A Level Set Formulation of Geodesic Curvature Flow on Simplicial Surfaces

Chunlin Wu, Xuecheng Tai

Abstract-Curvature flow (planar geometric heat flow) has been extensively applied to image processing, computer vision and material sciences. To extend the numerical schemes and algorithms of this flow on surfaces is very valuable for corresponding motion of curves and images defined on surfaces. In this work we are interested in geodesic curvature flow over triangulated surfaces using a level set formulation. We at first present geodesic curvature flow equation on general smooth manifolds via curve energy minimization. The equation is then discretized by semiimplicit FVM (finite volume method). For convenience of description, we call the discretized geodesic curvature flow as dGCF. Existence and uniqueness of the discrete problem dGCF are discussed. Regularization behavior of dGCF will also be studied. Finally we apply our dGCF to three problems: closed curve evolution on manifolds, discrete scale-space construction and edge detection of images painted on triangulated surfaces. Our method works for compact triangular meshes of arbitrary geometry (as long as there's no degenerate triangles) and topology. The implementation of the method is also easy.

Index Terms—geodesic curvature flow, level set, triangular mesh surfaces, curve evolution, scale-space, edge detection.

I. INTRODUCTION

The geodesic curvature flow, also known as geometric heat flow or curve shortening flow, has been studied for many years in both pure and applied mathematics. Therein closed curve evolution under geodesic curvature flow is an important tool [15] of closed geodesic theory, which is a fundamental part of Riemannian geometry. It also gains much attention in practice and is extensively applied via level set formulations [31] to image processing, computer vision and material science, *etc*.

The curve shortening flow on Euclidean plane is widely studied both theoretically and practically. It has been proven in theory that every simple closed curve (either convex or nonconvex) shrinks into a circular point in a finite time [10], [12], [14]. In applications, the flow is usually written in level set formulations or so called Eulerian framework and, has made great successes in the following problems. In multi-phase physical simulations and material sciences, this flow plays an important role for topology adaptive front propagation with curvature-dependent speed; see [31] and references therein. In image processing, people establish morphological scale-space [18] of an image via curve shortening flow. Scale-space is a fundamental concept for multi-scale representations and analysis in image processing, and has been extensively studied in recent years with many applications such as image compression, transmission, segmentation, feature detection and objects retrieval. The basic idea for scale-space construction is to introduce a family of images which progressively become simpler in a sense that significant structures remain while unimportant details vanish. So far people have derived various scalespaces [40], [20], [1], [43], [16], [32], [36], [18], [38], [37], [39], [6] from convolution method or time-dependent PDEs (the most popular way in this area), etc. Properties of these scalespaces have been carefully analyzed in [20], [43], [18], [23], [24], [25], [26], [34], [37], [38], [9], [21], [22], including the existence and uniqueness, continuity dependency on initial data, grey level shift invariance, semi-group property, average grey invariance and critical point theory, information reduction and limit behavior, etc. Some basic properties such as existence and uniqueness, semigroup property and information reduction, hold in all scale-spaces. Different scale-spaces may satisfy different other properties. In morphological scale-space (based on curvature flow) [18], the image is regarded as a collection of iso-intensity curves. Evolving the image is just evolving all the level curves. As shown by Grayson, the order of these level curves is preserved during the scale-space evolution. This leads to the so called inclusion order preserving [8], [18], the most advantage over other scale-spaces, which is particularly convenient for shape and image analysis. In computer vision, by introducing image content related weights to the curve length functional, geodesic active contour models as small variations of the curve shortening flow are proposed in [28], [3] for shape modelling, image edge detection and segmentation; see [13] for fast algorithms and references therein. Compared with other techniques such as the very similar snake model [17], geodesic active contour model by level set formulation has many advantages, such as topology adaption (automatic curve splitting and merging) which are very important in image segmentation. In addition, it drives automatically the contour to the center of the band of the image edges regardless of the diversity of the image gradient at edges [3]. This property makes the method available for not only general images, but also blurred images.

However, geodesic curvature flow on surfaces is much more complicated, and the theoretical results on this topic are very sparse; see [11], [27], [30]. Even the existence and uniqueness of solutions are proved conditionally. And possible shapes of the final curve are diverse. Unlike the planar case, a closed initial curve on a manifold will evolve into one of two shapes under the geodesic curvature flow. It may disappear or become closed geodesic(s). Moreover, geodesics on manifolds can be stable or unstable [30]. On the other hand, geodesic curvature flow has many applications in curve and image motion on surfaces [19], [33]. Therefore, to numerically compute the flow on manifolds will be very valuable for, not only simulating some properties of the flow [5], [4], but also extensive applications [19], [33]. In [5] the authors studied this flow by using standard methods for manifolds, i.e., cutting the interested manifold into a set of charts and solving geodesic curvature flows on these charts separately. This is a strategy based on piecewise parametrization. Various

Manuscript received, revised.

Chunlin Wu, Division of Mathematical Science, School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore. Xuecheng Tai, Division of Mathematical Science, School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore and Department of Mathematics, University of Bergen, Norway.

possible cases of final curves were illustrated. Under the level set framework, curve evolution over implicit surfaces was well studied by Cheng *et.al.* in [4]. In Cheng's work, both the surface and the curve are represented as level sets of functions defined in \mathbb{R}^3 and, the flow equations are solved in a narrow band of 3D Cartesian grids near the surface. In [19], Kimmel studied geodesic curvature flow on parametric manifolds with applications in image processing. He proposed bending invariant scale-space concepts of images painted on surfaces via geodesic curvature flow and presented numerical methods to solve the problem. The technique can be applied to post processing of texture mapping. This flow with even more applications on parametric surfaces was reported in [33] very recently.

The novel contributions made in this paper are in three aspects. (1) We derive the numerical scheme for geodesic curvature flow on triangulated surfaces using level set formulation, leading to dGCF. Our method works for any compact triangular meshes as long as there's no degenerate triangles. (2) The discretized flow dGCF is theoretically analyzed with a rigorous proof on its existence and uniqueness, as well as a discussion on the regularization behavior by eigen analysis. In contrast, for geodesic curvature flow in continuous setting, even the existence and uniqueness are proved conditionally. (3) We apply our dGCF to curve evolution, scale-space calculation and edge detection of images painted on triangular meshes. In the 1st application, we present an algorithm to find stable closed geodesics on meshes. To our knowledge, this problem has not been investigated so far, although there are lots of publications (see [35] and references therein) on computing geodesic pathes between given points on meshes. In the 2nd application, the scale-space calculated via geodesic curvature flow behaves very differently from some other smoothing techniques such as Laplacian smoothing [41]. It provides much clearer multi-scale representations of images, which seems retaining the inclusion order preserving property of planar geometric heat flow. We also observe that the limit behavior of this scale-space depends on the scale parameter, which is much more complicated than Laplacian smoothing. Although we cannot prove the limit behavior rigorously at this stage, we interpret this phenomenon by the results of the 1st application, by using the grey level shift invariance property. These observations have not been noticed before. In the 3rd application, our method can capture image edges with various complicated structures, even if the edges are blurred and contain multiple connected components. It should be pointed out that the (mean) curvature flow with no weight was addressed in [2] for triangulated domains. However, all the examples in [2] are for curves on planar Euclidean domains or surfaces in spatial Euclidean domains. Therefore, in [2] the pure (mean) curvature flow was computed, and not the geodesic one. Furthermore, there's little theoretical discussion on their discretized flow in [2].

Here we emphasize the differences between our method and those reported in [41]. Basically the geodesic curvature flow equation describes geometric evolution of curves, whose mechanism is totally different from diffusion equations studied in [41]. Hence the numerical scheme is naturally different from those in [41]. Consequently the method presented in this paper has many different applications, i.e., curve evolution and edge detection on manifolds. The approaches in [41] cannot handle these two applications, because we have proved in [42] that the diffusion models in [41] satisfy constant limit behavior. That is to say, using these models, the flow function will tend to a constant finally, which is impossible to capture the geodesics. In addition, even in the same application of our method and [41], say, scalespace image evolution, geodesic curvature flow behaves much differently from diffusion equations, and offers the advantage of inclusion order preserving.

The outline of the paper is as follows. We first give some compulsory notation in Section 2. Then in Section 3 we review the flow equation on general surfaces using a weighted curve energy minimization. The equation is discretized via semi-implicit FVM in Section 4, leading to our dGCF. We will also in this section prove the existence and uniqueness, and study the regularization behavior of the discretized flow, followed by a comparison to Laplacian smoothing. In Section 5 we discuss three applications of dGCF. Conclusion and future work are included in Section 6.

II. NOTATION

We need some notation; see also [41]. Assume $M \subset \mathbb{R}^3$ is a compact triangulated surface of arbitrary topology with no degenerate triangles. The set of vertices, the set of edges and the set of face triangles of M are denoted as $\{v_i : i = 0, 1, \dots, V-1\},\$ $\{e_i : i = 0, 1, \cdots, E - 1\}$ and $\{\tau_i : i = 0, 1, \cdots, T - 1\},\$ respectively. Here V, E and T are the numbers of vertices, edges and triangles, respectively. We explicitly denote an edge e whose endpoints are p, q as [p, q]. Similarly a triangle τ whose vertices are p, q, r is denoted as [p, q, r]. If v is an endpoint of an edge e, then we denote it as $v \prec e$. Similarly, e is an edge of a triangle τ is denoted as $e \prec \tau$; v is a vertex of a triangle τ is denoted as $v \prec \tau$. We use $BC(\tau)$, BC(e) and BC(v) to denote the barycenters of a triangle τ , an edge e and a vertex v, respectively. Let $N_1(i)$ be the 1-neighborhood of vertex v_i , which is the set of indices of vertices connecting to v_i . Let $D_1(i)$ be the 1-disk of the vertex v_i . $D_1(i)$ is the set of triangles containing v_i .

