Geometric Flow for Quality Surface/Volumetric Modeling (Extended Abstract)

Chandrajit L. Bajaj *
Department of Computer Science,
Center for Computational Visualization,
Institute for Computational Engineering and Sciences
University of Texas, Austin, TX 78712
Email: bajaj@cs.utexas.edu

Guoliang Xu†, Qin Zhang
LSEC, Institute of Computational Mathematics,
Academy of Mathematics and System Sciences,
Chinese Academy of Sciences, Beijing, 100080
Email: {xuguo, zqyork}@lsec.cc.ac.cn

Abstract

We present a general variational framework for a higher-order spline level-set (HLS) method and apply this to smooth surface constructions. Starting from a first order energy functional, we derive the general level set formulation, and provide an efficient solution of a second order , time-dependent, geometric partial differential equation (termed geometric flow), using a C^2 B-spline basis. We also present a fast cubic C^2 B-spline interpolation algorithm based on convolution and the Z-transform, which exploits the local relationship of interpolatory cubic spline coefficients with respect to given function data values. We provide two demonstrative smooth surface construction examples of our HLS method. The first is the construction of a smooth surface model (an implicit solvation interface) of bio-molecules in solvent, given their individual atomic coordinates and solvated radii. The second is the smooth surface reconstruction from a cloud of points generated from a 3D surface scanner.

1 Introduction

Level set methods are increasingly being used in the solution of time-dependent (evolutionary) partial differential equations [6]. Here we consider its application to smooth surface construction

*Supported in part by NSF grant CNS-0540033 and NIH contracts P20-RR020647, R01-GM074258, R01-GM07308,R01-EB004873.

†Supported in part by NSFC grant 10371130 and National Key Basic Research Project of China (2004CB318000).

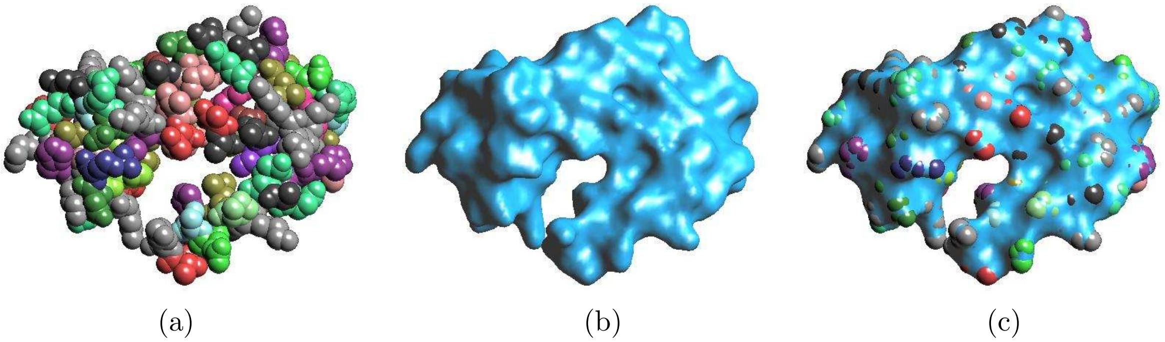

Fig 1.1: Variational C^2 cubic B-spline Molecular Surface Reconstruction[2, 10, 4]: (a) shows the van der Waals surface of a molecule. (b) shows the corresponding solvent excluded molecular surface constructed using our C^2 tri-cubic spline level-set method. (c) illustrates that the smooth solvent excluded surface constructed tightly encloses the van der Waals surface (a).

by presenting a higher order spline level set generalization and solution of an appropriate geometric evolutionary partial differential equation. The level set geometric formulation is derived by minimizing an energy functional defined with respect to the surface and its first derivatives.

