Xiao CX. Multi-level partition of unity algebraic point set surfaces. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 26(2): 229–238 Mar. 2011. DOI 10.1007/s11390-011-1125-8

Multi-Level Partition of Unity Algebraic Point Set Surfaces

Chun-Xia Xiao (肖春霞), *Senior Member*, *CCF*, *Member*, *ACM*

School of Computer, Wuhan University, Wuhan 430072, China

E-mail: cxxiao@whu.edu.cn

Received August 24, 2010; revised December 5, 2010.

Abstract We present a multi-level partition of unity algebraic set surfaces (MPU-APSS) for surface reconstruction which can be represented by either a projection or in an implicit form. An algebraic point set surface (APSS) defines a smooth surface from a set of unorganized points using local moving least-squares (MLS) fitting of algebraic spheres. However, due to the local nature, APSS does not work well for geometry editing and modeling. Instead, our method builds an implicit approximation function for the scattered point set based on the partition of unity approach. By using an octree subdivision strategy, we first adaptively construct local algebraic spheres for the point set, and then apply weighting functions to blend together these local shape functions. Finally, we compute an error-controlled approximation of the signed distance function from the surface. In addition, we present an efficient projection operator which makes our representation suitable for point set filtering and dynamic point resampling. We demonstrate the effectiveness of our unified approach for both surface reconstruction and geometry modeling such as surface completion.

Keywords moving least squares, surface reconstruction, implicit modeling, partition of unity approximation

1 Introduction

Multi-media have seen three waves so far: sound, images, and video. The arrival of 3D scanning has created the fourth digital media: 3D geometry, also called 3D geometric models. Applications of 3D geometry already cover a wide range of areas from multimedia, entertainment and computer graphics, to biomedical computing, reverse engineering and scientific computing. This new multi-media wave brings with it the need to design efficient algorithms for the acquisition, reconstruction, analysis, manipulation, simulation and transmission of complex 3D models. Among these algorithms, surface reconstruction from the raw range data is one of the most fundamental and challenging tasks.

Digital scanning devices are now commonly used to acquire high-resolution 3D models. Current scanners are able to produce large amounts of raw, dense point sets. One of the principal challenges faced today is the development of surface reconstruction techniques which deal with the inherent noises of the acquired dataset. Many surface reconstruction methods have been proposed in recent years, among these methods, point set surfaces (PSS) which are defined by local moving least-squares (MLS) approximations of the data[1-2] have been proven to be a powerful approach for meshless surface representation. PSS is one of the most flexible surface representations for point sets and has been successfully used in a wide range of applications[3].

Initial Levin's definition[1] and PPS definition[2] are relatively expensive to compute. Although significant progress[4-5] has been made to design simpler and more efficient definitions, the central limitation of the robustness of PSS is the plane fit operation that is highly unstable in regions of high curvature where the sampling rate drops below a threshold. Recently, Guennebaud *et al.*[6-7] proposed an algebraic point set surfaces (APSS) framework to locally approximate the data using algebraic spheres. Compared with MLS approximations[2], this strategy exhibits high tolerance with respect to low sampling densities while retaining a tight approximation of the surface.

Although algebraic point set surfaces (APSS)[6] are successful for representing point set surface, APSS also has its own drawbacks. One problem is that algebraic point set surfaces are defined locally, which make it inconvenient for geometry modeling operations such as shape blending and deformations, and the local feature also makes the approximation accuracy for the point set

Regular Paper

This work was partly supported by the National Natural Science Foundation of China under Grant Nos. 60803081, 61070081, the National High Technology Research and Development 863 Program of China under Grant No. 2008AA121603, the Fundamental Research Funds for the Central Universities under Grant No. 6081005, and the National Research Foundation for the Doctoral Program of Higher Education of China under Grant No. 200804861038.

not easy to control. Although Guennebaud *et al.*[7] presented a real-time rendering method using a dynamic upsampling strategy, this approach does not work well for surface reconstruction of incomplete data with large holes due to its local nature. Another drawback of APSS[7] is that a local MLS approximations is computed for each point, thus, it is not efficient enough to process large point set.

