathcal{M}(t)$. If $\frac{dA(t)}{dt} = 0$, we say the flow is area-preserving. Similarly, the flow is volume-preserving if $\frac{dV(t)}{dt} = 0$.

Let $\mathcal{M}_0$ be a compact orientable surface in $\mathbb{R}^3$ with boundary Γ. Then the sixth-order flow constructed from the Euler-Lagrange equation (3.1) is

$$\begin{cases} \dfrac{\partial \mathbf{r}}{\partial t} = \left[\Delta^2 H + 2(2H^2 - K)\Delta H + 2\langle \nabla H, \diamond H\rangle - 2H\|\nabla H\|^2\right]\mathbf{n}, \quad \mathbf{r} \in \mathcal{M}(t), \\ \mathcal{M}(0) = \mathcal{M}_0, \quad \partial\mathcal{M}(t) = \Gamma. \end{cases} \tag{3.8}$$

If $\mathcal{M}(t)$ is a closed constant mean curvature surface, (3.7) implies that $\frac{dA(t)}{dt} = 0$ and $\frac{dV(t)}{dt} = 0$ for the flow (3.8). In general, this area-preserving or volume-preserving properties are not valid.

4 Numerical Solving of the GPDE

Discretizations of curvatures and geometric differential operators. To solve the geometric PDE (3.8) over a triangular surface mesh M with vertex set $\{\mathbf{r}_i\}$ using a divided-difference-like method, discrete approximations of the mean curvature, Gaussian curvature and various differential operators are required. In order to use a semi-implicit scheme, we require the approximations of the differential operators mentioned above at $\mathbf{r}_i$ to have the following form

$$\Theta f(\mathbf{r}_i) = \sum_{j \in N_1(i)} w_{ij}^{\Theta} f(\mathbf{r}_j),$$

where Θ represents one of above mentioned differential operators and $w_{ij}^{\Theta} \in \mathbb{R}$ or $w_{ij}^{\Theta} \in \mathbb{R}^3$, $N_k(i)$ is the index set of the k-ring neighbor vertices of $\mathbf{r}_i$. Although there are several discretization schemes of Laplace-Beltrami operator and Gaussian curvature (see [17,19] for a review), the discretizations of Gaussian curvature are not in the required form and may be not consistent in the following sense.

Definition 4.1. *A set of approximations of differential geometric operators is said consistent if there exists a smooth surface $\mathcal{S}$, such that the approximate operators coincide with the exact counterparts of $\mathcal{S}$.*

Here we use a biquadratic fitting of the surface data and function data to calculate the approximate differential operators. The algorithm we adopted is from [18]. Let $\mathbf{r}_i$ be a vertex of M with valence n, $\mathbf{r}_j$ be its neighbor vertices for $j \in N_1(i)$. Then approximations of the used differential operators are represented as (see [18] for detail)

$$\nabla f(\mathbf{r}_i) \approx \textstyle\sum_{j \in N_1(i)} w_{ij}^{\nabla} f(\mathbf{r}_j), \qquad \diamond f(\mathbf{r}_i) \approx \textstyle\sum_{j \in N_1(i)} w_{ij}^{\diamond} f(\mathbf{r}_j),$$
$$\Delta f(\mathbf{r}_i) \approx \textstyle\sum_{j \in N_1(i)} w_{ij}^{\triangle} f(\mathbf{r}_j), \qquad K(\mathbf{r}_i) \approx \textstyle\sum_{j \in N_1(i)} (w_{ij}^{K})^T \mathbf{r}_j,$$

where w_{ij}^{∇}, $w_{ij}^{\diamond}$, $w_{ij}^{K} \in \mathbb{R}^3$ and $w_{ij}^{\triangle} \in \mathbb{R}$. Using the relation (2.1), we have

