Mar. 2006, Vol.21, No.2, pp.249–254 *J. Comput. Sci. & Technol.*

Specification of Initial Shapes for Dynamic Implicit Curve/Surface Reconstruction*

Zhou-Wang Yang (杨周旺), Chun-Lin Wu (吴春林), Jian-Song Deng (邓建松), and Fa-Lai Chen (陈发来)

Department of Mathematics, University of Science and Technology of China, Hefei 230026, P.R. China

E-mail: {yangzw, dengjs, chenfl}@ustc.edu.cn

Revised January 18, 2006.

Abstract The dynamic implicit curve/surface reconstruction demands no special requirement on the initial shapes in general. In order to speed up the iteration in the reconstruction, we discuss how to specify the initial shapes so as to reflect the geometric information and the topology structure of the given data. The basic idea is based on the combination of the distance function and the generalized eigenvector fitting model.

Keywords Sampson distance, generalized eigenvector fitting, dynamic implicit surface reconstruction

1 Introduction

Surface reconstruction from an unorganized data point set is a challenging problem in Computer Aided Design and Computer Graphics. Generally, since the connectivity of the data sets and the topology of the real surfaces could be rather complicated, it is difficult to find a robust solution to the problem. A reconstruction procedure is expected to be able to deal with complicated topology and geometry as well as noise and non-uniformity of the data. The reconstructed surface should be a good approximation of the data set and with certain smoothness.

Traditionally, the parametric forms[1] are used to represent the reconstructed surfaces. In the process, the parametrization of points in the data set is a hard problem. Recently, implicit surfaces have attracted significant attentions[2,3] and are used to represent the reconstructed surfaces as well, where no parametrization is needed.

The authors have proposed a dynamic implicit curve and surface reconstruction scheme based on the Sampson distance[4] and a trust region method in optimization theory. The process gets started from a properly specified initial shape and converges toward the target data set through iterative minimization. The convergence ratio usually depends heavily on the initial shapes.

In order to speed up the convergence in the dynamic implicit curve/surface reconstruction, we wish that the initial shapes should reflect the geometry information and the topology structure of the given data as near as possible. In this paper, we discuss the specification of initial shapes for the dynamic implicit surface reconstruction. The idea is based on distance function and generalized eigenvector fitting model. The distance function describes the geometric information of the given data perfectly, and the generalized eigenvector fitting model can obtain a good approximation to the topology structure of the data set. We will combine them together to specify the initial shapes for the dynamic implicit curve/surface reconstruction.

The rest of the paper is organized as follows. In Section 2, we reviewed the dynamic implicit curve/surface reconstruction algorithm proposed in [4]. In order to specify a "good" initial shape for the algorithm, in Section 3 we simplify the objective function in the reconstruction algorithm and solve it with the generalized eigenvector method. The result shape reflects the topological structure of the given data, but with large geometric error. To improve the initial shape, we make use of the distance function, which is reviewed in Section 4. The computational method is presented as well in this section. Then in Section 5, by assembling the distance function with the sign of implicit function obtained from Section 3, we give the specification of initial shapes for dynamic implicit curve/surface reconstruction. Some curve and surface examples are also implemented to illustrate the efficiency of the scheme. Finally, we come to summary our proposed approach in Section 6.

2 Dynamic Implicit Curve/Surface Reconstruction

In this section, we review the dynamic implicit curve/surface reconstruction method proposed in [4].

2.1 Algebraic Tensor-Product B-Spline Surfaces

A tri-variate tensor-product B-spline function of degree (d, d, d) is defined as

$$f(x, y, z) = \sum_{r,s,t=1}^{m,n,o} c_{rst} M_r(x) N_s(y) O_t(z), \qquad (1)$$

*A preliminary version of this paper appeared in Proc. the 1st Korea-China Joint Conference on Geometric and Visual Computing.

This work is supported by the Outstanding Youth Grant of the National Natural Science Foundation of China (Grant No. 60225002), the National Natural Science Foundation of China (Grant Nos. 60533060 and 60473132), the National Basic Research 973 Program of China (Grant No. 2004CB318000), the TRAPOYT in Higher Education Institute of MOE of China, and SRF for ROCS, SEM.

where $\{c_{rst}\}_{m\times n\times o}$ are the real coefficients (also called *control coefficients*), $\{M_r(x)\}_{r=1}^{m}$, $\{N_s(y)\}_{s=1}^{n}$ and $\{O_t(z)\}_{t=1}^{o}$ denote the B-spline basis functions of degree d with respect to some user specified knot vectors $\boldsymbol{\xi}=\{\xi_r\}_{r=1}^{m+d+1}$, $\boldsymbol{\eta}=\{\eta_s\}_{s=1}^{n+d+1}$ and $\boldsymbol{\zeta}=\{\zeta_t\}_{t=1}^{o+d+1}$.