Given a non-negative function $g(\mathbf{x})$ over a domain $\Omega \in \mathbb{R}^3$. Find a surface Γ in Ω, such that the energy functional

$$E(\Gamma) = \int_{\Gamma} g(\mathbf{x}) d\mathbf{x} + \epsilon \int_{\Gamma} h(\mathbf{x}, \mathbf{n}) d\mathbf{x} \tag{1.1}$$

is minimal, where $\mathbf{x}$ and $\mathbf{n}$ is a surface point and its surface normal, respectively. Further, $h(\mathbf{x}, \mathbf{n})$ is another given non-negative function defined over $\mathbb{R}^3 \times \mathbb{R}^3$ which is used for regularizing the constructed smooth surface. Finally, $\epsilon \geq 0$ is a constant. Many solid and physical modeling problems, such as surface (solid boundary) reconstruction, and physically based simulation of deformable interfaces could be formulated as minimizing an energy in the form of (1.1). By minimizing the energy functional (1.1), a partial differential equation (PDE) in the level-set formulation can be generated. The PDE is solved using the higher-order spline level-set (HLS) method that we present here. Fig. 1.1 and 1.2 show two examples of smooth surface construction, an interface (surface) that separates a molecule's atoms from the solvent atoms (typically water), and a smooth surface reconstruction fit to a point set generated by a 3D surface scanner.

Why Use a Level-set Method? In shape deformation simulations, topology changes may occur. This topology change makes parametric form surface tracking difficult. However, implicit form surface deformation could overcome this difficulty. Implicit surface splines, such as tetrahedral A-patches and prism A-patches, have been successfully used in computer graphics and surface modeling in the past decades (see [1, 5, 12] for references), however mostly used in static surface modeling. The level-set method described here allows one to dynamically deform and track an implicit surface spline using a governing PDE, which describes various laws of motion depending on geometry, external forces, or a desired energy minimization (see [6, 7, 9, 11]). Fur-

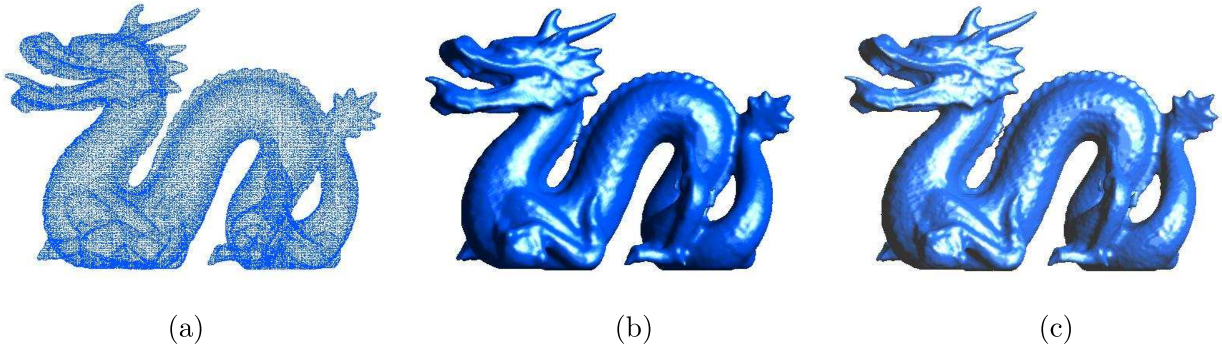

Fig 1.2: Variational C^2 cubic B-spline Point - Sampled Surface Reconstruction[]: (a) shows the point set data produced by a 3D surface scanner. (b) shows the the reconstruction result using our C^2 tri-cubic spline level-set method. (c) shows the reconstruction result for a C^0 tri-linear level-set method (also generated from our implementation).

thermore, the underlining data structure is simple and topological changes are handled easily, with the computation being restricted to a thin shell (traditionally called a narrow band for evolution) surrounding the level-set.