To overcome the above problems, inspired by multi-level partition of unity implicit (MPU) techniques[8-9], we present an MPU algebraic point surfaces which can be represented by either a projection or an implicit form. We use the partition of unity approach to build an implicit approximation function for the scattered point set. We first use an octree subdivision strategy to adaptively construct local algebraic spheres for the point set, and then apply weighting functions to blend together these local shape functions, resulting in an error-controlled approximation of the signed distance function from the surface. The signed distance function makes the degree of the approximation to be easily controlled, and is convenient for surface reconstruction (polygonization), which is the zero level set isosurface extraction of the constructed distance function. We also present an efficient and robust projection operator, making our approach robust even for filtering noisy data and point set resampling.

Our method has the advantages of both multi-level partition of unity implicits method[9] and algebraic point set surfaces (APSS)[6]. As a consequence, compared with PSS definition[2], our method is simpler and more robust at the surface reconstruction at the regions of high curvature. Compared with [9], algebraic point set surface we applied is more robust. The initial MPU approach[9] uses either bi-variate polynomials when the data are flat enough, or tri-variate implicit quadrics otherwise, however, both this selection procedure and the fitting of implicit quadrics (which not very robust) make the approach more complicated and less stable. Compared with the APSS[6-7], our method is much faster at surface reconstruction, since we apply an octree subdivision method adapting to the complexity of the local surface shape. In addition, our method is more convenient than APSS[6-7] for geometry modeling such as surface completion. A shorter version of this paper appeared in [10].

The rest of our paper is organized as follows. Section 2 reviews related work. Section 3 gives a technical description for multi-level partition of unity algebraic point set surfaces (MPU-APSS). In Section 4, we propose dynamic point resampling, and in Section 5, we give the experimental results and discussions. Finally, we conclude our paper in Section 6.

2 Related Work

Point Set Surface (PSS)[2] is an efficient smooth surface representation for point sets, which is constructed using Levin's moving least squares projection operator[1]. The projection is an iterative procedure where at each step the point is projected onto a polynomial approximation of the neighboring data, and the polynomial approximation is fitted from a local reference plane computed by a non-linear optimization. PSS also has been applied in geometry modeling. Pauly *et al.*[11] introduced a point-based modeling system, in which they defined an implicit PSS function to perform constructive solid geometry (CGS) operations on objects. By omitting the polynomial fitting step, Amenta and Kil[4] showed that the same surface can be computed by weighted centroids and a smooth gradient field, this method is also extended to achieve convex interpolation using Hermite centroid evaluations[12]. Fleishman *et al.*[13] presented a robust MLS technique for reconstructing a piecewise smooth surface from a noisy point set, they used techniques from robust statistics to guide the creation of the neighborhoods used by the MLS computation.

Rather than fitting a plane, inspired by [14], Guennebaud *et al.*[6] defined the surface by means of sphere fitting which significantly improves the robustness against low sampling density. For PSS[2], even though fitting polynomials allows achieving tighter approximations, the approach fails when the data cannot be locally represented as a height field. Another limitation of the robustness of PSS[2] is that the plane fit operation becomes highly unstable when the sampling rate drops down. However, moving least squares of fitting of algebraic spheres performs better at the regions of high curvature. Recently, Guennebaud *et al.*[7] provided a more generic solution that includes intuitive parameters for curvature control of the fitted spheres, and also presented a real-time rendering system for such surfaces using a dynamic up-sampling strategy combined with a conventional splatting algorithm.

Many surface construction methods have been proposed. Pioneering work in surface reconstruction was done by Hoppe *et al.*[15], who created a piecewise smooth surface in a multi-phase process that was based on implicit modeling of a distance field. Amenta *et al.*[16] presented a surface reconstruction in computational geometry view, however, their method usually fails in both noisy and undersampled models or models with sharp features. An alternative approach is to interpolate a set of points with radial basis functions (RBF) which offers a smooth object representation. Typically this requires the minimization of a thin-plate spline

energy functional. Computing an RBF interpolation is performed by solving a system of equations of the size up to $3N \times 3N$, where N is the number of input points. Both Carr *et al.*[17] and Morse *et al.*[18] presented a fast solver for RBF which greatly reduced the complexity of RBF. Dinh *et al.*[19] used a nonsymmetric RBF function which aimed at capturing sharp features. They identified edges using covariance analysis of a neighborhood of a point to determine the shape of the function assigned to the point. Kazhdan *et al.*[20] reconstructed surface from oriented points by casting surface reconstruction as a spatial Poisson problem. Although details can be reconstructed better, however, as this method has to solve a large linear system, which makes it not efficient to process large point data.