The zero set of the function f in domain $\mathcal{D}\subset\mathbb{R}^3$ is defined by

$$V(f)=\{(x,y,z)\in\mathcal{D}\mid f(x,y,z)=0\}, \tag{2}$$

known as *algebraic tensor-product B-spline surface*. For a fixed set of basis functions, the algebraic tensor-product B-spline surface is determined by its control coefficients.

In the dynamic implicit surface reconstruction, we assume the given data point set is $\{\boldsymbol{P}_i=(x_i,y_i,z_i)^\tau\}_{i=1}^{N}$, which represents the shape of some surface Γ. We are expected to look for a piecewise polynomial function, whose zero contour approximates the target shape.

For simplicity, the control coefficients and the basis functions are gathered (in a suitable ordering) into two column vectors, denoted by $\boldsymbol{f}$ and $\boldsymbol{q}(x,y,z)$ respectively. By the notation, $f(x,y,z)$ is rewritten in the form

$$f(x,y,z)=\sum_{l=1}^{L}f_lB_l(x,y,z)=\boldsymbol{q}(x,y,z)^\tau\boldsymbol{f},$$

where $\boldsymbol{f}=(f_1,\ldots,f_L)^\tau=(c_{111},\ldots,c_{mno})^\tau$ and $\boldsymbol{q}(x,y,z)=(B_1(x,y,z),\ldots,B_L(x,y,z))^\tau=(M_1(x)N_1(y)O_1(z),\ldots,M_m(x)N_n(y)O_o(z))^\tau$. Hence, we can express the value of f at a given point as $f(\boldsymbol{P}_i)=f(x_i,y_i,z_i)=\boldsymbol{q}_i^\tau\boldsymbol{f}$, and the gradient of f at $\boldsymbol{P}_i$ is $\nabla f(\boldsymbol{P}_i)=(\frac{\partial f}{\partial x}(\boldsymbol{P}_i),\frac{\partial f}{\partial y}(\boldsymbol{P}_i),\frac{\partial f}{\partial z}(\boldsymbol{P}_i))^\tau=(\boldsymbol{u}_i^\tau\boldsymbol{f},\boldsymbol{v}_i^\tau\boldsymbol{f},\boldsymbol{w}_i^\tau\boldsymbol{f})^\tau$.

A generic method consists of determining a surface that almost passes through (or approximates) a set of points and at the same time satisfies some application-dependent criteria, i.e., many conditions that measure the "quality" of the surface. A frequently used example is the simplified thin plate energy

$$\begin{aligned}Eng(\boldsymbol{f})&=\iiint_{\mathcal{D}}\|\nabla^2 f(x,y,z)\|^2_{\text{Frobenius}}\mathrm{d}x\mathrm{d}y\mathrm{d}z\\&=\boldsymbol{f}^\tau\boldsymbol{H}\boldsymbol{f},\end{aligned} \tag{3}$$

which is quadratic in the control coefficient vector $\boldsymbol{f}$. Here the symmetrical matrix $\boldsymbol{H}$ can be computed by Gauss integration.

2.2 Sampson Distance Error

Given a collection of unorganized data points $\{\boldsymbol{P}_i\}_{i=1}^{N}$ in 3D space, the aim is to generate an algebraic tensor-product B-spline surface $V(f)$ to fit the points. Firstly, one needs to define a meaningful metric

$$Err(\boldsymbol{f})=Err(\{\boldsymbol{P}_i\}_{i=1}^{N},V(f)),$$

as error function of data points set $\{\boldsymbol{P}_i\}_{i=1}^{N}$ to the implicit surface $V(f)$. We will prefer Sampson distance error which had been firstly used for conic fitting in [5].