We also introduce the concepts of dual meshes and dual cells; see Fig. 1. For any triangular mesh surface, a *barycentric dual* is formed by connecting the midpoint of each edge with the barycenters of each of its incident faces, as illustrated in Fig. 1 (a). The original mesh consists of black lines while the dual mesh is in blue. In a dual mesh, one assigns a *dual cell* C_i to each vertex v_i of mesh surface M ([29]). The dual cell of a vertex v_i is part of its 1-disk which is near to v_i in the dual mesh. Fig. 1 (b) shows the dual cell C_i to each vertex.

For each vertex v_i , we denote the usual hat function as φ_i , which is linear over each triangle and $\varphi_i(v_j) = \delta_{ij}$, $i, j = 0, 1, \dots, V - 1$, where δ_{ij} is the Kronecker delta. Functions $\{\varphi_i : i = 0, 1, \dots, V - 1\}$ satisfy

- 1) local support: supp $\varphi_i = D_1(i)$;
- 2) nonnegativity: $\varphi_i \ge 0, i = 0, 1, \cdots, V 1;$

B) partition of unity:
$$\sum_{0 \le i \le V-1} \varphi_i \equiv 1.$$

A function u defined over the triangulated surface M is considered to be a piecewise linear function. Suppose u reaches value u_i at vertex v_i , $i = 0, 1, \dots, V-1$. Then $u(p) = \sum_{\substack{0 \le i \le V-1}} u_i \varphi_i(p)$ for any $p \in M$. Accordingly, piecewise linear vector-valued functions $(u_1(p), u_2(p), \dots, u_d(p))$ on M can be defined. Besides, one may have piecewise constant scalar (vector-valued) functions over M, that is, a single value (vector) is assigned to each triangle of M.



(c) dual cell of a boundary vertex (d) computation of $c_{ij,\tau}, c_{jk,\tau}$ and $c_{ki,\tau}$

Fig. 1. Barycentric dual mesh, dual cells and computation of coefficients.

III. GEODESIC CURVATURE FLOW OVER SMOOTH SURFACES USING LEVEL SET FORMULATION

In this section, we present geodesic curvature flow over general smooth surfaces using level set formulation via a weighted curve energy minimization.

Assume \mathcal{M} is a general 2-dimensional manifold embedded in \mathbb{R}^3 and ∇ is the intrinsic gradient operator on \mathcal{M} . Suppose that $C \subset \mathcal{M}$ is a curve defined on \mathcal{M} , and represented by the zero level set of a function $\phi : \mathcal{M} \to \mathbb{R}$; see Fig. 2 (just a local view) for illustration of geodesic curvature of C and the usual geodesic curvature flow (with no weight).

Here we consider a more general geodesic curvature flow which decreases a weighted curve length, by using level set formulation. Assume that $g: \mathcal{M} \to \mathbb{R}^+$ is a positive scalar function serving as a weight (or stopping function). Then the weighted length of C is defined as

$$E(C) = \int_{C:\phi=0} g dl,$$
(1)

which, by the Co-Area Formula, can be reformulated into

$$E(C) = E(\phi) = \int_{\mathcal{M}} g\delta(\phi) |\nabla\phi| d\mathcal{M}, \qquad (2)$$

an energy functional of $\phi,$ where $\delta(\cdot)$ is the Dirac function.

Similarly with [4], we derive the Euler-Lagrange equation for the energy functional (2) as

$$\begin{cases} -\nabla \cdot (g \frac{\nabla \phi}{|\nabla \phi|}) \delta(\phi) = 0\\ \frac{\partial \phi}{\partial \vec{n}}|_{\partial \mathcal{M}} = 0 \end{cases},$$
(3)

where $\partial \mathcal{M}$ is the boundary of \mathcal{M} and \vec{n} is the intrinsic outer normal of $\partial \mathcal{M}$. For closed \mathcal{M} , the boundary condition is automatically ignored.

Just as in the standard level set method, replacing $\delta(\phi)$, which is treated as the smoothed out delta function, by $|\nabla \phi|$, we obtain the following gradient descent flow equation

$$\begin{cases} \phi_t = |\nabla \phi| \nabla \cdot (g \frac{\nabla \phi}{\sqrt{|\nabla \phi|^2 + \beta}}) \\ \frac{\partial \phi}{\partial \vec{n}}|_{\partial \mathcal{M}} = 0 \\ \phi(0) = \phi_0 \end{cases}, \tag{4}$$

where β is a small positive number introduced to avoid division by zero. This flow on implicit (with g = 1) and parametric manifolds was investigated in [4] and [33], respectively.



Fig. 2. The curvature vector $k\mathbf{N}$ of curve C on manifold \mathcal{M} has two orthogonal components: the normal curvature vector $k_n \mathbf{\hat{N}}$ and the geodesic curvature vector $k_g \mathbf{\tilde{N}}$. The usual geodesic curvature flow (with no weight) decreases the length of C, say, $\int_C dl$. It reads $C_t = k_g \mathbf{\tilde{N}}$ in Lagrangian framework, and $\phi_t = |\nabla \phi| \nabla \cdot (\frac{\nabla \phi}{|\nabla \phi|})$ in Eulerian framework. Similarly with the curve evolution on planar domain, Lagrangian framework handles both open and closed curves quite well, but suffers from the difficulty of changing the topology of the curve; Eulerian framework works particularly well for closed curve evolution and benefits from its flexible topology adaptivity, but has difficulty for open curves. There's also one other important difference between these two frameworks for curve evolution on surfaces. In Lagrangian framework, each point of the curve moves on the tangent plane of the surface. Hence in numerical computation (the manifold is also discretized), one should be very careful to avoid the points of the curve to go out of the surface. However, Eulerian framework does not suffer from this problem.

IV. DISCRETIZED GEODESIC CURVATURE FLOW OVER TRIANGULATED SURFACES: DGCF

We now assume that \mathcal{M} is triangulated to be $M \subset \mathbb{R}^3$. Then we come to the discrete setting; see Section 2. The function ϕ is piecewise linear, which interpolates function values at vertices of M; and the weight g is piecewise constant as face samples. Under these assumptions, the curve C as the zero level set of ϕ is also piecewise linear and represented as line segments. In this section we give our dGCF via semi-implicit FVM discretization of Eqn. (4). The existence and uniqueness of the discretized flow will be proven. We also discuss the regularization behavior of dGCF, as well as its differences from Laplacian smoothing.

A. Discretized geodesic curvature flow (dGCF)

We discretize the flow equation (4) via semi-implicit FVM: semi-implicit time discretization and spatial discretization by integrating the PDE over some dual cells. This is a linearization technique where only some of the variables in the nonlinear term are evaluated at the next time step, leading to finally a linear system for unknowns.

For each vertex v_i of M, we integrate the two sides of Eqn. (4) on the dual cell C_i :

$$\int_{C_i} \phi_t dA = \int_{C_i} |\nabla \phi| \nabla \cdot (g \frac{\nabla \phi}{\sqrt{|\nabla \phi|^2 + \beta}}) dA.$$
 (5)

Because of the existence of $|\nabla \phi|$ outside the divergence operator, the integral in the right hand side of (5) can not be directly

calculated. Our strategy is firstly approximating $|\nabla \phi|$ outside the divergence operator by a constant over the dual cell. Since barycentric dual mesh is used, the average over the dual cell is just the average over the 1-Disk. Therefore,

$$(|\nabla \phi|)_i = \frac{\sum\limits_{\tau, v_i \prec \tau} |\nabla \phi|_{\tau} |s_{\tau}}{\sum\limits_{\tau, v_i \prec \tau} s_{\tau}}$$

where s_{τ} is the area of the triangle τ . This is a little different variant of the approach used in [2]. (We cannot directly adopt the method in [2] due to that the triangles of the 1-Disk are not coplanar in general.) In this way, we have

$$\int_{C_i} \phi_t dA = \frac{\sum\limits_{\tau \in D_1(i)} |\nabla \phi|_{\tau} |s_{\tau}|_{s_{\tau}}}{\sum\limits_{\tau \in D_1(i)} s_{\tau}} \int_{C_i} \nabla \cdot (g \frac{\nabla \phi}{\sqrt{|\nabla \phi|^2 + \beta}}) dA.$$
(6)

We then apply the divergence theorem to the integral in the right hand side of Eqn. (6). Note that g is a piecewise constant function. With a similar derivation as in [41], we have

$$\int_{C_i} \nabla \cdot (g \frac{\nabla \phi}{\sqrt{|\nabla \phi|^2 + \beta}}) dA = \sum_{\tau = [v_i, v_j, v_k] \in D_1(i)} \frac{g|_{\tau}}{\sqrt{|\nabla \phi|_{\tau}|^2 + \beta}} (\phi_i c_{ii,\tau} + \phi_j c_{ij,\tau} + \phi_k c_{ik,\tau}),$$
(7)

where

$$\begin{cases} c_{ij,\tau} = \nabla \varphi_j \cdot \int_{\substack{e,v_i \prec e \prec \tau \\ e,v_i \prec e \prec \tau}} [BC(e), BC(\tau)] \vec{n} dl = \frac{1}{2} \cot \theta_k \\ c_{ik,\tau} = \nabla \varphi_k \cdot \int_{\substack{e,v_i \prec e \prec \tau \\ e,v_i \prec e \prec \tau}} [BC(e), BC(\tau)] \vec{n} dl = \frac{1}{2} \cot \theta_j \quad , \quad (8) \\ c_{ii,\tau} = -c_{ij,\tau} - c_{ik,\tau} \end{cases}$$

as shown in [41], [42]; see also Fig. 1 (d). This is a uniform spatial discretization for any compact mesh surfaces of arbitrary topology.