Why Use a Higher-order Spline Method? The level-set surfaces obtained from classical level-set methods are generally bumpy due to the use of piecewise tri-linear interpolation from the discrete function data computed on a rectilinear grid. To produce a better quality surface, a denser grid needs to be used. However, the increased grid resolution substantially increases the computation costs. Another drawback of using discrete data over grids is the non-trivial requirement of estimating derivatives for smooth interpolation. In many surface construction problems, such as the construction of molecular surface, the underlining surface is at least C^1 smooth. Therefore, a smooth level-set function is highly desirable. In this paper, we are solving second-order geometric partial differential equations. In the solutions of these PDE's we utilize accurate estimates of mean surface curvature Therefore, we use C^2 tri-cubic spline as the level-set function basis. Note that tri-cubic is the lowest order B-spline that could achieve C^2 continuity in 3D. The advantages using C^2 spline function bases include:

1. Since the level-set function is C^2 smooth, the level-set surface is G^2 smooth. There do exist a finite number of critical level-set values where the level-set may have a finite number of isolated singular points (i.e the gradient of the level-set function vanishes). However working in a finite precision numerical domain one automatically avoids these finite set of critical level-set values.

2. Derivatives up to the second order and curvatures, which appear in the governing geometric partial differential equations, are easily and exactly computed from the C^2 level-set function.

3. Using smooth level-set functions implies that larger and higher-order spline iso-surface

patches could be directly generated.

2 Mathematical Notation and Definitions

We first introduce some useful notation and definitions for geometric quantities on level-sets $\mathcal{M}$ in terms of the corresponding level-set function ϕ. Let $\phi : \Omega \to \mathbb{R}$ be some smooth function on a domain $\Omega \subset \mathbb{R}^3$. Suppose $\mathcal{M}_c := \{x \in \Omega : \ \phi(\mathbf{x}) = c\}$ is a level-set of ϕ for the level value c. For the sake of simplicity,we simply write $\mathcal{M} = \mathcal{M}_c$ and assume that $\|\nabla\phi\| \neq 0$ on $\mathcal{M}$. Hence by the implicit function theorem, $\mathcal{M}_c$ is a smooth surface and the normal

$$\mathbf{n} = \frac{\nabla\phi}{\|\nabla\phi\|}$$

on the tangent space $\mathcal{T}_x\mathcal{M}$ is defined for every x on $\mathcal{M}$. Using a co-area formula, the energy functional can be defined as

$$\mathrm{E}[\phi] := \int\limits_{\mathbb{R}} \mathrm{e}[\mathcal{M}_c]dc = \int\limits_{\Omega} \|\nabla\phi\| h(\mathrm{x}, \mathrm{n})\mathrm{d}x.$$

Next we compute

$$\langle \mathrm{E}'[\phi], \vartheta\rangle = \frac{\mathrm{d}}{\mathrm{d}\epsilon}\mathrm{E}[\phi + \epsilon\vartheta]\big|_{\epsilon=0} \tag{2.1}$$

$$\begin{aligned}
&= \frac{\mathrm{d}}{\mathrm{d}\epsilon}\int\limits_{\Omega} \|\nabla(\phi+\epsilon\vartheta)\| h\left(x, \frac{\nabla(\phi+\epsilon\vartheta)}{\|\nabla(\phi+\epsilon\vartheta)\|}\right)\mathrm{d}x\Bigg|_{\epsilon=0} \\
&= \int\limits_{\Omega} \frac{\mathrm{d}}{\mathrm{d}\epsilon}(\|\nabla(\phi+\epsilon\vartheta)\|)\big|_{\epsilon=0} h(\mathrm{x}, \mathrm{r})\mathrm{d}x \\
&+ \int\limits_{\Omega} \|\nabla\phi\| \left(h_{\mathrm{x}}\frac{d\mathrm{x}}{\mathrm{d}\epsilon} + h_{\mathrm{n}}\frac{\mathrm{d}\left[\frac{\nabla(\phi+\epsilon\vartheta)}{\|\nabla(\phi+\epsilon\vartheta)\|}\right]}{\mathrm{d}\epsilon}\right)\Bigg|_{\epsilon=0}\mathrm{d}x
\end{aligned} \tag{2.2}$$