$$\mathbf{H}(\mathbf{r}_i) \approx \frac{1}{2} \sum_{j \in N_1(i)} w_{ij}^{\triangle} \mathbf{r}_j, \quad H(\mathbf{r}_i) \approx \frac{1}{2} \sum_{j \in N_1(i)} w_{ij}^{\triangle} \mathbf{n}(\mathbf{r}_i)^T \mathbf{r}_j.$$

we present a semi-implicit scheme, which leads to a linear system of equations. The basic idea for forming the linear equations is to decompose each of the terms of (3.1) as a product of a linear term and a remaining term. The linear term is discretized using the discretized differential operator. The remaining term is computed from previous approximation of the surface. Specifically, the terms of the equation (3.8) are approximated as follows:

$$
\begin{aligned}
&\frac{\partial \mathbf{r}}{\partial t} \approx \frac{\mathbf{r}_i^{(k+1)} - \mathbf{r}_i^{(k)}}{\tau},\\
&\Delta^2 H \approx \Delta(\Delta H_i^{(k+1)}),\\
&2(2H^2 - K)\Delta H \approx (2H_i^{(k+1)} H_i^{(k)} - K_i^{(k+1)})\Delta H_i^{(k)} + [2(H_i^{(k)})^2 - K_i^{(k)}]\Delta H_i^{(k+1)},\\
&2\langle \nabla H, \diamond H\rangle \approx \langle \nabla H_i^{(k+1)}, \diamond H_i^{(k)}\rangle + \langle \nabla H_i^{(k)}, \diamond H_i^{(k+1)}\rangle,\\
&2H\langle \nabla H, \nabla H\rangle \approx H_i^{(k+1)} \langle \nabla H_i^{(k)}, \nabla H_i^{(k)}\rangle + H_i^{(k)} \langle \nabla H_i^{(k+1)}, \nabla H_i^{(k)}\rangle,
\end{aligned}
$$

where τ is the time step-size, the subscript i denotes the corresponding quantity is evaluated at the vertex $\mathbf{r}_i$, the superscript (k) denotes the quantity is at the time $k\tau$, the superscript $(k+1)$ denotes the quantity is at the time $(k+1)\tau$. The quantities at $(k+1)\tau$ are unknowns. Using these approximations, we can discretize the equation(3.8) recursively, and derive a linear system with $\mathbf{r}_i^{(k+1)}$ as unknowns. For instance,

$$
\begin{aligned}
\mathbf{n}_i \Delta(\Delta H_i^{(k+1)}) &\approx \mathbf{n}_i \sum_{j\in N_1(i)} w_{ij}^{\Delta} \Delta H_j^{(k+1)}\\
&\approx \sum_{j\in N_1(i)} w_{ij}^{\Delta} \sum_{l\in N_1(j)} w_{jl}^{\Delta} (\mathbf{n}_i \mathbf{n}_l^T) \mathbf{H}_l^{(k+1)}\\
&\approx \frac{1}{2} \sum_{j\in N_1(i)} w_{ij}^{\Delta} \sum_{l\in N_1(j)} w_{jl}^{\Delta} (\mathbf{n}_i \mathbf{n}_l^T) \sum_{m\in N_1(l)} w_{lm}^{\Delta} \mathbf{r}_m^{(k+1)}.
\end{aligned}
$$

Similarly,

$$
\begin{aligned}
\mathbf{n}_i \langle \nabla H_i^{(k+1)}, \diamond H_i^{(k)}\rangle &\approx \sum_{j\in N_1(i)} \langle w_{ij}^{\nabla}, \diamond H_i^{(k)}\rangle \mathbf{n}_i H_j^{(k+1)}\\
&\approx \sum_{j\in N_1(i)} \langle w_{ij}^{\nabla}, \diamond H_i^{(k)}\rangle (\mathbf{n}_i \mathbf{n}_j^T) \mathbf{H}_j^{(k+1)}\\
&\approx \frac{1}{2} \sum \langle w_{ij}^{\nabla}, \diamond H_i^{(k)}\rangle (\mathbf{n}_i \mathbf{n}_j^T) \sum w_{jl}^{\Delta} \mathbf{r}_l^{(k+1)},
\end{aligned}
$$

vertex $\mathbf{r}_i$ and finally derive a linear equation. This equation is a linear combination of the three-ring neighbor vertices of $\mathbf{r}_i$.