The Sampson distance is defined as

$$\|\delta_{\boldsymbol{P}}\|=\frac{|f(\boldsymbol{P})|}{\|\nabla f(\boldsymbol{P})\|}, \tag{4}$$

which could be regarded as a first-order approximation of the geometric distance, and the sum of $\|\delta_{\boldsymbol{P}_i}\|^2$ be Sampson distance error:

$$Err_{\text{sps}}(\boldsymbol{f})=\sum_{i=1}^{N}\frac{f(\boldsymbol{P}_i)^2}{\|\nabla f(\boldsymbol{P}_i)\|^2}=\sum_{i=1}^{N}\frac{\boldsymbol{f}^\tau\boldsymbol{A}_i\boldsymbol{f}}{\boldsymbol{f}^\tau\boldsymbol{B}_i\boldsymbol{f}}, \tag{5}$$

where $\boldsymbol{A}_i=\boldsymbol{q}_i\boldsymbol{q}_i^\tau$ and $\boldsymbol{B}_i=\boldsymbol{u}_i\boldsymbol{u}_i^\tau+\boldsymbol{v}_i\boldsymbol{v}_i^\tau+\boldsymbol{w}_i\boldsymbol{w}_i^\tau$.

2.3 Algorithm Scheme

The algorithm scheme of the dynamic implicit curve/surface reconstruction is outlined as follows.

0) Input a given data points set $\{\boldsymbol{P}_i\}_{i=1}^{N}$. Specify $V(f^0)$ as the initial shape, and set $w_0>0$, $k:=0$.

1) Iteratively apply steps a.–b.–c. until the error reaches a predefined threshold or some stopping criteria are satisfied.

a. Approximate the objective function $R(\boldsymbol{f})=Err(\boldsymbol{f})+w_k Eng(\boldsymbol{f})$ at current coefficient vector $\boldsymbol{f}^k$ with a local quadratic model

$$Q^{(k)}(\boldsymbol{g})=R(\boldsymbol{f}^k)+\boldsymbol{b}_k^\tau\boldsymbol{g}+\frac{1}{2}\boldsymbol{g}^\tau\boldsymbol{M}_k\boldsymbol{g},$$

where $w_k=\sigma^k w_0$ with $0<\sigma<1$.

b. Calculate a trail step by solving the trust region subproblem:

$$\min Q^{(k)}(\boldsymbol{g})\quad\text{s.t.}\quad\boldsymbol{g}^\tau\boldsymbol{g}\leqslant e_k^2,$$

and get the solution $\boldsymbol{g}^k$. Here $e_k>0$ is a trust-radius at the k-th iteration.

c. Renew the coefficient vector with displacement $\boldsymbol{g}^k$ as setting $\boldsymbol{f}^{k+1}=\frac{\boldsymbol{f}^k+\boldsymbol{g}^k}{\|\boldsymbol{f}^k+\boldsymbol{g}^k\|}$, and let $k:=k+1$.

2) Output $V(f^{k+1})$ as the final approximation to the target shape model.

The trust region procedure ensures that the dynamic implicit curve/surface $V(f^k)$ deforms towards the target shape in a robust way. As for details, please refer to [4].

3 Generalized Eigenvector Fitting Model

Let $\{\boldsymbol{P}_i\}_{i=1}^{N}$ be the given data points. We are looking for a surface $V(f)$ with implicit representation to approximate the data points set. Then, the implicit surface approximation is modeled as minimizer of

$$\min_{\|\boldsymbol{f}\|=1}R(\boldsymbol{f})=\sum_{i=1}^{N}\frac{f(\boldsymbol{P}_i)^2}{\|\nabla f(\boldsymbol{P}_i)\|^2}+w\boldsymbol{f}^\tau\boldsymbol{H}\boldsymbol{f}. \tag{6}$$

The first part of the objective function is the Sampson distance error from the given data points $\{\boldsymbol{P}_i\}_{i=1}^{N}$

to the implicit surface $V(f)$. With the Sampson error $Err_{\text{sps}}(\boldsymbol{f})$ minimization, one would fit an implicit surface to the target shape of the data points. The second part is a fairing term $Eng(\boldsymbol{f})$ with certain non-negative weight w, and is used to enforce the fairness of the result surface. We add the fairing term to pull the approximating surface towards a simpler shape and keep away the unwanted branches.