Thus with a semi-implicit time integral (from t^n to t^{n+1}), Eqn. (4) is discretized as follows

$$s_{i} \frac{\phi_{i}^{n+1} - \phi_{i}^{n}}{t^{n+1} - t^{n}} = \frac{\sum_{\tau \in D_{1}(i)} |\nabla \phi|_{\tau}^{n} |s_{\tau}}{\sum_{\tau \in D_{1}(i)} s_{\tau}} \cdot \sum_{\tau \in D_{1}(i)} \frac{g|_{\tau}}{\sqrt{|\nabla \phi|_{\tau}^{n}|^{2} + \beta}} (\phi_{i}^{n+1} c_{ii,\tau} + \phi_{j}^{n+1} c_{ij,\tau} + \phi_{k}^{n+1} c_{ik,\tau}),$$

where s_i is the area of the dual cell of v_i . Denoting $\Phi = (\phi_0, \phi_1, \cdots, \phi_{V-1})'$, the above equation is formulated in matrix form (with initial value added)

$$\begin{cases} (S + (t^{n+1} - t^n)G(\Phi(t^n))H(\Phi(t^n)))\Phi(t^{n+1}) = S\Phi(t^n) \\ \Phi(t^0) = \Phi_0 \end{cases},$$
(9)

where $S = \operatorname{diag}(s_0, s_1, \cdots, s_{V-1})$ and $G(\Phi(t^n)) = \operatorname{diag}(\frac{\nabla \phi|_{\tau}^n|s_{\tau}}{\sum_{\tau \in D_1(0)} s_{\tau}}, \frac{\nabla \phi|_{\tau}^n|s_{\tau}}{\sum_{\tau \in D_1(1)} s_{\tau}}, \cdots, \frac{\nabla \phi|_{\tau}^n|s_{\tau}}{\sum_{\tau \in D_1(V-1)} s_{\tau}})$ are two diagonal matrices and $H(\Phi(t^n)) = (-h_{ij})$ with

$$h_{ij} = \begin{cases} \sum_{\tau, [v_i, v_j] \prec \tau} \frac{g|_{\tau}}{\sqrt{|\nabla \phi|_{\tau}^n|^2 + \beta}} c_{ij,\tau}, & j \in N_1(i) \\ \sum_{\tau \in D_1(i)} \frac{g|_{\tau}}{\sqrt{|\nabla \phi|_{\tau}^n|^2 + \beta}} c_{ii,\tau}, & j = i \\ 0, & \text{others} \end{cases}$$
(10)

Definition 1: Under an arbitrary time sequence $t^n, n = 0, 1, 2, \cdots$ with $t^0 = 0$, the sequence $\{\Phi(t^n), n = 0, 1, 2, \cdots\}$ determined by Eqn. (9) is called a *dGCF* (discretized geodesic curvature flow) of initial function Φ_0 .

Note that the time steps need not be a constant. In the following subsections, we discuss some fundamental theoretical aspects of dGCF, as well as some comparisons to Laplacian smoothing.

B. The existence and uniqueness of dGCF

The existence and uniqueness of the discretized flow can be proved for any compact simplicial surfaces with no degenerate triangles (the coefficients defined in Eqn. (8) are bounded), which ensure our claim in the abstract. We need two lemmas.

Lemma 1: (1) Assume that D is a diagonal matrix with nonnegative elements and A is symmetric positive semi-definite. Then the principle minors of the matrix DA are all nonnegative. (2) Assume A is a V × V matrix and λ is a constant. Then

 $\det(\lambda I + A)$

$$=\lambda^{\mathcal{V}} + \sum_{1 \le k \le \mathcal{V}} \lambda^{\mathcal{V}-k} \sum_{i_1 < i_2 < \dots < i_k} A\left(\begin{array}{cccc} i_1 & i_2 & \dots & i_k \\ i_1 & i_2 & \dots & i_k \end{array}\right),$$
(11)

where *I* is an identity matrix and $A\begin{pmatrix} i_1 & i_2 & \cdots & i_k \\ i_1 & i_2 & \cdots & i_k \end{pmatrix}$ is a principle minor of *A*.

Proof (1) Let $D = \text{diag}(d_1, d_2, \dots, d_V)$. The assertion follows immediately from

$$(DA)\begin{pmatrix} i_1 & \cdots & i_r \\ i_1 & \cdots & i_r \end{pmatrix} = d_{i_1} \dots d_{i_r} A \begin{pmatrix} i_1 & \cdots & i_r \\ i_1 & \cdots & i_r \end{pmatrix}$$

for any $1 \leq i_1 < \cdots < i_r \leq V$.

(2) This can be proved in a similar way on pages 180-182 in [7].

Lemma 2: The matrix $H(\Phi(t^n))$ with elements defined via Eqn. (10) is symmetric and positive semi-definite with rank(H) = V - 1.

Proof The symmetry of $H(\Phi(t^n))$ is obvious since $c_{ij,\tau} = c_{ji,\tau}$. On the other hand, for any vector v, we have

$$\begin{split} v'H(\Phi(t^{n}))v &= \sum_{ij} -h_{ij}v_{i}v_{j} = -\sum_{i} h_{ii}v_{i}^{2} - \sum_{i} \sum_{j \in N_{1}(i)} h_{ij}v_{i}v_{j} \\ &= -\sum_{i} \sum_{\tau, \tau \in D_{1}(i)} \frac{g|\tau}{\sqrt{|\nabla \phi|_{\tau}^{n}|^{2} + \beta}} c_{ii,\tau}v_{i}^{2} \\ &- \sum_{i} \sum_{j \in N_{1}(i)} \sum_{\tau, [v_{i}, v_{j}] \prec \tau} \frac{g|\tau}{\sqrt{|\nabla \phi|_{\tau}^{n}|^{2} + \beta}} c_{ij,\tau}v_{i}v_{j} \\ &= \sum_{e = [v_{i}, v_{j}]} \sum_{\tau, e \prec \tau} \frac{g|\tau}{\sqrt{|\nabla \phi|_{\tau}^{n}|^{2} + \beta}} c_{ij,\tau}(v_{i}^{2} + v_{j}^{2}) \\ &- \sum_{e = [v_{i}, v_{j}]} \sum_{\tau, e \prec \tau} \frac{g|\tau}{\sqrt{|\nabla \phi|_{\tau}^{n}|^{2} + \beta}} c_{ij,\tau}(v_{i} - v_{j})^{2} \\ &= \sum_{e = [v_{i}, v_{j}, v_{k}]} \frac{g|\tau}{\sqrt{|\nabla \phi|_{\tau}^{n}|^{2} + \beta}} \cdot \\ &(c_{ij,\tau}(v_{i} - v_{j})^{2} + c_{ik,\tau}(v_{i} - v_{k})^{2} + c_{jk,\tau}(v_{j} - v_{k})^{2}) \\ &= \sum_{\tau = [v_{i}, v_{j}, v_{k}]} \frac{g|\tau}{\sqrt{|\nabla \phi|_{\tau}^{n}|^{2} + \beta}} \frac{1}{2} \cdot \end{split}$$

$$(\cot \theta_k (v_i - v_j)^2 + \cot \theta_j (v_i - v_k)^2 + \cot \theta_i (v_j - v_k)^2) = \sum_{\tau = [v_i, v_j, v_k]} \frac{g|_{\tau}}{\sqrt{|\nabla \phi|_{\tau}^n|^2 + \beta}} \int_{\tau} |\nabla v(p)|^2 d\tau \ge 0,$$

where v(p) is the piecewise linear interpolation of vertex data v on M. This proves the semi-positiveness of $H(\Phi(t^n))$. The positiveness of g implies rank(H) = V - 1.

Theorem 1: For any initial value Φ_0 , and any time sequence $\{t^n, n = 0, 1, 2, \dots\}$ with $t^0 = 0$, there exists a unique dGCF.

Proof In fact we only need to prove that the coefficient matrix of the system (9) is invertible. We compute the determinant of $(S + (t^{n+1} - t^n)G(\Phi(t^n))H(\Phi(t^n)))$. Let

$$\lambda_i = \frac{\sum\limits_{\tau \in D_1(i)} |\nabla \phi|_{\tau}^n |s_{\tau}|}{\sum\limits_{\tau \in D_1(i)} s_{\tau}}, i = 0, 1, \cdots, V - 1,$$

we have

$$\begin{aligned} \det(S + (t^{n+1} - t^n)G(\Phi(t^n))H(\Phi(t^n))) \\ &= \det(S + (t^{n+1} - t^n)\operatorname{diag}(\lambda_0, \lambda_1, \cdots, \lambda_{V-1})H(\Phi(t^n))) \\ &= \det S \cdot \det(I + (t^{n+1} - t^n)\operatorname{diag}(\frac{\lambda_0}{s_0}, \frac{\lambda_1}{s_1}, \cdots, \frac{\lambda_{V-1}}{s_{V-1}})H(\Phi(t^n))) \\ &:= \det S \cdot \det(I + \Lambda H) \\ &= \det S \cdot (1 + \sum_{1 \le k \le V} \sum_{i_1 < i_2 < \cdots < i_k} (\Lambda H) \left(\begin{array}{ccc} i_1 & i_2 & \cdots & i_k \\ i_1 & i_2 & \cdots & i_k \end{array} \right)) \end{aligned}$$

> 0,

by Lemmas 1 and 2. This proves the invertibility of the coefficient matrix of the system (9).