$$\mathbf{r}_i^{(k+1)} + \tau \sum_{j \in N_3(i)} w_{ij} \mathbf{r}_j^{(k+1)} = \mathbf{r}_i^{(k)}, \quad w_{ij} \in \mathbb{R}^{3\times 3}.$$

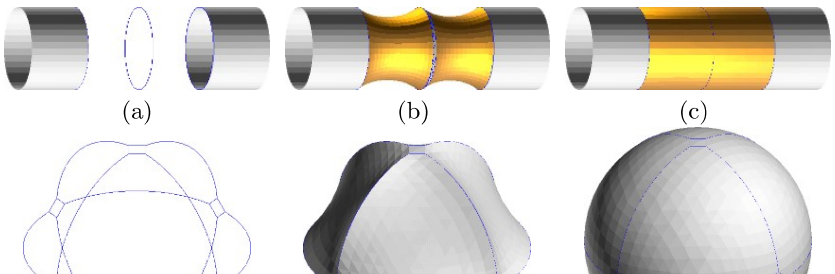
If an involved vertex $\mathbf{r}_j^{(k+1)}$ is not an interior one, $\mathbf{r}_j^{(k+1)} = \mathbf{r}_j$ is fixed and the term $\tau w_{ij} \mathbf{r}_j^{(k+1)}$ is moved to the right hand side of the equation. Such a treatment of the boundary condition leads to a system of n equations with n unknowns. Here n is the number of interior vertices. The idea of this boundary treatment is adopted from [20].

Solving the linear system. The result system is highly sparse. An iterative approach for solving the system is desirable. We employ Saad's iterative method (see [13]), named GMRES to solve the system. The experiments show that this iterative method works very well.

Remark 4.2. The experiments show that the semi-implicit discretization scheme proposed equipped with Saad's solver of the linear system is very stable. The time step-size could be oftentimes chosen fairly large (see Table 5.1).