Ohtake *et al.*[9] introduced an implicit function surface representation defined by a blend of locally fitted implicit quadrics (MPU). Each quadric approximates points in a local neighborhood, although the quadrics fitting alleviate the limitations of polynomial fitting, the quadrics are fitted by adding a few point constraints away the surface neglecting the fact that the algebraic distance is not linear. Xie *et al.*[21] extended the MPU technique to handle noisy datasets, and they described separate procedures for outlier detection and noise removal for robust surface reconstruction. Another related approach is sparse low-degree implicit (SLIM)[22] where the geometry consists of points equipped with bivariate polynomials. Efficient rendering is accomplished by blending the primitives in screen space. However this approach still suffers from the polynomial fitting limitations and does not properly define a smooth surface since it depends on the view direction.

Sometimes, the density of the point set might not be sufficient for high quality rendering or surface reconstructions, then the point sets have to be upsampled. Particle simulation procedure[23-24] have been widely used to control the sampling density of a point cloud spread over an implicit surface. Guennebaud *et al.*[25-26] presented iterative refinement schemes for upsampling, while these methods are too costly to be able to handle dynamic data. Xiao *et al.*[27] re-sampled the in-between morphing object dynamically and adaptively using MLS surfaces. Alexa *et al.*[2] presented an upsampling algorithm where the tangent plane of each input sample is uniformly upsampled and the sampled points are projected onto a precomputed polynomial approximating the underlying surface. Although this method is difficult to generate near-uniform sampled surface, this scheme can be efficient in parallel GPU implementation. Based on this upsampling procedure, Guennebaud *et al.*[7] provided a real-time rendering system using a dynamic up-sampling strategy, in which the generated splats are projected onto the APSS to make a smooth surface. However, since the APSS are defined locally, for incomplete point sets with large holes, this upsampling method does not work well. However, as our MPU algebraic point set surfaces blend the local shape approximations for the point sets, we can upsample and compete the point set with large missing data.

Different from completion methods based on surface reconstruction, Xiao *et al.*[28] presented a new approach for appearance and geometry completion over point-sampled geometry. They converted the problem of context-based geometry completion into a task of texture completion on the surface. By using point set surfaces filtering[29], the geometric detail is peeled from the models and is converted into a piece of signed gray-scale texture on the base surface of the point set surface. Point set surfaces aggregation[30] and segmentation[31] also can help to perform surface completion.

3 MPU Algebraic Point Set Surfaces

The partition of unity approach[8-9] is typically used to integrate locally defined approximates into an implicit approximation. Inspired by the partition method[8-9], which makes reconstruction of huge models or complex surface possible and efficient, we now present an MPU based algebraic point set surface, in which we employee Guennebaud's algebraic sphere[6-7] as the only local approximating function, and from which we benefit not only robustness but also simplicity.

3.1 Algebraic Point Set Surfaces

The key idea of APSS[6] is to locally approximate the point cloud by a fitted algebraic sphere that moves continuously in space. Let point sets $P = \{p_i \in \mathbb{R}^3\}$ be equipped with normals $\boldsymbol{n}_i$ and r_i radii representing the local point spacing. The normals $\boldsymbol{n}_i$ can be estimated from the initial scans during the point set acquisition, or can be computed using local least-squires fitting[32]. The radii r_i can be computed using a local estimation of the density. These radii are used to define the following adaptive weighting scheme $w_i(x) = \phi((\|p_i - x\|)/h_i(x))$ describing the weight of the point p_i for any point x. Here, h is a global scale factor allowing to adjust the influence radius of every point, and ϕ is a smooth, decreasing weight function.