Usually some well-established nonlinear optimization techniques, such as trust region methods, are suggested to solve the model. This numerical intractability can be lessened by an idea of successive minima technique, which is similar in spirit to those "iterative weighted least-squares algorithms" and "reweight procedure" that appear in the literatures[6,7]. First, we assume to have obtained an estimated $\boldsymbol{f}^{(k)}$, whose gradient gives a guess to $\nabla f(\boldsymbol{P}_i)$, and denote $\theta_{i,k} = 1/\|\nabla f^{(k)}(\boldsymbol{P}_i)\|$. The Sampson error can be replaced with $\sum_{i=1}^{N} \theta_{i,k}^2 f(\boldsymbol{P}_i)^2$ currently and the objective function of problem (6) turns into a quadratic form of the control coefficient vector $\boldsymbol{f}$. That is

$$Q^{(k)}(\boldsymbol{f}) = \boldsymbol{f}^\tau \boldsymbol{A}^{(k)} \boldsymbol{f}, \tag{7}$$

where $\boldsymbol{A}^{(k)} = \sum_{i=1}^{N} \theta_{i,k}^2 \boldsymbol{A}_i + w\boldsymbol{H}$ is a positive definite matrix. Second, since two implicit representations up to a scalar define the same surface, we induct a quadratic constraint on the control coefficients

$$\sum_{i=1}^{N} \|\nabla f(\boldsymbol{P}_i)\|^2 = \boldsymbol{f}^\tau \boldsymbol{B} \boldsymbol{f} = N \tag{8}$$

to obtain a nontrivial solution. The constraint function is data-dependent with the matrix $\boldsymbol{B} = \sum_{i=1}^{N} \boldsymbol{B}_i$ nonnegative definite.

Based on the former analysis, we will minimize (7) under the constraint in (8), and form a constrained optimization subproblem

$$\min Q^{(k)}(\boldsymbol{f}) = \boldsymbol{f}^\tau \boldsymbol{A}^{(k)} \boldsymbol{f} \quad \text{s.t.} \quad \boldsymbol{f}^\tau \boldsymbol{B} \boldsymbol{f} = N \tag{9}$$

at every iteration step in the successive minima method. From the principle of Lagrange-multiplier, the constrained optimization subproblem (9) reduces to a generalized eigenvector fitting model

$$(\boldsymbol{A}^{(k)} - \lambda \boldsymbol{B})\boldsymbol{f} = 0, \tag{10}$$

where λ is a generalized eigenvalue of $\boldsymbol{A}^{(k)}$ with respect to $\boldsymbol{B}$. The procedure based on the generalized eigenvector fitting should find the solution as a generalized eigenvector corresponding to the minimal generalized eigenvalue $\lambda_{\min}$ at a low cost.

With the above analysis, we outline the algorithm of generalized eigenvector fitting model as follows.

(a) Input the given data points $\{\boldsymbol{P}_i\}_{i=1}^{N}$, and generate the matrices $\boldsymbol{H}$ and $\boldsymbol{B}$. Set $w > 0$, $\varepsilon > 0$, $\{\theta_{i,0} = 1\}_{i=1}^{N}$, and $k := 0$.

(b) Compute $\theta_{i,k}$ and synthesize the matrix $\boldsymbol{A}^{(k)}$. Solve the subproblem (9) based on generalized eigenvector fitting, which gives a solution $\widetilde{\boldsymbol{f}}$. Resize the vector $\widetilde{\boldsymbol{f}} = (N/\widetilde{\boldsymbol{f}}^\tau \boldsymbol{B} \widetilde{\boldsymbol{f}})^{1/2} \widetilde{\boldsymbol{f}}$.

(c) Calculate the ratio $\rho = \frac{Err_{\text{sps}}(\widetilde{\boldsymbol{f}})}{Err_{\text{sps}}(\boldsymbol{f}^{(k)})}$, and set

$$\boldsymbol{f}^{(k+1)} = \begin{cases} \widetilde{\boldsymbol{f}}, & \text{if } \rho < 1; \\ \boldsymbol{f}^{(k)}, & \text{otherwise.} \end{cases}$$

(d) If $\rho < 1 - \varepsilon$, set $k := k + 1$ and go to step (b); otherwise, output the control coefficient vector $\widehat{\boldsymbol{f}} = \boldsymbol{f}^{(k+1)}$.

Thus, the control coefficient vector $\widehat{\boldsymbol{f}}$ would determine an implicit surface

$$V(\widehat{f}) = \{(x, y, z) | \widehat{f}(x, y, z) = \boldsymbol{q}(x, y, z)^\tau \widehat{\boldsymbol{f}} = 0\}$$

as a final approximation to the target shape model.

Here, we implement several curve examples to illustrate the effect of implicit curve approximation algorithm based on the generalized eigenvector fitting model.