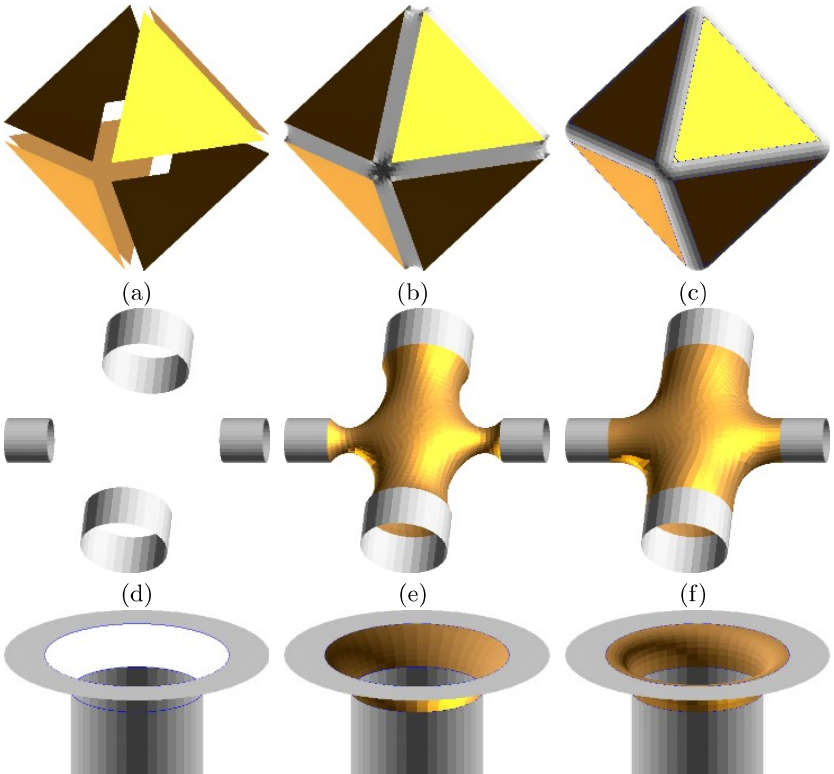
5 Illustrative Examples

Recover property to some surfaces. We have mentioned that constant mean curvature surfaces are the solutions of equation (3.1). Fig. 5.1 is used to illustrate that constant mean curvature surfaces can be recovered from their perturbed counterparts by MMCVF. The test is performed as follows. We first replace



certain parts of a given constant mean curvature surface with another surface, and then we use our geometric flow to evolve the surface.

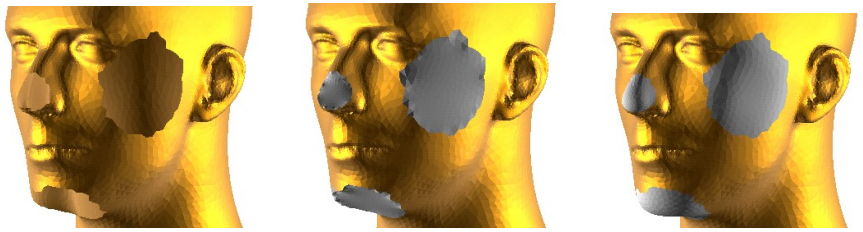
The first row of Fig. 5.1 shows that a cylinder is recovered, where (a) is a cylinder with certain parts missing. Figure (b) shows the minimal surface filling of the missing parts. This minimal surface acts as an initial surface $\mathcal{M}_0$ for the geometric flow. (c) shows the evolution result. It can be seen that the cylinder is correctly recovered. The second row of Fig. 5.1 shows that a sphere is recovered, where (d) is a wire-frame of a sphere with eight openings. These openings are filled with minimal surfaces as shown in (e). These minimal surfaces act as an initial surface $\mathcal{M}_0$ of the geometric flow. (f) shows the evolution result. It is easy to see that the sphere is perfectly recovered.



Smooth blending of surfaces. Given a collection of surface meshes with boundaries, we construct a fair surface to blend smoothly the meshes at the boundaries. Fig 5.2 shows the case, where surfaces to be blended are given (figure (a), (d) and (g)) with initial minimal surface constructions (figure (b), (e) and (h)) using [1] and then mean curvature flow. The surfaces (c), (f) and (i) are the blending meshes generated using our sixth-order flow.

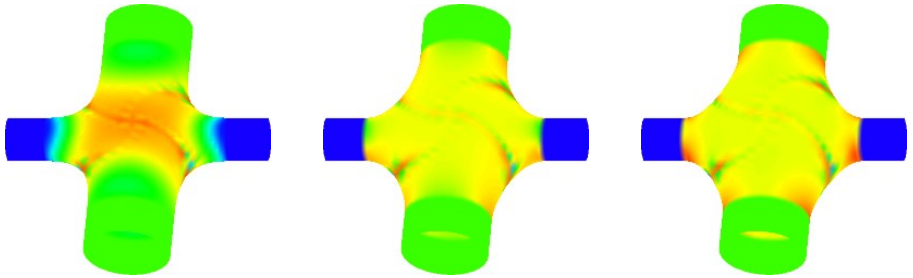
N-sided hole filling. Given a surface mesh with holes, we construct a fair surface to fill smoothly the holes with G^2 continuity on the boundary. Fig 5.3 shows such an example, where a head mesh with several holes in the nose, face and jaw subregions is given as input (figure (a)). An initial G^0 filler of the holes are shown in (b) using [1] and then evolved with the mean curvature flow. The blending surface (c) is generated using flow (3.8).