An algebraic sphere is defined as the 0-isosurface of the scalar field $S_{\boldsymbol{u}(x)}(x) = [1, x^{\mathrm{T}}, x^{\mathrm{T}}x]\boldsymbol{u}$, where $\boldsymbol{u} = [u_0, \ldots, u_4]^{\mathrm{T}}$ is the vector of scalar coefficients describing the sphere. For $u_4 \neq 0$, the corresponding center $\boldsymbol{c}$ and r radius is computed as $\boldsymbol{c} = -\frac{1}{2u_4}[u_1, u_2, u_3]$,

and $r = \sqrt{\boldsymbol{c}^{\mathrm{T}}\boldsymbol{c} - u_0/u_4}$. The APSS S_p approximating the point cloud P yields as the zero set of an implicit scalar field $g(x)$ representing the distance between the evaluation point x and a locally fitted algebraic sphere $\boldsymbol{u}(x)$:

$$g(x) = S_{\boldsymbol{u}(x)}(x) = [1, x^{\mathrm{T}}, x^{\mathrm{T}}x]\boldsymbol{u}(x). \tag{1}$$

The sphere $\boldsymbol{u}(x)$ is obtained by minimizing given distances between itself and the neighbors of x in a weighted least square sense. The original algorithm minimizes the positional constraints $S_{\boldsymbol{u}}(p_i) = 0$ and the derivative constraints $\nabla S_{\boldsymbol{u}}(p_i) = \boldsymbol{n}_i$ simultaneously such that:

$$\boldsymbol{u}(x) = \arg\min \sum_i w_i(x)(S_{\boldsymbol{u}}(p_i)^2 + \|\nabla S_{\boldsymbol{u}}(p_i) - \boldsymbol{n}_i\|^2). \tag{2}$$

This minimization yields a standard system of linear equations. Guennebaud *et al.*[7] proposed a generalized fitting procedure to solve (2) by minimizing the two key constraints (positional, derivative) separately and starts with the derivative constraints, and obtain the following explicit solution for the coefficients of $u(x)$:

$$u_4 = \beta\frac{1}{2}\frac{\sum w_i p_i^{\mathrm{T}}\boldsymbol{n}_i - \sum \tilde{w}_i p_i^{\mathrm{T}} \sum w_i \boldsymbol{n}_i}{\sum w_i p_i^{\mathrm{T}} p_i - \sum \tilde{w}_i p_i^{\mathrm{T}} \sum w_i p_i} \tag{3}$$

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \sum \tilde{w}_i \boldsymbol{n}_i - 2u_4 \sum \tilde{w}_i p_i \tag{4}$$

$$u_0 = -[u_1\ u_2\ u_3] \sum \tilde{w}_i p_i - u_4 \sum \tilde{w}_i p_i^{\mathrm{T}} p_i \tag{5}$$

where $w_i = w_i(x)$ and $\tilde{w}_i$ is the normalized weight of the sample p_i: $\tilde{w}_i = w_i / \sum_j w_j$. β is an additional scalar parameter that, for the time being, is equal to 1. The equations for the gradients are omitted, but can be derived easily.

3.2 MPU Algebraic Point Set Surfaces

The partition of unity approach[8] is used to build a global function from a set of locally defined functions, which resembles the behavior of each function within their respective domains of influence. The basic idea of the partition of unity approach is to break the data domain into several pieces, approximate the data in each subdomain separately, and then blend the local solutions together using smooth, local weights that sum up to one everywhere on the domain, more details refer to [9].

We now employ the partition of unity approach to build an approximation and interpolation function for the scatted scanned scattered points. We first compute the bounding cube Ω of the points in M, then we apply an adaptive octree-based subdivision to Ω. Let $u_i(x)$ be the locally fitted function defined in vicinity c_i (the center of the cell) using the algebraic sphere. The domain of influence of each local function is designated by assigning to it a nonnegative weight function $w_i(x)$. So the result blended function defined on Ω by the partition of unity approach can be formulated as:

$$f(x) = \frac{\sum_{i=1}^n w_i(x)u_i(x)}{\sum_{i=1}^n w_i(x)}. \tag{6}$$

Let $\varphi_i(x) = w_i(x)/\sum_{i=1}^n w_i(x)$, then the sum of this set of nonnegative compactly supported functions $\{w_i(x)\}$ is unity:

$$\sum_{i=1}^n \varphi_i(x) \equiv 1 \quad \text{on} \quad \Omega. \tag{7}$$

(6) and (7) constitute the core of the our function approximation employing partition of unity approach[8] as shown in Fig.1. If an approximation of M is required, we use the quadratic B-spline $b(t)$ as [9] to generate weight functions, $w_i = b(3|x - c_i|/2R_i)$, if an interpolation of M is required, we use the inverse-distance singular weights[33].