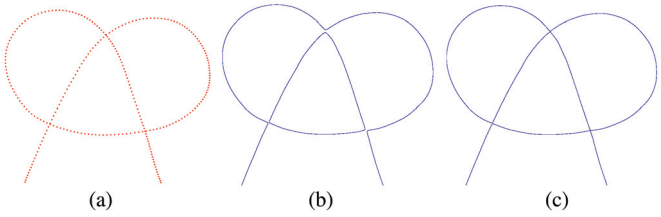


Fig.1. Curve example 1. (a) Data points. (b) After iter. 1. (c) After iter. 2.

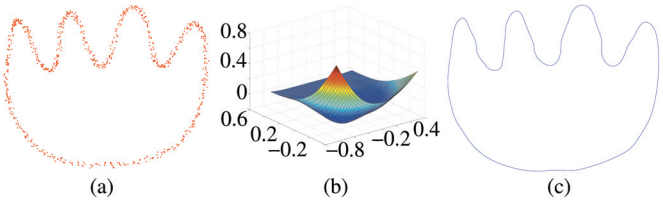


Fig.2. Curve example 2. (a) Data points. (b) Implicit func. (c) Result curve.

It should be noted that in all these examples, after two or more iterations, we will obtain a good approximation of the topology structure to the data points. But the geometric errors are large still, and the corresponding implicit representations are not stable. See Fig.2(b) for an example, where the most parts are flat, the implicit curves will collapse nearly to the plane when we want to reduce the geometric error by more iterations. The similar situations happen in the implicit surface fitting. This is the reason why we will combine the current model to the distance functions.

4 Distance Functions

In recent years, signed distance functions are widely applied in Computer Graphics for its good geometric properties. In this section we will review the definition

of signed distance function and some principal methods for computing distance function.

A signed distance function of a given surface Γ is one of its implicit descriptions. Let $\psi(x,y,z)$ be the signed distance function of the given surface. Then Γ is just the zero level set of implicit function ψ, that is $\psi(x,y,z)=0, (x,y,z)\in\Gamma$. For any points $(x,y,z)\notin\Gamma$, the function value $\psi(x,y,z)$ is the signed distance of point (x,y,z) to the given surface. Here the "distance" is the Euclidean distance. The Euclidean distance of a point $\boldsymbol{X}$ to the surface Γ is defined as follows:

$$d(\boldsymbol{X})=\mathrm{dist}(\boldsymbol{X},\Gamma)=\min_{\boldsymbol{P}\in\Gamma}\|\boldsymbol{X}-\boldsymbol{P}\|_{\mathrm{E}},$$

where $\|\boldsymbol{X}-\boldsymbol{P}\|_{\mathrm{E}}$ is the Euclidean distance between two points $\boldsymbol{X}$ and $\boldsymbol{P}$. The signed distance of a point to a surface is determined by the Euclidean distance and the topological relationship between the point and the surface. The absolute value of the signed distance is just the Euclidean distance. If the surface is open, then the signed distance is just "+" for any points in 3D space. If the surface is close, then the sign of the signed distance function is used to show whether a point is inside or outside of the surface. Usually the sign of the point inside the surface is supposed to be "−", otherwise "+". We call the function $d(x,y,z)=|\psi(x,y,z)|$ the distance function of the given surface. Usually, a singed distance function is generated in two steps: first computing the distance function, and then tagging it with proper signs.

It is time-consuming to compute distance functions directly by the definition. In fact, the distance function $d(x,y,z)$ of Γ is a solution of the so-called Eikonal equation:

$$\begin{cases}|\nabla d(x,y,z)|=1, & (x,y,z)\in\mathbb{R}^3\\ d(x,y,z)=0, & (x,y,z)\in\Gamma.\end{cases}\tag{11}$$

In real applications, a gridding will be given and only the distance function values at these grid points are computed. Usually there are two methods to compute the numerical solution of (11). The first method changes the Eikonal equation to a developing Hamilton equation by introducing a time variable and numerically solves the Hamilton equation by difference schemes. The second is the fast sweeping method proposed in [8]. Experiments show that the fast sweeping method is much more efficient than the first one.

The fast sweeping method is designed based on Gauss-Seidel iteration. For simplicity, let $\{(x_i,y_j,z_k)|1\leqslant i\leqslant I, 1\leqslant j\leqslant J, 1\leqslant k\leqslant K\}$ be the uniform gridding in solution region with space steps $\Delta x=\Delta y=\Delta z=h$. Assume that u^h stands for the numerical solution under the grid scale h. Then we outline the fast sweeping method into three parts as follows.