Point interpolation. For the point interpolation problem, we are given some points as the input data, and we wish to construct a fair surface mesh to interpolate this multi-dimensional data. Fig. 5.4 shows this surface construction approach, where a dodecahedron is served as input as shown in figure (a). The constructed surface is required to interpolate the vertices of the input polygon. Each face of the input polygon is triangulated by subdividing the 5-sided face



(a) (b) (c)

Fig. 5.3. (a) shows a head mesh with several holes. (b) shows an initial filler construction. (c) is the smooth filling surface, after 50 iterations, generated by using equation (3.8).

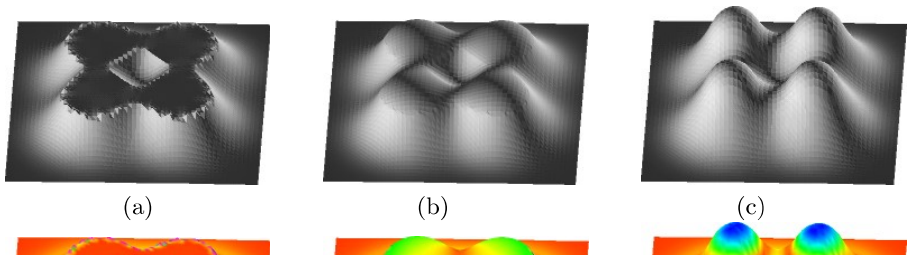


(a) (b) (c)

Fig. 5.5. (a), (b) and (c) are the mean curvature plots of the evolution results of MMCVF, SDF and WF, respectively.

into three triangles. Then each triangle is subdivided into 64 sub-triangles. The GPDE is applied to the triangulated polygon with the input vertices fixed. (b) is the evolution result of MMCVF. (c) shows an intermediate result of the evolution.

Comparison with lower order flows. Now we compare the used sixth-order flow MMCVF with three well-known lower order flows (see [20]): mean curvature flow (MCF), surface diffusion flow (SDF) and Willmore flow (WF). From the definition of MMCVF, we know that the main difference of MMCVF from the lower order flows is that the former yields G^2 and mean curvature uniformly distributed surfaces. Fig. 5.5 shows the evolution results of the sixth- and fourth-order flows for the input (d) of Fig. 5.2, where (a), (b) and (c) show the mean curvature plots of the evolution results of the MMCVF, SDF and WF, respectively. From these figures, we can observe that the surface produced by MMCVF is mean-curvature continuous at the blending boundaries, while the surfaces produced by SDF and WF are not.



(a) (b) (c)

Table 5.1. Running Times

Examples	# unknowns	Time step-size	Form matrix	# steps	Solving Time
Fig 5.1(c)	3480	1.0	0.12	100	2.11
Fig 5.1(f)	11160	1.0	0.42	350	6.72
Fig 5.2(c)	3222	1.0	0.12	5	1.62
Fig 5.2(f)	4410	1.0	0.17	50	2.75
Fig 5.2(i)	2400	0.01	0.09	10	1.43
Fig 5.3(c)	1296	0.5	0.05	10	1.26
Fig 5.6(c)	2520	2.0	0.09	5	1.50

In Fig. 5.6, the surface to be evolved is defined as a graph of a function g: $\mathbf{x}(u,v) = [u, v, g(u,v)]^T$, $g(u,v) = e(u,v)+e(u+1,v)+e(u,v+1)+e(u+1,v+1)$, with $(u,v) \in \Omega := [-1,1]^2$ and $e(u,v) = \exp[-\frac{81}{16}(u-0.5)^2 + (v-0.5)^2]$. This surface is uniformly triangulated using a 60×60 grid over the domain Ω. We evolve a part of the surface, where $g > 1.5$. Figures (a), (b) and (c) show the results of MCF, WF and MMCVF, respectively. Figures (d), (e) and (f) are the mean curvature plots of (a), (b) and (c), respectively. It is easy to see that the second and the fourth-order flows are not curvature continuous at the boundaries of the evolved surface patch.