C. Regularization behavior of dGCF

In this subsection we discuss the regularization behavior of dGCF. As stated in the definition, in practice the dGCF is calculated step by step using a time sequence. Therefore its one step regularization behavior is of fundamental importance. Here our interpretation is based on eigenvalue/eigenvector analysis since we are in discrete setting. As an obvious result, the L_2 stability can be obtained directly via our analysis.

Theorem 2: For the flow function $\Phi(t^n)$, we assume that $K = \{i | \phi_j^n = \phi_i^n, j \in N_1(i)\}$ and $L = \{0, 1, \dots, V-1\} \setminus K$ are two index sets. Then $(I + \triangle tS^{-1}G(\Phi(t^n))H(\Phi(t^n)))^{-1}$ has eigenvalues $0 < \mu_0, \mu_1, \dots, \mu_{V-1} \leq 1$ with corresponding eigenvectors $\{b_i = (b_{i,0}, b_{i,1}, \dots, b_{i,V-1})', i = 0, 1, \dots, V-1\}$, which is complete. Moreover,

- If K is empty, then the largest eigenvalue μ_{max} = 1 with a unique eigenvector (1, 1, ..., 1);
- (2) If K is non-empty, then for all i ∈ K, μ_i = 1 is the eigenvalue with (1, 1, · · · , 1) as one of the corresponding eigenvectors; for i ∈ L, 0 < μ_i < 1 and b_{i,j} = 0, j ∈ K in its corresponding eigenvector b_i.

Proof Let

$$D = S^{-1}G(\Phi(t^n)),$$

which is a diagonal matrix with nonnegative elements.

(1) Since K is empty, it's obvious that G is invertible. We then have

$$S^{-1}G(\Phi(t^n))H(\Phi(t^n)) = DH = \sqrt{D}\sqrt{D}H$$

 $\sim (\sqrt{D})^{-1}\sqrt{D}\sqrt{D}H\sqrt{D} = \sqrt{D}H\sqrt{D},$

implying that DH is similar to symmetric $\sqrt{D}H\sqrt{D}$ by the symmetry of H. Therefore $I + \Delta tS^{-1}G(\Phi(t^n))H(\Phi(t^n))$ is similar to a symmetric matrix and hence its inverse. This shows that all the eigenvalues of $I + \Delta tS^{-1}G(\Phi(t^n))H(\Phi(t^n))$ are real and the set of eigenvectors is complete. On the other hand, by Lemmas 1 and 2,

$$\det(\lambda I - DH) = (-1)^{\mathrm{V}} \det((-\lambda)I + DH) = (-1)^{\mathrm{V}} \cdot \left((-\lambda)^{\mathrm{V}} + \sum_{1 \le k \le \mathrm{V}} (-\lambda)^{\mathrm{V}-k} \sum_{i_1 < i_2 < \cdots < i_k} DH \begin{pmatrix} i_1 & i_2 & \cdots & i_k \\ i_1 & i_2 & \cdots & i_k \end{pmatrix} \right) \neq 0, \text{ for } \lambda < 0.$$

This shows that all the eigenvalues of DH are greater than or equal to zero. Moreover, by Lemma 2 and the vanishing row sums of H, we know that the minimal eigenvalue of the matrix H is 0 with a unique eigenvector $(1, 1, \dots, 1)$. Therefore in this case the assertion follows.

(2) If K is non-empty, then G is not invertible. For convenience we use a series of permutations to relocate the zero diagonal elements of D together to the later part of the diagonal line. Denoting the element number of K as |K| = V - r, there exists an orthogonal matrix P (the product of a series of fundamental matrices exchanging two rows each) such that

$$D = P^{-1} \left(\begin{array}{cc} D_{11} & 0\\ 0 & 0 \end{array} \right) P,$$

where D_{11} is an $r \times r$ (r < V) diagonal sub-matrix with positive diagonal elements. Hence

$$S^{-1}G(\Phi(t^{n}))H(\Phi(t^{n})) = DH(\Phi(t^{n}))$$

= $P^{-1}\begin{pmatrix} D_{11} & 0\\ 0 & 0 \end{pmatrix} PH(\Phi(t^{n})) \sim \begin{pmatrix} D_{11} & 0\\ 0 & 0 \end{pmatrix} PH(\Phi(t^{n}))P$
:= $\begin{pmatrix} D_{11} & 0\\ 0 & 0 \end{pmatrix} \begin{pmatrix} B_{11} & B_{12}\\ B'_{12} & B_{22} \end{pmatrix} = \begin{pmatrix} D_{11}B_{11} & D_{11}B_{12}\\ 0 & 0 \end{pmatrix},$

where B_{11} is an $r \times r$ symmetric positive definite matrix from Lemma 2.

Let's analyze the matrix $\begin{pmatrix} D_{11}B_{11} & D_{11}B_{12} \\ 0 & 0 \end{pmatrix}$. By a similar argument as in (1), one can show that the eigenvalues of $D_{11}B_{11}$ are positive. Therefore we can denote all the eigenvalues by $\lambda_0, \lambda_1, \dots, \lambda_{r-1}, \lambda_r = \lambda_{r+1} = \dots = \lambda_{V-1} = 0$ with corresponding eigenvectors $e_0, e_1, \dots, e_{r-1}, e_r, \dots, e_{V-1}$. In the next we reveal structures of e_i and show that for each λ_i , its algebraic multiplicity equals to its geometric multiplicity. Consider the following system of equations

$$\begin{pmatrix} D_{11}B_{11} & D_{11}B_{12} \\ 0 & 0 \end{pmatrix} \begin{pmatrix} x_0 \\ \vdots \\ x_{r-1} \\ x_r \\ \vdots \\ x_{V-1} \end{pmatrix} = \lambda_i \begin{pmatrix} x_0 \\ \vdots \\ x_{r-1} \\ x_r \\ \vdots \\ x_{V-1} \end{pmatrix},$$
$$i = 0, 1, \cdots, V - 1.$$

By setting $x_r = x_{r+1} = \cdots = x_{V-1} = 0$, the eigenvalue λ_i and the corresponding eigenvector e_i for $i = 0, \ldots, r-1$ can be determined by the sub linear system of $D_{11}B_{11}$. Noticing that $D_{11}B_{11}$ is similar to a diagonal matrix, one gets that the algebraic multiplicity and geometric multiplicity of each λ_i are the same, i = 0, ..., r - 1. Also the later V – r elements of e_i vanish. For $\lambda_r = \lambda_{r+1} = \cdots = \lambda_{V-1} = 0$, its geometric multiplicity is

$$V - rank \begin{pmatrix} D_{11}B_{11} & D_{11}B_{12} \\ 0 & 0 \end{pmatrix} = V - r$$

which is exactly the algebraic multiplicity. Therefore the set of eigenvectors $\{e_i, i = 0, 1, \dots, V-1\}$ is complete.

Now we come back to $(I + \Delta t S^{-1} G(\Phi(t^n)) H(\Phi(t^n)))^{-1}$ for this case. Since $S^{-1} G(\Phi(t^n)) H(\Phi(t^n)) = DH = DH = P^{-1} \begin{pmatrix} D_{11}B_{11} & D_{11}B_{12} \\ 0 & 0 \end{pmatrix} P$, we immediately obtain the result by taking into account the structure of P and the vanishing row sums of H.

The following interpretation can be inferred from this theorem. Suppose the present flow function is $\Phi(t^n)$. Since the eigenvectors $\{b_i, 0 \le i \le V-1\}$ are complete, $\Phi(t^n)$ can be decomposed as

$$\Phi(t^n) = \sum_{0 \le i \le V-1} \alpha_i b_i, \tag{12}$$

and will evolve into

$$\Phi(t^{n+1}) = \sum_{0 \le i \le V-1} \mu_i \alpha_i b_i, \tag{13}$$

according to Theorem 2. As all the eigenvalues $\{\mu_i, 0 \le i \le V-1\}$ belong to (0, 1], the components $\{\alpha_i b_i, 0 \le i \le V-1\}$ in the decomposition (12) will never blow up. The components $\alpha_i b_i$ will be kept unchanged if $\mu_i = 1$; whereas other components $\alpha_i b_i$ will even shrink.