1) The discrete scheme of the Eikonal equation. For interior grid points, the upwind difference scheme is used,

$$[(u^h_{i,j,k}-u^h_{x\min})^+]^2+[(u^h_{i,j,k}-u^h_{y\min})^+]^2+[(u^h_{i,j,k}-u^h_{z\min})^+]^2=h^2,\tag{12}$$

where $i=2,\ldots,I-1$, $j=2,\ldots,J-1$, $k=2,\ldots,K-1$, and

$$\begin{aligned}u^h_{x\min}&=\min(u^h_{i-1,j,k},u^h_{i+1,j,k}),\\ u^h_{y\min}&=\min(u^h_{i,j-1,k},u^h_{i,j+1,k}),\\ u^h_{z\min}&=\min(u^h_{i,j,k-1},u^h_{i,j,k+1}).\end{aligned}$$

Here

$$(x)^+=\begin{cases}x, & x>0\\ 0, & x\leqslant 0\end{cases}$$

is the truncated power function. For those boundary grid points, one-sided difference scheme is used. Taking grid points with index $i=1$ for an example, the one-sided difference scheme is as follows:

$$[(u^h_{1,j,k}-u^h_{2,j,k})^+]^2+[(u^h_{i,j,k}-u^h_{y\min})^+]^2+[(u^h_{i,j,k}-u^h_{z\min})^+]^2=h^2.\tag{13}$$

2) Initialization. For those grid points which is on (or near) the surface Γ, assign them the accurate distance. These values keep unchanged after the fast sweeping procedure. For other grid points, assign a large enough positive number to them. These values will be updated after the fast sweeping procedure.

3) Gauss-Seidel iteration. At any grid point (x_i,y_j,z_k), compute the solution of (12) or (13) by taking $u^h_{i,j,k}$ as unknown. The solution is denoted to be $\widetilde{u}$. Then we update the distance value at grid point (x_i,y_j,z_k) by the minimum of original $u^h_{i,j,k}$ and $\widetilde{u}$: $u^h_{i,j,k}\leftarrow\min(u^h_{i,j,k},\widetilde{u})$. Several iterations will be done according to different loop orders of index (i,j,k). In detail, we make the following eight iterations:

3.1) $i=1:I$, $j=1:J$, $k=1:K$;
3.2) $i=1:I$, $j=1:J$, $k=K:1$;
3.3) $i=1:I$, $j=J:1$, $k=1:\mathrm{K}$;
3.4) $i=1:I$, $j=J:1$, $k=K:1$;
3.5) $i=I:1$, $j=1:J$, $k=1:K$;
3.6) $i=I:1$, $j=1:J$, $k=K:1$;
3.7) $i=I:1$, $j=J:1$, $k=1:K$;
3.8) $i=I:1$, $j=J:1$, $k=K:1$.

In the Gauss-Seidel iteration, we have to solve an equation as follows:

$$[(x-a)^+]^2+[(x-b)^+]^2+[(x-c)^+]^2=h^2$$

Without loss of generality, assume $a\leqslant b\leqslant c$. We find the unique solution

$$\bar{x}=\begin{cases}a+h, & \text{Case 1}\\ \dfrac{a+b+\sqrt{2(h^2-a^2-b^2)+(a+b)^2}}{2}, & \text{Case 2}\\ \dfrac{a+b+c+\sqrt{3(h^2-a^2-b^2-c^2)+(a+b+c)^2}}{3}, & \text{Case 3}\end{cases}$$

where

$$\text{Case 1}: b-a \geqslant h;$$
$$\text{Case 2}: b-a < h \text{ and } (c-a)^2+(c-b)^2 \geqslant h^2;$$
$$\text{Case 3}: (c-a)^2+(c-b)^2 < h^2.$$

For the given gridding, the time complexity of the fast sweeping algorithm is $O(IJK)$. Besides this, the generalization to other dimensional problem is straightforward. It takes only 2^ν iterations for ν dimensional problem.

The distance function describes the Euclidean distance between any point in the space and the implicit surface. It represents the geometry of the given target model perfectly, but lacks information of topology. The main shortcoming is that the distance function cannot indicate inside or outside of a close curve or surface. Hence the signed distance function is needed.