The remaining task is to calculate

$$\frac{\mathrm{d}}{\mathrm{d}\epsilon}(\|\nabla(\phi+\epsilon\vartheta)\|)\big|_{\epsilon=0} = \frac{\nabla\phi\cdot\nabla\vartheta}{\|\nabla\phi\|}, \tag{2.3}$$

$$\begin{aligned}
\frac{\mathrm{d}\left(\left[\frac{\nabla(\phi+\epsilon\vartheta)}{\|\nabla(\phi+\epsilon\vartheta)\|}\right]\right)}{\mathrm{d}\epsilon}\Bigg|_{\epsilon=0} &= \frac{\nabla\vartheta\|\nabla\phi\| - (\nabla\phi\otimes\nabla\phi)\nabla\vartheta/\|\nabla\phi\|}{\|\nabla\phi\|^2} \\
&= \|\nabla\phi\|^{-1}\mathrm{P}\nabla\vartheta,
\end{aligned} \tag{2.4}$$

where operator $\mathrm{P} = \mathrm{I} - \frac{\nabla\phi}{\|\nabla\phi\|}\otimes\frac{\nabla\phi}{\|\nabla\phi\|}$ is the projection onto the tangent space and I indicates the identity mapping.

Next we compute $\mathrm{x}'(0)=\frac{d\mathrm{x}(\epsilon)}{\mathrm{d}\epsilon}|_{\epsilon=0}$. Since

$$\phi(x(\epsilon))+\epsilon\vartheta(x(\epsilon))=c,$$

Taking differential with respect to ϵ, and then taking ϵ to be zero, we have

$$(\nabla\phi)^T\mathrm{x}'(0)+\vartheta(\mathbf{x})=0. \tag{2.5}$$

Let

$$\mathrm{x}'(0)=\alpha(\mathbf{x})\frac{\nabla\phi}{\|\nabla\phi\|}+\beta(\mathbf{x})T(\mathbf{x}), \tag{2.6}$$

where$T(\mathbf{x})$ is a tangent vector at x. substituting (2.6) into (2.5), we obtain

$$\alpha(\mathbf{x})=-\frac{\vartheta(\mathbf{x})}{\|\nabla\phi\|}.$$

Since the motion in the tangential direction does not alter the shape of the surface, we ignore the tangential movement and take

$$\mathrm{x}'(0)=-\vartheta(\mathbf{x})\frac{\nabla\phi}{\|\nabla\phi\|^2} \tag{2.7}$$

Substituting (2.3), (2.4) and (2.7) into (2.2), we obtain

$$\begin{aligned}(2.2)\ =\ &\int_\Omega \frac{\nabla\phi\cdot\nabla\vartheta}{\|\nabla\phi\|}h(\mathrm{x},\mathrm{n})\\ &+\ \|\nabla\phi\|\left(-\frac{(\nabla h)^T\nabla\phi\ \vartheta}{\|\nabla\phi\|^2}+\frac{(\nabla_{\mathrm{n}}h)^T\mathrm{P}\nabla\vartheta)}{\|\nabla\phi\|}\right)\mathrm{d}x\end{aligned} \tag{2.8}$$

Equation (2.8) is the weak formulation of Euler-Lagrange equation. If we utilize the formula of integrate by parts taking into account $\vartheta\in\mathrm{C}_0^\infty(\Omega)$, Euler-Lagrange equation

$$-\mathrm{d}iv\left[h(\mathrm{x},n)\frac{\nabla\phi}{\|\nabla\phi\|}+\mathrm{P}\nabla_n h\right]-\frac{(\nabla h)^T\nabla\phi}{\|\nabla\phi\|}=0 \tag{2.9}$$

is deduced. Obviously, this equation is a second-order nonlinear partial differential equation. If we write the left hand side of (2.9) as an operator acted on function ϕ, we can construct geometric (gradient) flow as follows:

$$\partial_t\phi=-\|\nabla\phi\|L(\phi).$$

For further details, please see [3].