Let's investigate this more carefully. There are two cases. If K is empty, the zero frequency component along eigen-direction $(1, 1, \dots, 1)$ (a constant function on the surface M in function point of view) is the only one retained and other components with higher frequency will shrink. If K is non-empty, we rewrite Eqns. (12) and (13) as

$$\Phi(t^n) = \sum_{i \in K} \alpha_i b_i + \sum_{i \in L} \alpha_i b_i, \qquad (14)$$

and

$$\Phi(t^{n+1}) = \sum_{i \in K} \alpha_i b_i + \sum_{i \in L} \mu_i \alpha_i b_i,$$
(15)

where we use the result that $\mu_i = 1$ for $i \in K$. Eqn. (15) implies that components $\alpha_i b_i$ for $i \in K$ are preserved while other $\alpha_i b_i$ for $i \in L$ shrink. The preserved components may contain high frequency, not only the zero frequency along $(1, 1, \dots, 1)$. However, keep in mind where these components come from. We recall that $\Phi(t^n) = (\phi_0^n, \phi_1^n, \cdots, \phi_j^n, \cdots, \phi_{V-1}^n)'$. According to Theorem 2, in the decomposition of Eqn. (14), only the first summation $\sum_{i \in K} \alpha_i b_i$, in which components are fully preserved, contributes to $\phi_j^n, j \in K$. On the other hand, from the definition of K we know that these well preserved $\phi_i^n, j \in K$ should be piecewise constant. The shrinking components $\alpha_i b_i, i \in L$ have the supports in transition domains between piecewise constant regions of $\Phi(t^n)$ and serve as high frequency signals. Here we see a regularization effect: constant or piecewise constant components preserve, while high frequency components supported in transition domains shrink. We mention that this regularization behavior can also be found from the flow equation in the continuous setting. In spite of this, we derive the behavior based on eigen-analysis, since we are in discrete setting. Our interpretation also benefits from the result of the completeness of the eigenvectors.

We then come to two observations which are very useful in applications. The first one is the suggestion for designing initial flow functions in our first and last applications. Good initial flow functions should not be piecewise constant. The second observation is that the flow provides edge preservation when applied to image regularization. The inner parts of piecewise constant regions will be preserved, and gradually changing intensity in the transition domains (such as textures) will shrink fast; see the Lena example in our second application.

A direct corollary of Theorem 2 is the stability as follows. Corollary 1: For any initial value Φ_0 , and any time sequence $\{t^n, n = 0, 1, 2, \dots\}$ with $t^0 = 0$,

$$\|\Phi(t^{n+1})\|_2 \le \|\Phi(t^n)\|_2$$

D. Differences between dGCF (g = 1) and Laplacian smoothing

Here we emphasize that the discretized geodesic curvature flow is totally different from Laplacian smoothing [41], although we use a constant approximation to the gradient outside the divergence operator in the derivation.

The Laplacian smoothing [41] is, in our context, as follows (with a little different notation)

$$\begin{cases} (S + (t^{n+1} - t^n)W)\Phi(t^{n+1}) = S\Phi(t^n) \\ \Phi(t^0) = \Phi_0 \end{cases}, \quad (16)$$

where $W = (-w_{ij})$ is a positive semi-definite symmetric matrix with

$$w_{ij} = \begin{cases} \sum_{\substack{\tau, [v_i, v_j] \prec \tau \\ T \in D_1(i) \\ \tau \in D_1(i) \\ 0, \text{ others}}} c_{ij, \tau}, \quad j = i \\ 0, \text{ others} \end{cases}$$
(17)

There are three ways to see the differences between dGCF and Laplacian smoothing. Firstly we compare the schemes; see Eqn. (9) and (16). In general $G(\Phi(t^n))H(\Phi(t^n)) \neq W$. The only case for $G(\Phi(t^n))H(\Phi(t^n)) = W$ is that the absolute gradient of the discrete flow function $\Phi(t^n)$ reaches a same value in all triangles. Such kind of flow function can be determined by a nonlinear system of T - 1 equations with V unknowns, which has only finite number of solutions. However, the system is very hard to solve. The second way is to directly compare the solutions of Eqn. (9) and (16). Suppose that both the dGCF and Laplacian smoothing start from the initial flow function Φ_0 . We can find out the conditions for Φ_0 to give the same solution of dGCF and Laplacian smoothing under the same time sequence t^0, t^1, \cdots . From t^0 to t^1 , we get a nonlinear equation for Φ_0 ; From t^1 to t^2 , we get another nonlinear equation for Φ_0 ; and so on. Putting all these nonlinear equations (actually infinitely many) together gives the constraint for Φ_0 . Unfortunately, this system is also very hard to solve due to its nonlinearity. These view points at least show that dGCF and Laplacian smoothing does not have the same solution in general. The third way is numerical simulation and their limit behavior; see also the scale-space construction of images painted on manifolds in the application section. The dGCF and Laplacian smoothing behave very differently in topology simplification of images. In addition, they give different limit behaviors. We have proved in [42] that Laplacian smoothing gives constant limit for each initial flow function. However, the limit behavior of dGCF is much more complicated and depends on the geometry of the manifold and the initial flow function, as well as the scale parameter. Although we cannot prove it at this stage,

we conjecture that dGCF gives piecewise constant limit behavior. It seems impossible for dGCF to give a constant limit behavior in general, otherwise it cannot be applied to our curve evolution and edge detection applications.

V. APPLICATIONS

In this section we discuss some applications of dGCF.

A. Curve evolution under geodesic curvature flow

If we set the weight function g = 1, then we get curve evolution under geodesic curvature dependent velocity. As mentioned in the introduction, a closed initial curve on a manifold will evolve into one of two possible shapes under the geodesic curvature flow. It may disappear or become closed geodesic(s). Moreover, geodesics on manifolds can be stable or unstable [30].

We show some examples calculated with dGCF. In Fig. 3, 4 and 5 evolutions of the same initial curve on three mountain surfaces with different heights are illustrated, where we use the function

Mountain
$$(x, y) = \begin{cases} 2^{-1/(1-x^2-y^2)}, & x^2+y^2 < 1\\ 0, & x^2+y^2 \ge 1 \end{cases}$$

to build the surfaces. The initial curves are defined to be the same ellipse. In the next example we show the evolution of great circles on the unit sphere under the geodesic curvature flow; see Fig. 6. It's well known that there are infinitely many closed geodesics on the unit sphere and all of them are great circles and unstable. This example illustrates the instability of these closed geodesics. Fig. 7, 8 and 9 show several other examples about the evolution of curves under geodesic curvature flow on Dumbbell surface. Three different geodesic curvature flows with different initial curves are illustrated. In the former two examples in Fig. 7 and 8, the initial curves are chosen to be at two different sides of an unstable geodesic at one arm of the dumbbell. The geodesic curvature flow generates different results. Fig. 9 illustrates an example about the instability of geodesics on Dumbbell.



Fig. 3. Curve evolution on Mountain 1 surface defined as (x, y, z(x, y)) where z(x, y) = Mountain(x - 1, y) + Mountain(x + 1, y). The curve evolves from (a) via (b)(c)(d)(e) to (f), where it vanishes finally. This is reasonable, since the mountains of this surface are too low and hence near to the base plane so that one can expect the same result with the curve shortening flow on the 2D plane.

Let's investigate the next more interesting examples about curve evolution on the bunny surface; see Fig. 10 and 11. In these two examples, we start from the same initial curve and the same



Fig. 4. Curve evolution on Mountain 2 surface defined as (x, y, z(x, y)) where z(x, y) = 3Mountain(x - 1, y) + Mountain(x + 1, y). The curve evolves from (a) via (b)(c)(d)(e) to (f). Here the curve shrinks faster on the lower mountain and then fully moves to the higher one. Finally it disappears.



Fig. 5. Curve evolution on Mountain 3 surface defined as (x, y, z(x, y)) where z(x, y) = 3Mountain(x - 1, y) + 3Mountain(x + 1, y). The curve evolves slowly from (a) via (b)(c) to (d), a closed geodesic.

initial flow function with different time steps. In Fig. 10, the time step is small; in Fig. 11, the time step is large (10 times of the former). As one can see, two different results are generated. Both final curves are stable closed geodesics. The reason for this phenomenon is that the first geodesic around the neck of the bunny is not stable "enough" and consequently too much singlestep evolution (a large time step) may skips it. This is not safe for finding closed geodesics on manifolds, which is now still a challenging problem for general manifolds in pure mathematics and the numerical understanding to this problem is rather little. In our dGCF, the time step $\triangle t$ can be arbitrary according to Theorem 1. Flows with smaller time steps behave more safely, but the curve will evolve slowly. On the other hand, in this problem too large time steps result in skipping possible geodesics, since geodesics are locally shortest. Taking both efficiency and accuracy into account, we present a time step adaptivity algorithm; see Algorithm 1. The flexible time step setting strategy is ensured by Theorem 1 and Corollary 1 to avoid computational instability.

Different from the usually used strategy of fixing the time step, Algorithm 1 sets new time step dynamically according to the change rate of the curve lengths. The main part of this algorithm



Fig. 6. Curve evolution on the unit sphere: from (a) a great circle, via (b)(c)(d)(e)(f)(g)(h) to (i). The initial curve (a) is chosen to be the Equator, by setting the flow function to be $\phi(x, y, z) = z$. As one can see, the curve evolves very slowly at the beginning and oscillates around the geodesic. This is because the forces from the both sides are nearly balanced. Once this balance is broken, the curve evolves much faster and finally disappears.



Fig. 7. Curve evolution on Dumbbell: from (a) via (b)(c) to (d), where the curve finally vanishes.

is the loop in step 6, in which two cases are considered for the modification of the time step Δt . In the first case, the length L_k of the current curve C_k is larger than the previous one. This indicates that we have used a too big step from k-1 to k and the curve is now very near to a geodesic. Hence we set the flow function backward a step to $\Phi(t^{k-1})$ and use a smaller time step, to recompute the evolution from k-1 to k. The resetting of the time step is based on the reasonable assumption that at this moment the change rate of the curve length $\frac{dL}{dt}$ has the same absolute value with different signs at the two sides of the potential



Fig. 8. Curve evolution on Dumbbell: from (a) via (b)(c) to (d), a closed geodesic.