To generate the signed distance function on a given gridding, the tagging procedure based on recursion presented in [8] is hard and expensive. The recursion depth is constrained by the memory size and the stack often overflows for 3D problems. Furthermore, for the target shapes with complex structure of topology, an improperly tagged signed distance function would lead to the shape with incorrect topology, as shown in Fig.3. Instead of recursive tagging procedure, we obtain the sign on grid points via the result function $\widehat{f}$ of generalized eigenvector fitting model in Section 3.

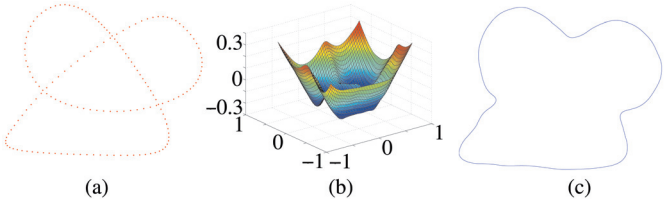


Fig.3. Curve example 3. (a) Data points. (b) Signed dist. function. (c) Result curve.

5 Synthesized Scheme of Initial Shapes Specification

The result surface obtained from the generalized eigenvector fitting model could reflect the complicated topology of data set with possibly large geometric error. On the other hand, the distance function represents the geometry of the target shape perfectly.

For simplicity, let $\{\boldsymbol{X}_{ijk}=(x_i,y_j,z_k)\}$ be the uniform gridding in the solution region. First, we compute the distance function of $\Gamma=\{\boldsymbol{P}_i\}_{i=1}^N$ on the gridding without signs. By assembling the distance function with the sign of result function $\widehat{f}$ obtained in Section 3, then we define an implicit function

$$\phi(\boldsymbol{X}_{ijk})=\text{sign}[\widehat{f}(\boldsymbol{X}_{ijk})]\cdot d(\boldsymbol{X}_{ijk}). \tag{14}$$

One can fit the function value of ϕ at these fixed grid points $\{\boldsymbol{X}_{ijk}\}_{i,j,k=1}^{I,J,K}$ with least-squares method. Let $f(x,y,z)=\sum_{l=1}^L f_l B_l(x,y,z)$ be the tensor-product B-spline function with user-specified basis functions $\{B_l(x,y,z)\}_{l=1}^L$ and unknown control coefficient vector $\boldsymbol{f}=(f_1,\ldots,f_L)^\tau$. Solving the least-squares model

$$\min \sum_{i,j,k=1}^{I,J,K}(f(\boldsymbol{X}_{ijk})-\phi(\boldsymbol{X}_{ijk}))^2+wEng(\boldsymbol{f}), \tag{15}$$

we get an implicit function $\overline{f}(x,y,z)$ with coefficient vector $\overline{\boldsymbol{f}}$, and immediately obtain a good specification of the initial shape $V(\overline{f})$ for the dynamic implicit surface reconstruction.

We have applied our new scheme to some data examples. Comparing the result in Fig.2, where only generalized eigenvector fitting is used, the result in Fig.4 is more stable. Note the difference between two graphs in Figs. 2(b) and 4(b). More curve and surface examples are given in Figs. 5, 6, and 7. The computational statistics of surface examples by a PIV-2.2GHz PC are listed in Table 1, including the iterations and time for generalized eigenvector fitting model (GEFM), and the grids and time for distance function computing (DFC).

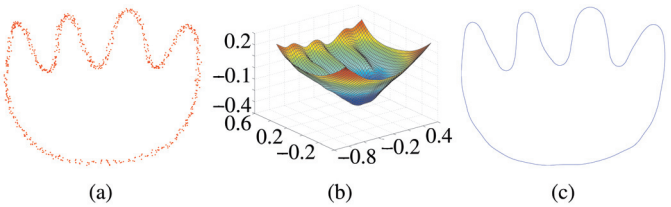


Fig.4. Example 2 by synthesized scheme. (a) Data points. (b) Implicit func. (c) Init. curve.

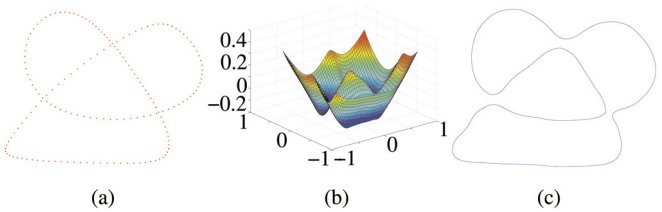


Fig.5. Example 3 by synthesized scheme. (a) Data points. (b) Implicit func. (c) Init. curve.