3 Algorithm Outline

From minimizing the energy (1.1), we obtain the following evolution equation (see [3] for details).

$$\begin{aligned}\frac{\partial \phi}{\partial t} &= (g+\epsilon h)\mathrm{d}iv\left(\frac{\nabla \phi}{\|\nabla \phi\|}\right)\|\nabla \phi\| + \epsilon\ \mathrm{d}iv(\mathrm{P}\nabla_n h)\|\nabla \phi\| \\ &+ 2[\nabla(g+\epsilon h)]^T\nabla\phi = L(\phi) + H(\nabla\phi), \end{aligned} \tag{3.1}$$

where

$$\begin{aligned} L(\phi) &= (g+\epsilon h)\mathrm{d}iv\left(\frac{\nabla \phi}{\|\nabla \phi\|}\right)\|\nabla \phi\|, \\ H(\nabla\phi) &= \epsilon\ \mathrm{d}iv(\mathrm{P}\nabla_n h)\|\nabla \phi\| + 2[\nabla(g+\epsilon h)]^T\nabla\phi, \end{aligned}$$

$\mathrm{P} = \mathrm{I} - \frac{\nabla\phi}{\|\nabla\phi\|} \otimes \frac{\nabla\phi}{\|\nabla\phi\|}$ is a projection operator onto the tangent space and I indicates the identity mapping. ∇ and ∇_n denote the usual gradient operator with respect to $\mathbf{x}$ and $\mathbf{n}$. Note that $L(\phi)$ is a parabolic term and $H(\nabla\phi)$ is a hyperbolic term. Hence, in solving equation (3.1) in the following, the first order term $H(\nabla\phi)$ is computed using an upwind scheme (see [8] for the reason of using an upwind scheme) over a finer grid, the higher order term $L(\phi)$ is computed using a spline presentation defined on a coarser grid.

Consider the solution of equation (3.1) over the domain $\Omega = [a,b] \times [c,d] \times [e,f] \in \mathbb{R}^3$. For simplicity, we assume $b-a = d-c = f-e > 0$. We suppose that the domain Ω is uniformly partitioned with vertices $G_0 = \{\mathbf{x}_{ijk}\}_{ijk=0}^n := \{x_i\}_{i=0}^n \times \{y_j\}_{j=0}^n \times \{z_k\}_{k=0}^n$, where

$$x_i = a + i\Delta x,\ \ y_j = c + j\Delta y,\ \ z_k = e + k\Delta z,$$

and $\Delta x = \Delta y = \Delta z = (b-a)/n$. Let G_l be the set of vertices of the grid which is generated by binary subdividing G_0 uniformly l times. Let ϕ be a piecewise tri-linear level-set function over the grid G_l, Φ be a tri-cubic spline approximation of ϕ over the grid G_0. In general, l is chosen as 0 or 1 or 2. In our implementation, we take $l = 1$. If $l = 0$, Φ and ϕ are defined on the same grid G_0. This is the simplest case. The aim of the following algorithm is to compute the spline level set function Φ.

Algorithm Steps.

1. **Initialization**. Given a initial Γ, construct a piecewise tri-linear level-set function ϕ over the grid G_l. If necessary, apply a re-initialization step to set ϕ to be a signed distance function to Γ (see [3] for details). Convert ϕ to Φ (see [3]).

2. **Evolution**. Resampling Φ to obtain a new ϕ over the grid G_l. Compute $L(\Phi)$ and $H(\nabla\phi)$ in the thin shell $N = \{(x_i, y_j, z_k) \in G_l : |\phi(x_i, y_j, z_k)| < \mathcal{H}\}$. Update ϕ in N for one time step to get $\tilde{\phi}$ by an ODE time stepping method (see [3] for details).