Fig. 9. Curve evolution on Dumbbell: from (a) an unstable geodesic, via (b)(c) to (d). Under the geodesic curvature flow, the curve shrinks and breaks into two curves. Both curves continue shrinking and finally disappear.

geodesic; see Fig. 12(a). We just find the intersection of the two lines at both sides of the potential geodesic. The t variable of this intersection is assumed to be the geodesic location and the new time step is just the difference between t and t^{k-1} . If this difference $\Delta t < 0$, we just set it to be $\frac{1}{2}(t^k - t^{k-1})$. In the second case as shown in Fig. 12(b)(c) with two subcases, the length of the curve keeps decreasing. We then use the decreasing rate of the curve length $\frac{dL}{dt}$ to determine a new time step. We know $\frac{dL}{dt}$ should be zero at geodesics. This gives a new time step setting in the first subcase shown in (b) where the decreasing rate from C_{k-1} to C_k is smaller than that from C_{k-2} to C_{k-1} . In this subcase we can hope that in the new iteration the curve will be near to a geodesic. Therefore we set a new time step according to the intersection of the t and the line passing through $\left(\frac{dL}{dt}\right)|_k$ and $\left(\frac{dL}{dt}\right)|_{k=1}$. However, we do not use the precise difference between the intersection and t^k as the new time step. We use its one half instead since we should avoid to march too much in a single step evolution. If the decreasing rate from C_{k-1} to C_k is larger than that from C_{k-2} to C_{k-1} , the geodesic is far from here. Hence we just set the new time step to be proportional to $\left(\frac{dL}{dt}\right)|_k/\left(\frac{dL}{dt}\right)|_{k-1}$ as stated in the algorithm.



Fig. 10. Curve evolution under geodesic curvature flow with time step 0.005: from (a) via (b)(c) to (d).



Fig. 11. Curve evolution under geodesic curvature flow with time step 0.05: from (a) via (b)(c)(d)(e) to (f). The initial curve and flow function are the same as Fig. 10.

Since the main loop in step 6 uses three-step information to determine the new time step, we need to compute one step evolution from the initial curve to start it up. We use another loop procedure in step 3. There are two possibilities to terminate this loop, depending on whether the initial curve C_0 is a geodesic. If C_0 is chosen to be a geodesic, then this loop will end with $|L^1 - L^0| < \epsilon$ and $L^1 > L^0$ and consequently step 4 will terminate the whole algorithm. If the algorithm starts with a non-geodesic C_0 , the loop in step 3 will in general end with $L^1 < L^0$ (in fact the loop will perform just once by setting a small initial time step). Then the loop in step 6 starts up. In this way, the algorithm generates a sequence of $\{C_k, k = 0, 1, 2, \dots\}$ and the Algorithm 1 Time step adaptive geodesic curvature flow of Φ_0 on M

1. Set an initial time step $\triangle t$; Set $t^0 = 0$;

2. Compute the length L^0 of the zero level set C_0 of $\Phi(t^0) = \Phi_0$; 3. Do

Set $\Delta t = \frac{1}{2}\Delta t$; Set $t^1 = \Delta t$; Evolve $\Phi(t^0)$ to $\Phi(t^1)$; Compute the length L^1 of the zero level set C_1 of $\Phi(t^1)$; While $(L^1 > L^0$ and $|L^1 - L^0| \ge \epsilon$) 4. If $(|L^1 - L^0| < \epsilon$ and $L^1 > L^0$) Exit; 5. Set k = 1; 6. Do

Set k = k + 1; Set $t^k = t^{k-1} + \Delta t$; Evolve $\Phi(t^{k-1})$ to $\Phi(t^k)$; Compute the length L^k of the zero level set C_k of $\Phi(t^k)$; If $(L^k > L^{k-1})$ Set $\Delta t = \frac{1}{2}((t^k - t^{k-1}) + (t^{k-1} - t^{k-2})(L^k - L^{k-1})/(L^{k-1} - L^{k-2}))$; If $(\Delta t < 0)$ Set $\Delta t = \frac{1}{2}(t^k - t^{k-1})$; Backward the flow function a step to $\Phi(t^{k-1})$; Set k = k - 1; Else

 $\begin{aligned} & \operatorname{CR}_{\text{pre1}} = (L^{k} - L^{k-1})/(t^{k} - t^{k-1}); \\ & \operatorname{CR}_{\text{pre2}} = (L^{k-1} - L^{k-2})/(t^{k-1} - t^{k-2}); \\ & \text{If } (\operatorname{CR}_{\text{pre1}} > \operatorname{CR}_{\text{pre2}}) \\ & \text{Set } \Delta t = \frac{1}{2}(t^{k} - t^{k-1})(0 - \operatorname{CR}_{\text{pre1}})/(\operatorname{CR}_{\text{pre1}} - \operatorname{CR}_{\text{pre2}}); \\ & \text{Else} \\ & \text{Set } \Delta t = (t^{k} - t^{k-1})\operatorname{CR}_{\text{pre1}}/\operatorname{CR}_{\text{pre2}}; \end{aligned}$

While(not $|L_{k-1} - L_k| < \epsilon$)



Fig. 12. adaptive time step setting.

curves become shorter and shorter. Hence the curve length series $\{L^k, k = 0, 1, 2, \cdots\}$ is monotonically decreasing with 0 as an obvious lower bound. This gives the existence of the limit of L^k which implies that the curve sequence converges to local geodesic(s) for $\lim_{k \to +\infty} L^k > 0$, or disappears for $\lim_{k \to +\infty} L^k = 0$. In the case that the limit is geodesic(s), they are obviously stable. We then come to the following

Proposition 1: For any given initial curve C_0 as the discrete zero level set of the flow function Φ_0 on a compact M, Algorithm 1 is convergent. Specially, the sequence of $\{C_k, k = 0, 1, 2, \dots\}$ will disappear finally or evolve into stable closed geodesic(s).

Here we use the change rate of curve lengths to design a convergent time step adaptivity algorithm. Theorem 1 ensures the feasibility and the sophisticated time step modification strategy gives the convergence of the algorithm. Some examples are provided in Fig. 13 and 14. We mention that our strategy is just one adaptive time step method. It's definitely true that one can have other approaches to modify the time step dynamically.



Fig. 13. Curve evolution under geodesic curvature flow with adaptive time steps for finding closed geodesics on the bunny surface: (a)(d) two initial curves; (b)(e) curves at middle time points; (c)(f) two final curves.



Fig. 14. Curve evolution under geodesic curvature flow with adaptive time steps for finding closed geodesics on the horse surface: (a)(c)(e) three initial curves; (b)(d)(f) final curves corresponding to (a)(c)(e) respectively.

B. Discrete intrinsic scale-space construction of images

Due to its good property of inclusion order preserving over other scale-spaces, the morphological scale-space concept of planar images [18] has been extended to images painted on parametric surfaces in [19], [33]. This generalization is valuable for multi-scale analysis of images on surfaces, and can be applied to post processing of texture mapping. Considering the wide use of triangle meshes in computer graphics, here we construct the corresponding concept for images on triangulated surfaces. We consider the discrete version since in practice scale-space evolutions are evaluated exclusively at a finite number of scales.

Definition 2: Let f be an initial image given on a triangulated surface M. The image sequence $\{T_n f = u^{(n)} = u(t^n) = u(n \triangle t), n = 0, 1, 2, \dots\}$ calculated from

$$\begin{cases} (S + \Delta t G(u^{(n)}) H(u^{(n)})) u^{(n+1)} = S u^{(n)} \\ u^{(0)} = f \end{cases}$$
(18)

is called dGCF scale-space of f and n is the discrete scale. Here $\triangle t$ is fixed and S, G, H are defined as in the last section with the weight function $g \equiv 1$.

The scale-space in the above definition is intrinsic to the surface since the calculation automatically includes the surface metric. Before showing examples, we briefly discuss some basic properties generally required by most scale-spaces. At first Theorem 1 immediately gives the existence and uniqueness. Also trivial verifications show the discrete semi-group property and the grey level shift invariance as follows.

Proposition 2: For any $n_1 \ge 0$ and $n_2 \ge 0$, $T_{n_1+n_2}f = T_{n_2}(T_{n_1}f) = T_{n_1}(T_{n_2}f)$.

Proposition 3: Let $C = (c, c, \dots, c)$ be a V-dimensional vector, then $T_n(f + C) = T_n(f) + C$.

In addition, Theorem 2 and the interpretation following it give the information reduction property (a regularization effect). The zero frequency component as a constant function keeps unchanged while high frequency components such as gradually changing intensity shrink fast in the scale-space evolution. Piecewise constant regions will shrink slowly since in a single scale evolution only transition domains between these regions change.