Running Times. We summarize in Table 5.1 the computation time needed by some of our examples. The algorithm was implemented in C++ running on a Dell PC with a 3.0GHz Intel CPU. All the examples presented in this section are the approximate steady solution ($t \to \infty$). Hence, the total time costs depend greatly on how far we go in the time direction, which in turn depend on how far the initial surface away from the final solution. In Table 5.1, we list the costs for a single iteration. The second column in Table 5.1 is the number of unknowns. These numbers are counted as $3n_0$ (each vertex has x, y, z variables). Here n_0 is the number of interior vertices. The third column is the used time step-size. The fourth column in the table is the time (in seconds) for forming the coefficient matrix. The fifth column is the number of evolution steps. The last column is the time for solving the linear systems for one time step.

6 Conclusions

We have derived a sixth-order nonlinear geometric flow from the functional $\mathscr{F} = \int_M \|\nabla H\|^2 dA$. We name it as minimal mean-curvature-variation flow.

References

1. C. Bajaj and I. Ihm. Algebraic surface design with Hermite interpolation. *ACM Transactions on Graphics*, 11(1):61–91, 1992.
2. G. P. Bonneau, H. Hagen, and St. Hahmann. Variational surface design and surface interrogation. *Computer Graphics Forum*, 12(3):447–459, 1993.
3. M. Botsch and L. Kobbelt. An intuitive framework for real-time freeform modeling. *ACM Transaction on Graphics*, 23(3):630–634, 2004. Proceedings of the 2004 SIGGRAPH Conference.
4. H. Du and H. Qin. Dynamic PDE-based surface design using geometric and physical constraint. *Graphical Models*, 67(1):43–71, 2005.
5. A. Gray. *Modern Differential Geometry of Curves and Surfaces with Mathematica*. CRC Press, second edition, 1998.
6. G. Greiner. Variational design and fairing of spline surface. *Computer Graphics Forum*, 13:143–154, 1994.
7. M. Kallay. Constrained optimization in surface design. In B. Falcidieno and T. L. Kunii, editors, *Modeling in Computer Graphics*, pages 85–93. Springer-Verlag, Berlin, 1993.
8. E. Kuwert and R. Schätzle. The Willmore flow with small initial energy. *J. Differential Geom.*, 57(3):409–441, 2001.
9. H. B. Lawson. *Lectures on Minimal Submanifolds*. Publish or Perish, Berkeley, CA, 1980.
10. U. F. Mayer. Numerical solutions for the surface diffusion flow in three space dimensions. *Comput. Appl. Math.*, 20(3):361–379, 2001.
11. H. P. Moreton and C. H. Séquin. Functional optimization for fair surface design. *SIGGRAPH'92 Conference Proceedings*, pages 167–176, 1992.
12. J. Oprea. *Differential Geometry and Its Applications*. Pearson Education, Inc., second edition, 2004.
13. Y. Saad. *Iterative Methods for Sparse Linear Systems*. Second Edition with corrections, 2000.
14. R. Schneider and L. Kobbelt. Geometric fair meshes with G^1 boundary conditions. In *Geometric Modeling and Processing*, pages 251–261, 2000. Hongkong, China.
15. W. Welch and A. Witkin. Variational surface modeling. *Computer Graphics*, 26:157–166, 1992.
16. T. J. Willmore. *Riemannian Geometry*. Clarendon Press, Oxford, England, 1993.
17. G. Xu. Discrete Laplace-Beltrami operators and their convergence. *Computer Aided Geometric Design*, 21(8):767–784, 2004.
18. G. Xu. Consistent approximation of some geometric differential operators. Research Report No. ICM-06-01, Institute of Computational Mathematics, Chinese Academy of Sciences, 2006.
19. G. Xu. Convergence analysis of a discretization scheme for Gaussian curvature