Since the important properties of the blending function $f(x)$ such as the maximum error and convergence order are inherited from the local behavior, to construct a recursive procedure for assembling an MPU approximation at point x with precision ε_0, we first computed a local shape function $u_i(x)$ for a cell within a specified threshold. We estimate a local max-norm approximation error applying the Taubin distance[34] $\varepsilon = \max|u(p_i)|/|\nabla u(p_i)|$, if ε is greater than a user-specified threshold ε_0, the cell is subdivided and the fitting process is performed for the child cells.

Similar to [9], we blend these local shape functions to build an implicit approximation function. By using

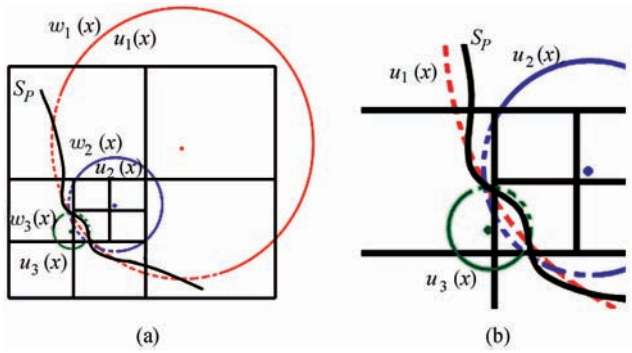

Fig.1. (a) Blending of a set of locally defined functions using partition of unity approach. The resulting function (black solid curve S_p) is constructed from three local APSS functions ($u_1(x), u_2(x)$, and $u_3(x)$) with their associated weight functions ($w_1(x), w_2(x)$, and $w_3(x)$). (b) Shows in a clear way of the procedure.

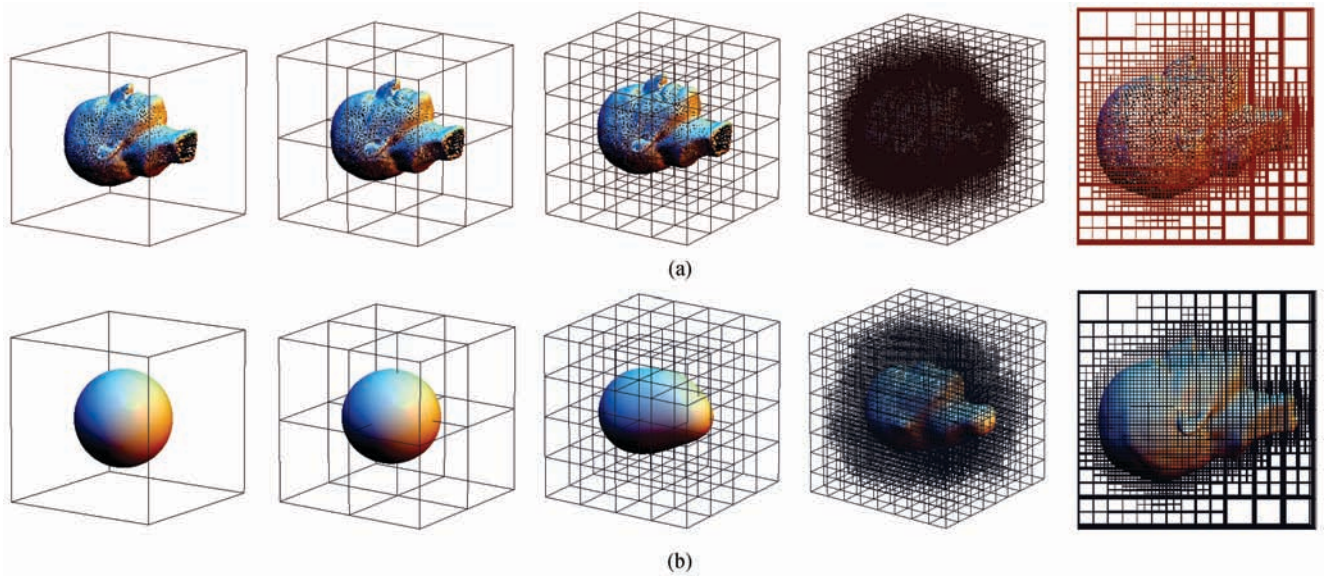

Fig.2. Multi-level partition of unity algebraic point set surface for multi-level surface. Based on the hierarchical octree division of pint set (a), the surface is reconstructed at different level (b).

changing precision ε, it enables us to compute an error-controlled approximation of the signed distance function from the surface. With the implicit distance function, we can reconstruct surface (polygonization) for the underlying point set which is the zero level set iso-surface extraction of the distance function as shown in Fig.2.