Some examples are provided in Fig. 15, 16, 17 and 18, with comparisons to Laplacian smoothing [41]. As one can see, the topologies of the images in the scale-spaces get simpler when scales get larger. This property is basic for scale-spaces of planar images, and is preserved for images over mesh surfaces. It also can be observed in the Lena example the edge preservation effect that in the scale-space evolution piecewise constant regions evolved slowly while small oscillations such as textures vanish very quickly. In the bunny example shown in Fig. 16, we compare dGCF scale-space and Laplacian smoothing. The dGCF scale-space exhibits inclusion order preserving clearly. Disconnected objects never interfere with each other. Also the red nose disappears first, and then the black eyes, and finally the feet. It's a gradual simplification procedure in which small objects with large geometric irregularities (such as small objects whose boundaries have large and tempestuously variational geodesic curvatures; e.g., the red nose of the bunny.) disappear first while large objects later as expected. Hence in dGCF scale-spaces, geometric information of objects in images play an important role. Differently from dGCF, Laplacian smoothing simplifies image structures via isotropically averaging the image intensity step by step and consequently, the objects get more and more blurred, as shown in Fig. 16. From the examples in Fig. 15 and 16, one may expect a constant limit behavior of dGCF scale-space. In fact this is not true in general. Remind the curve evolution

application in the previous subsection. According to Prop. 3, the scale-space evolution is transformed to be a curve evolution procedure if one subtracts a constant from the image intensity. The possible existence of closed geodesics tells us that on general manifolds the constant limit behavior may not be true. However, we guess that the limit image should be piecewise constant, and the intensity change can only happen at stable closed geodesics. Our first application on curve evolution also suggests that the piecewise constant limit image depends on the scale parameter $\triangle t$. We particularly verify this in the examples shown in Fig. 17 and 18. In Fig. 17, we compute dGCF scale-space evolution of a black-white image on the unit sphere. Here the initial image is specially chosen to be a half-black and half-white image where the intensity change happens at the equator (an unstable closed geodesic) of the sphere. As one can see, the intensities of both sides of the equator become closer and closer along the increasing scale and finally the same value, i.e., a constant limit. Actually on manifolds who have no stable closed geodesics, the constant limit behavior can always be expected. In the example of Fig. 18, the limit behavior of dGCF scale-spaces with different scale parameter $\triangle t$ of an image on the bunny surface have been studied, with comparisons to Laplacian smoothing. As one can see, the limit of geodesic curvature flow scale-space (piecewise constant with intensity changes at stable closed geodesics, but currently not proved) depends on the parameter for surfaces which have stable closed geodesics. In contrast, Laplacian smoothing exhibits much simpler limit behavior. It gives constant limit, which does not depend on the scale parameter and the geometry of the surface, as proved in [42].



Fig. 17. dGCF scale-space of a half-black image on the unit sphere: (a) the initial image; (b)(c)(d) images at increasing scales.

C. Edge detection

In this subsection we apply the geodesic curvature flow to detect edges of images on triangulated surfaces, where $g(\cdot)$ is no longer constant but a weight function depending on the intensity gradient of images on surfaces. This is a generalized work of active contour model in planar image processing [17], [28], [3], [13] and can be used to 3D painting and editing systems.

Assume that we have a gray image f on M. Since we want the contour to stop evolving at edges of the image, the weight function $g(\cdot)$ is usually chosen to be monotonically decreasing with respect to the absolute image intensity gradient $|\nabla f|$. Note that the piecewise constant $|\nabla f|$ gives piecewise constant weight over the triangulated surface M. In this work, we use

$$g(|\nabla f|) = \frac{1}{1 + k|\nabla f|^2}$$
(19)

where k is a constant. For color images, the function g can be similarly designed by combining gradients of all the channels.

Geodesic curvature flow using level set formulation can detect image edges with complex structures, such as multiple disjoint



Fig. 19. Edge detection using geodesic curvature flow on the horse surface. The curve evolves from (a) via (b)(c) to (d).



Fig. 20. Edge detection using geodesic curvature flow on the bunny surface. The curve evolves from (a) via (b)(c) to (d).

objects and blurred edges. We show some examples. In Fig. 19, the object to be detected is a leaf on the horse surface. The leaf has very sharp corners and slim shank. Our dGCF can easily capture the leaf without losing these micro structures. In the example of



Fig. 15. dGCF scale-space of Lena on an open surface. From left to right: increasing scales. (a) is the initial image.



Fig. 16. Comparison between dGCF scale-space (1st row) and Laplacian smoothing (2nd row) of a cartoon on the bunny. From left to right: increasing scales. (a)(f) are the initial images.



Fig. 18. Comparisons between limit behaviors of dGCF scale-spaces and Laplacian smoothing, as well as limit behaviors of dGCF scale-spaces with different scale parameters. (a) an image that gradually changes from black to white on the bunny surface. (b) the image of dGCF scale-space with scale parameter $\Delta t = 0.005$ at n = 10000. (c) the image of Laplacian smoothing with scale parameter $\Delta t = 0.005$ at n = 10000. (d) the image of dGCF scale-space with scale-space with scale parameter $\Delta t = 0.05$ at n = 10000. (e) the image of Laplacian smoothing with scale parameter $\Delta t = 0.05$ at n = 10000.



Fig. 21. Edge detection on a dumbbell surface. The curve evolves from (a) via (b)(c) to (d). (e)(f)(g)(h) are images at a different view point corresponding to (a)(b)(c)(d), respectively.

Fig. 20, the zero level curve of the flow function dynamically changes its topology and breaks into multiple branches when needed, and finally captures all the objects. In our last example, we test dGCF for detecting blurred objects that are very far from each other and located on different branches of the surface separated by many geodesics. In this case we use an initial flow function with two disjoint zero level sets. Each level curve locates on the corresponding branch as the object. Under the flow, the two curves evolve towards the expected objects and finally capture them, in spite of their shapes (square or circle). The final contour locates at the middle of the blurry band of the image edges. We provide two groups of pictures with different view points in Fig. 21. There's one thing that we should point out. Different from the previous two applications, here we use large time steps since we want to find the globally shortest (in weighted length) curve.

D. Remarks on the implementation and numerical experiments

The implementation consists of three parts. The first one is to build basic data structures for the calculation. From the given mesh data we extract the 1-Disks and 1-Neighbors for all vertices, which are stored in arrays indexed by vertices. We also compute the gradients of the piecewise linear basis functions and the coefficients $\{c_{ii,\tau}, c_{ij,\tau}, c_{ik,\tau}, c_{ji,\tau}, c_{jj,\tau}, c_{jk,\tau}, c_{ki,\tau}, c_{kj,\tau}, c_{kk,\tau}\}$. These data are arranged as arrays indexed by triangles. The second part is to solve the single step evolution, i.e., the linear system (9). We use PBCG method to solve (9), since it's a highly sparse linear system, say, Ax = b for simplification. We use an array to store the absolute gradient of the flow function indexed by triangles. It is updated dynamically. Using this array, together with the coefficients calculated above, we implement both Ax and A'x for PBCG method, due to the non-symmetry of the coefficient matrix in our problem. The third part is the implementation of Algorithm 1 and extraction of the zero level set of the flow function, which is quite straightforward. A simple marching-triangle-like approach is enough for the latter.

We also give some discussions on the initializations of flow functions in the curve evolution and edge detection applications, as well as the choice of the initial time step in Algorithm 1. There are two concerns for the initialization of flow function. At first, the discussion on the regularization behavior of dGCF in Section 4 suggests that the initial flow function should better be not piecewise constant but gradually changing. Secondly, the initialization should be as fast as possible (especially for large meshes). It's not necessary to be very precise. Therefore we use planes to initialize the flow function. In most of our examples we use only one plane since there's only one curve initially. The values of the initial flow function are set to be the signed Euclidean distance of the vertices to the plane. However, in our last example, we need two initial curves. Therefore we use two planes to generate the initial flow function. Concretely, we set

$$\begin{split} \Phi_0(v_i) &= \mathrm{SignMin}(P_1(v_i), P_2(v_i)) \\ &= \mathrm{Sign}(P_1(v_i)) \mathrm{Sign}(P_2(v_i)) \mathrm{Min}(|P_1(v_i)|, |P_2(v_i)|), \end{split}$$

where $\text{Sign}(\cdot)$ is the sign function and P_1, P_2 are algebraic representations of two planes. This simple trick can be extended to multi-planes. It transforms a partition with multi-planes to patches with different signs as expected. As for the initial time step in Algorithm 1, it depends on both the geometry of the initial curve and the initial flow function. To quantify this will be a long story, and in our experiments we simply choose a small initial step such as 0.005. One may consider the geodesic curvature of the initial curve and the gradient of the flow function to design an initial time step.

Some notes on numerical experiments are presented here. Table I gives the mesh information of models we used in this work. Therein SLTAR stands for the smallest largest triangle area ratio, and defined as

$$\text{SLTAR} = \frac{\min_{\tau} \text{ area of } \tau}{\max_{\tau} \text{ area of } \tau}.$$

Sslelr denotes the smallest one of ratios of shortest and longest edge lengths in triangles, which reads

$$\text{Sslelr} = \min_{\tau} \frac{\min_{\substack{e \prec \tau}} \text{ length of } e}{\max_{e \prec \tau} \text{ length of } e}$$

SLTAR describes globally the distribution of triangles, whereas Sslelr measures locally the quality of triangles. As shown in the table, some meshes are very irregular, such as the Dumbbell model. Our dGCF can handle all of them. The efficiency depends on the geometry of the surface, as well as the initializations of the curve and the flow function. If the curve is very near to a closed geodesic, then the flow converges very quickly; otherwise it will take some time. In our examples the CPU costs vary from several seconds to several minutes.