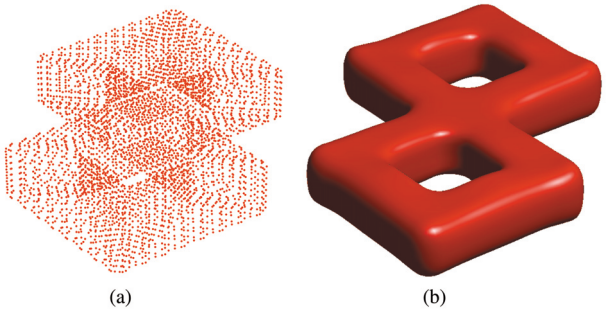
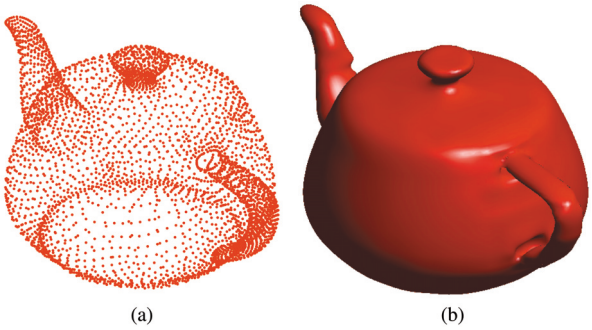


Fig.6. 2torus by synth. scheme. (a) Data points. (b) Init. surface.



(a) (b)

Fig.7. Teapot by synth. scheme. (a) Data points. (b) Init. surface.

Table 1. Computational Statistics of 3D Examples

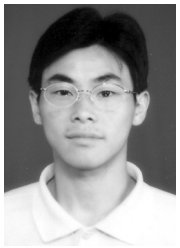
Exam.	# points	GEFM		DFC	
		iter.	time (s)	grids	time (s)
2torus	4352	3	5.0	$71 \times 71 \times 13$	2.1
Teapot	4255	3	6.2	$64 \times 41 \times 32$	2.7

6 Conclusions

A scheme of initial shapes specification has been developed in the paper, which is based on the generalized eigenvector fitting model and the distance function. The generalized eigenvector fitting model may perfectly reflect the topology structure of a given target shape. The distance functions can give the geometric information accurately. For a given data points set, we synthesize an implicit function by assembling the distance function with the sign obtained through generalized eigenvector fitting, and give a good specification of initial shapes for the dynamic implicit surface reconstruction.

References

[1] Farin G. Curves and Surfaces for CAGD — A Pratical Guide. 5th Edition, Morgan Kaufmann Publishers, 2002.
[2] Bloomenthal J *et al.* Introduction to Implicit Surfaces. Morgan Kaufmann Publishers, 1997.
[3] Jüttler B, Felis A. Least-squares fitting of algebraic spline surfaces. *Advances in Computational Mathematics*, 2002, 17: 135–152.
[4] Yang Z W, Deng J S, Chen F L. Dynamic implicit curve reconstruction based on approximate geometric distance. *Journal of Software*, 2004, 15(Suppl): 264–272.
[5] Sampson P D. Fitting conic sections to very scattered data: An iterative refinement of the Bookstein algorithm. *Computer Vision, Graphics and Image Processing*, 1982, 18: 97–108.
[6] Ramsay J. A comparative study of several robust estimates of slope, intercept, and scale in linear regression. *Journal of Amer. Stat. Assoc.*, 1977, 72: 608–615.
[7] Taubin G. Estimation of planar curves, surfaces, and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 1991, 13: 1115–1138.
[8] Zhao H K. A fast sweeping method for Eikonal equations. *Mathematics of Computation*, 2004, 74: 603–627.



Zhou-Wang Yang was born in Fujian, China, in 1975. He received his Ph.D. degree from the University of Science and Technology of China in 2005. His research interests include computer graphics and optimization theory.

Chun-Lin Wu was born in Jiangxi, China, in 1982. He is a Ph.D. candidate in University of Science and Technology of China. His research interests include numerical methods for PDEs, image processing and computer graphics.

Jian-Song Deng was born in Shandong, China, in 1971. He received his Ph.D. degree from the University of Science and Technology of China in 1998. His research interests include computer aided geometric design and computer graphics.

Fa-Lai Chen was born in Anhui, China, in 1966. He received his Ph.D. degree from the University of Science and Technology of China in 1994. He is a professor in the Department of Mathematics at the University of Science and Technology of China. His research interests include computer aided geometric design and computer graphics.