Besides the surface reconstruction, our MPU-APSS is able to perform surface completion. Guennebaud *et al.*[7] have provided a dynamic up-sampling strategy, in which the generated splats are projected onto the APSS to make a smooth surface. However, the APSS[6] defined locally makes this upsampling method not work well for completing point sets with large holes. Since our MPU algebraic point set surfaces blend the local shape approximation surfaces for the point sets, we can upsample and compete the point set with large missing data. The comparison results are shown in Fig.10.

4 Projection Operator and Resampling

Projection operator is useful for generating a smooth surface from the noisy point set. Practical problems of MLS surfaces[2] are a complicated non-linear optimization to compute a tangent frame and the normal to this tangent frame is usually not the surface normal. Similar to MLS surfaces, the sphere normal can differ from the actual surface normal. In this section, we use MPU algebraic point set surface definition to present simple, efficient projection operators, and to compute the exact normal of the surface. The projection operators are useful for filtering noisy point-sampled geometry and performing the re-sampling processing.

4.1 Projection Operator

Since our MPU algebraic point set surfaces is defined by blending the local shape approximations, we first come to the definition of the projection operator for one single algebraic sphere. We extend the almost orthogonal projection in [5] to perform the projection procedure for algebraic PSS. For an implicit function f, let $\boldsymbol{n}(x)$ and $a(x)$ define a tangent frame with origin in $a(x)$, then the projection $Q(x)$ of x onto the tangent is computed as:

$$Q(x) = x - \boldsymbol{n}(x)^{\mathrm{T}}(a(x) - x)\boldsymbol{n}(x). \tag{8}$$

The APSS S_p approximates or interpolates the point set $P = \{p_i \in \mathbb{R}^d\}$, and yields as the zero set of the implicit scalar field $f(x)$, where $f(x) = S_{\boldsymbol{u}(x)}(x) = [1, x^{\mathrm{T}}, x^{\mathrm{T}}x]\boldsymbol{u}(x)$ represents the algebraic distance between the evaluation point and the fitted sphere $\boldsymbol{u}(x)$.

Suppose we project point onto the algebraic sphere $\boldsymbol{u}(x)$, we compute $a(x)$ as the weighted average of points in the neighborhood $N(x)$ of x:

$$a(x) = \frac{\sum_{i=0}^{N-1} \theta(\|x - p_i\|)p_i}{\sum_{i=0}^{N-1} \theta(\|x - p_i\|)}. \tag{9}$$

The surface normal $\boldsymbol{n}(x)$ corresponding to x is defined as the gradient of the implicit $\boldsymbol{u}(x)$: $\boldsymbol{n}(x) = \nabla f(a(x))$.

Inspired by [5], for a given point x, we can adapt the projection procedure to make it almost orthogonal. By almost orthogonal we mean that the projection is in direction of $\boldsymbol{n}(x')$, if x' is the projection. The projection operator for APSS works as follows: the projection

always considers the original point x and not the intermediate points x'. Fig.3 shows two steps of an almost orthogonal projection of a point x onto the surface. Since $f(x') = \boldsymbol{n}(x')^{\mathrm{T}}(a(x') - x') \approx 0$, the terminated point is a point x' on the surface. More specifically, the following Table 1 gives the pseudocode that computes an orthogonal projection of x.

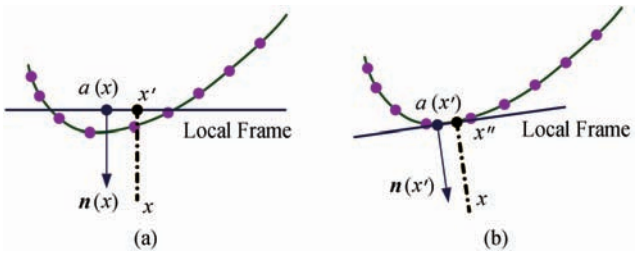

Fig.3. The first two steps of an almost orthogonal projection of a point x onto the surface. In each step the current approximation x' is used to build an orthogonal tangent frame using $\boldsymbol{n}(x')$ and $a(x')$, onto which x is projected to get a new approximation.