TABLE I Mesh information of models

| Model | # vertices | # triangles | SLTAR | Sslelr |
|------------|------------|-------------|--------------|-------------|
| Mountain 1 | 8649 | 16928 | 0.609129 | 0.501487 |
| Mountain 2 | 8649 | 16928 | 0.248045 | 0.240832 |
| Mountain 3 | 8649 | 16928 | 0.248045 | 0.240832 |
| Sphere | 62094 | 124184 | 0.000138661 | 0.0112244 |
| Dumbbell | 18878 | 37752 | 2.64159e-007 | 0.000507087 |
| Bunny | 34817 | 69630 | 0.00375069 | 0.0921986 |
| Horse | 48485 | 96966 | 0.000356748 | 0.0295239 |
| Lena | 65536 | 130050 | 0.303472 | 0.290264 |

VI. CONCLUSION AND FUTURE WORK

In this paper we discussed a level set formulation of geodesic curvature flow on triangulated surfaces. The discretized flow, dGCF, was rigorously analyzed and, applied to several problems about curve and image motion on surfaces. The method is totally different from diffusion equations [41] in mechanism, computation and applications. It offers a method to compute stable closed geodesics on triangulated manifolds, the problem which, to our knowledge, has not been studied so far. In the image motion application, geodesic curvature flow provides much better multi-scale representations than other approaches such as Laplacian smoothing. In addition, the level set formulation of the flow benefits from the topology adaption and the availability to blurred images, which is particularly useful in image edge detection. However, our method still has some disadvantages. In the case of finding the geodesic between two given points, the Lagrangian framework is much more suitable than Eulerian framework. Furthermore, examples showed that the limit behavior of image motion under geodesic curvature flow is much more complicated than Laplacian smoothing. With different scale parameters, different limit images may be generated. In contrast, Laplacian smoothing does not suffer from this uncertainty.

Several problems are left open. At first, more sophisticated adaptive time step control strategies are needed in the application of finding closed geodesics. Also better flows can be introduced to find all the possible closed geodesics including unstable ones. Secondly, in the scale-space construction application, the numerical examples showed the piecewise constant limit behavior of the flow. To rigorously prove this result will be very valuable for many applications, not only scale-space analysis. Besides, to precisely describe the dependency of the limit behavior on the scale parameter for images on a given surface is very interesting. On the third, error estimate of our dGCF to the original geodesic curvature flow on smooth manifold should be investigated, which will give how the dGCF simulates the original flow. This is not easy due to two reasons, the complexity of the flow in continuous setting and the unspecific underlying manifolds before triangulation. Finally, our method can be applied to many other problems, such as mesh segmentation.

ACKNOWLEDGEMENT

The research has been supported by MOE (Ministry of Education) Tier II project T207N2202 and IDM project NRF2007IDM-IDM002-010. In addition, support from SUG 20/07 is also gratefully acknowledged.

REFERENCES

- J. Babaud, A. Witkin, M. Baudin, and R. Duda. Uniqueness of the gaussian kernel for scale-space filtering. *IEEE Trans. Pattern Anal. Machine Intell.*, PAMI-8, 1986.
- [2] T. Barth and J. Sethian. Numerical schemes for the hamilton-jacobi and level set equations on triangulated domains. J. Comput. Phys., 145:1–40, 1998.
- [3] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. Int'l J. Comput. Vision, 22:61–79, 1997.
- [4] L.T. Cheng, P. Burchard, B. Merriman, and S. Osher. Motion of curves constrained on surfaces using a level-set approach. J. Comput. Phys., 175:604–644, 2002.
- [5] D.L. Chopp and J.A. Sethian. Flow under curvature: Singularity formations, minimal surfaces, and geodesics. *Experimental Mathematics*, 2(4):235–255, 1993.
- [6] A. Cunha, R. Teixeira, and L. Velho. Discrete scale spaces via heat equation. In Proceedings of sibgrapi 2001-xiv brazilian symposium on computer graphics and image processing. IEEE Press, 2001.
- [7] H.T. Davis and K.T. Thomson. Linear Algebra and Linear Operators in Engineering: With Applications in Mathematica. Academic Press, 2000.
- [8] L.C. Evans and J. Spruck. Motion of level sets by mean curvature i. J. Diff. Geom., 33:635–681, 1991.
- [9] L. Florack and A. Kuijper. The topological structure of scale-space images. *Journal of Mathematical Imaging and Vision*, 12(1):65–80, 2000.
- [10] M. Gage. Curve shortening makes convex curves circular. Invent. Math., 76:357–364, 1984.
- [11] M. Gage. Curve shortening on surfaces. Ann. Sci. Ecole Norm. Sup., 23(2):229–256, 1990.
- [12] M. Gage and R.S. Hamilton. The heat equation shrinking convex plane curves. J. Diff. Geom., 23:69–96, 1986.
- [13] R. Goldenberg, R. Kimmel, E. Rivlin, and M. Rudzsky. Fast geodesic active contours. *IEEE Trans. Image Process.*, 10(10):1467–1475, 2001.
- [14] M.A. Grayson. The heat equation shrinks embedded plane curves to round points. J. Diff. Geom., 26:285–314, 1987.
- [15] M.A. Grayson. Shortening embedded curves. Ann. of Math., 129:71– 111, 1989.
- [16] A. Hummel. Representations based on zero-crossings in scale-space. In Proc. IEEE Computer Vision and Pattern Recognition Conf., pages 204–209, 1986.
- [17] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: active contour models. *Int'l. J. Comput. Vis.*, 1:321–331, 1988.
- [18] B.B. Kimia and K. Siddiqi. Geometric heat equation and nonlinear diffusion of shapes and images. *Computer Vision and Image Understanding*, 64(3):305–322, 1996.
- [19] R. Kimmel. Intrinsic scale space for images on surfaces: The geodesic curvature flow. *Graphical Models and Image Processing*, 59(5):365– 372, 1997.
- [20] J. Koenderink. The structure of images. *Biol. Cybern.*, 50:363–370, 1984.
- [21] A. Kuijper and L.M.J. Florack. Understanding and modeling the evolution of critical points under gaussian blurring. In *Proceedings of the 7th European Conference on Computer Vision 2002, Part I*, pages 143–157, 2002.
- [22] A. Kuijper, L.M.J. Florack, and M.A. Viergever. Scale space hierarchy. Journal of Mathematical Imaging and Vision, 18(2):169–189, 2003.
- [23] T. Lindeberg. Scale-space behaviour of local extrema and blobs. *Journal of Mathematical Imaging and Vision*, 1(1):65–99, 1992.
- [24] T. Lindeberg. Scale-Space Theory in Computer Vision. Kluwer, Netherlands, 1994.
- [25] T. Lindeberg. On the axiomatic foundations of linear scale-space. In J. Sporring et al., editor, *Gaussian Scale Space Theory*. Kluwer, 1996.
- [26] T. Lindeberg. Scale-space: A framework for handling image structures at multiple scales. In Proc. CERN School of Computing, Egmond aan Zee, The Netherlands, 1996.
- [27] L. Ma and D.Z. Chen. Curve shortening flow in a riemannian manifold. arXiv:math.DG/0312463 v1, 2003.

- [28] R. Malladi, J. Sethian, and B. Vemuri. Shape modeling with front propagation: a level set approach. *IEEE Trans. Pattern Anal. Mach. Intell.*, 17(2):158–175, 1995.
- [29] M. Meyer, M. Desbrun, P. Schröder, and A. Barr. Discrete differentialgeometry operator for triangulated 2-manifolds. In H.-C. Hege and K. Polthier, editors, *Vis. and Math. III.* Springer Verlag, 2002.
- [30] K. Mikula and D. Ševčovič. Evolution of curves on a surface driven by the geodesic curvature and external force. *Applicable Analysis*, 85(4):345–362, 2006.
- [31] S. Osher and J.A. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulations. J. Comput. Phys., 72:12–49, 1988.
- [32] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Trans. Pattern Anal. Machine Intell.*, 12(7):629–639, 1990.
- [33] A. Spira and R. Kimmel. Geometric curve flows on parametric manifolds. J. Comput. Phys., 223:235–249, 2007.
- [34] J. Sporring. The entropy of scale-space. In ICPR, volume A, pages 900–904, 1996.
- [35] V. Surazhsky, T. Surazhsky, D. Kirsanov, S.J. Gortler, and H. Hoppe. Fast exact and approximate geodesics on meshes. ACM Trans. Graph., 24(3):553–560, 2005.
- [36] J. Weickert. Scale-space properties of nonlinear diffusion filtering with a diffusion tensor. Technical Report Report No. 110, Laboratory of Technomathematics, University of Kaiserslautern, 1994.
- [37] J. Weickert. Nonlinear diffusion scale-spaces: From the continuous to the discrete setting. In *ICAOS: Images, Wavelets, and PDEs*, pages 111–118, Paris, 1996.
- [38] J. Weickert. Anisotropic diffusion in image processing. PhD thesis, Teubner, Stuttgart, 1998.
- [39] J. Weickert and B. Benhamouda. A semidiscrete nonlinear scale-space theory and its relation to the perona-malik paradox. In Advances in Computer Vision, pages 1–10. Springer, 1997.
- [40] A. Witkin. Scale-space filtering. In Int. Joint. Conf. Artificial Intelligence, pages 1019–1021, Karlsruhe, West Germany, 1983.
- [41] C.L. Wu, J.S. Deng, and F.L. Chen. Diffusion equations over arbitrary triangulated surfaces for filtering and texture applications. *IEEE Trans. Visual. Comput. Graph.*, 14(3):666–679, 2008.
- [42] C.L. Wu, J.S. Deng, F.L. Chen, and X.C. Tai. Scale-space analysis of discrete filtering over arbitrary triangulated surfaces. Accepted by SIAM J. Imaging Science, 2009.
- [43] A. Yuille and T. Poggio. Scaling theorems for zero crossings. IEEE Trans. Pattern Anal. Machine Intell., PAMI-8, 1986.