3. **Re-initialization**. Applying re-initialization step to $\tilde{\phi}$ in the shell

   $$N = \{(x_i, y_j, z_k) \in G_l : \min_{-1 \le \mu,\nu,\lambda \le 1} |\phi_{i+\mu,j+\nu,k+\lambda}| \le \mathcal{H}\}.$$

   to get a new ϕ (see [3]). Convert ϕ to Φ (see [3] for details). Go back to step 2 if the termination condition is not satisfied.

4. **Iso-contouring**. Extract 3-sided or 4-sided iso-surface patches (vertices with normals) of $\Phi = c$, where c is a given iso-value.

Remark 2.1. For the problem of molecular surface construction, the grid size G_0 should be less than the radii of atoms so that atoms are distinguishable from the level set surface. In our implementation, the grid size is chosen to be one-half of the minimal value of the atom radii.

Remark 2.2. The aim of using $l > 0$ is to make ϕ a more accurate approximation of the signed distance function. The larger the value of l we use, the better approximation of the signed distance function we have. Since the scanned data to be approximated in general suffers from noise, we use the approximation Φ over a coarse grid G_0 for denoising. Furthermore, for generating larger level-set surface patches, again a coarse grid needs to be used.

Acknowledgment. A substantial part of this work in this paper was done when Guoliang Xu was visiting Chandrajit Bajaj at UT-CVC. His visit was also supported in part by the J. T. Oden ICES visitor fellowship.

References

[1] C. Bajaj, J. Chen, and G. Xu. Modeling with cubic A-patches. *ACM Transactions on Graphics*, 14(2):103–133, 1995.

[2] C. Bajaj, V. Pascucci, A. Shamir, R. Holt, and A. Netravali. Dynamic maintenance and visualization of molecular surfaces. *Discrete Applied Mathematics*, 127:23–51, 2003.

[3] C. Bajaj, G. Xu, and Q. Zhang. A Higher Order Level Set Method with Applications to Smooth Surface Constructions. ICES Report 06-18, Institute for Computational Engineering and Sciences, The University of Texas at Austin, 2006.

[4] C. Bajaj, G. Xu, and Q. Zhang. A Fast Variational Method for Construction of Smooth Molecular Surfaces. ICES Report 08-19, Institute for Computational Engineering and Sciences, The University of Texas at Austin, 2008.

[5] C. Bajaj, G. L. Xu, S. Evans R. J. Holt, and A. N. Netravali. NURBS Approximation of A-Splines and A-Patches. *Intl. J. on Computational Geometry and Applications*, 2001.

[6] S. Osher and R. Fedkiw. *Level Set Method and Dynamic Implicit Surfaces.* Springer, New York, 2003.

[7] S. Osher and J. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 79:12–49, 1988.

[8] S. Osher and C.W. Shu. High-order essentially nonoscillatory schemes for hamilton-jacobi equations. *SIAM Journal of Numerical Analysis*, 28(4):907–922, 1991.

[9] D. Peng, B. Merriman, S. Osher, H. Zhao, and M. Kang. A PDE-based fast local level set method. *Journal of Computational Physics*, 155:410–438, 1999.

[10] Y. Zhang, G. Xu, and C. Bajaj. Quality meshing of implicit solvation models of biomolecular structures. *Computer Aided Geometric Design*, 23:510–530, 2006.

[11] H. Zhao, S. Osher, B. Merriman, and M. Kang. Implicit nonparametric shape reconstruction from unorganized points using a variational level set method. *Computer Vision and Image Understanding*, 80(3):295–319, 2000.

[12] W. Zhao, G. Xu, and C. Bajaj. An Algebraic Spline Model for Molecular Surfaces. In *Proceedings of the 2007 ACM symposium on Solid and physical modeling*, pages 297–302, Beijing, China, 2007. IEEE pub.