Table 1. Pseudocode for Computing an Almost Orthogonal Projection of x

1.	Set $x' = x$.
2.	Compute $a(x')$ and $\boldsymbol{n}(x')$.
3.	Set $x' = x - \boldsymbol{n}(x')^{\mathrm{T}}(a(x') - x)\boldsymbol{n}(x')$.
4.	If $\|\boldsymbol{n}(x')^{\mathrm{T}}(a(x') - x')\| > \varepsilon$ go back to 2.

Then we come to the almost orthogonal projection for MPU algebraic PSS. As our MPU algebraic point set surfaces are defined by blending the local shape approximations, to define a projector for MPU algebraic PSS, instead of projecting x to the sphere it associates with during the MPU-APSS construction, we project the point x to the underlying MPU algebraic PSS $f(x)$ (see (6)), which is weighted average of the implicit spheres. Note that we apply decent gradient method to compute the normal $\boldsymbol{n}(x)$. Then using the iteration method presented in Table 1, we obtain the final projected point. The almost orthogonal projection for MPU-APSS is simple and efficient, and has reasonable convergence.

The almost orthogonal projection is useful for point set filtering. To make a smooth surface from the noisy point set, we can first compute an MPU-APSS based on the given precision for the underlying point sets, and then we project the point onto the underlying surface to obtain a smooth surface, as illustrated in Fig.9. This almost orthogonal projection is simple and efficient, converges fast, and provides a good coverage when the tangential components of the input samples do not overlap.

4.2 Re-Sampling and Completion

When the density of the point set is not sufficient enough for high quality rendering or surface reconstruction, the point set then have to be upsampled. With the implicit function built for the scattered point sets, and the proposed projection operator, we now come to the point set re-sampling operation. To obtain a complete geometry model, our upsampling procedure is performed in the following three steps. (a) Density estimation for detecting regions of insufficient sampling, (b) adaptive upsamping of the neighborhood of each detected points, and (c) projection of the sampled points on the MPU-APPS.

To detect regions with insufficient sampling density dynamically, we estimate and record the local sampling density for each point of the object. We estimate the local sampling density ρ_i for each $p_i \in P$ by finding the sphere with minimum radius r_i centered at p_i that contains the k-nearest neighbors to p_i, then ρ_i is defined as $\rho_i = k/{r_i}^2$. Let σ be the density threshold value for up-sampling, if $\rho < \sigma$, new sample points must be inserted in the neighborhood of p_i.

Once the point p_i of insufficient sampling density is detected, the points N_i nearby are projected onto a plane originated at p_i with normal $\boldsymbol{n}_i$. A bounding rectangle is established for the projected points, and points are uniformly re-sampled in the rectangle according to a user specified threshold[2], as shown in Fig.4. Then the MPU-APSS is constructed for the original region containing the point p_i, using the almost orthogonal projection for MPU-APSS, the re-sampled points are projected onto the underlying MPU-APSS to achieve the final up-sampling result. The normal at each sampled point can be computed directly by evaluating the gradient of the MPU-APSS. As shown in Fig.11, the up-sampling techniques eliminate the insufficient sampling regions.

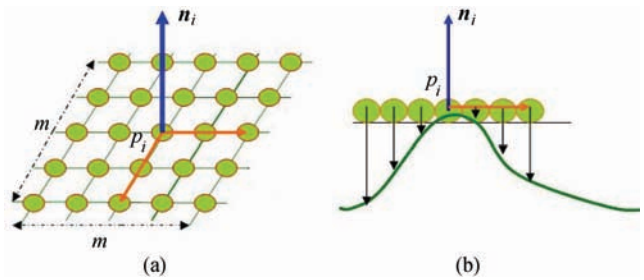

Fig.4. MPU-APSS resampling, for each point p_i, we first generate $m \times m$ sampled points according to its local point spacing and user's required density, then we project these $m \times m$ points onto the implicit MPU-APSS surface with the almost projection operator.

5 Experimental Results and Discussions

To demonstrate the efficiency of our method we

have implemented lots of examples including surface reconstruction (polygonization), geometry reconstruction for the incomplete scattered geometry with large holes, geometry reconstruction for the point set with heavily noise, we also present surface upsampling examples. In addition, we provide the comparison results with other surface reconstruction methods such as [6, 9]. We measure the performance of these algorithms on a machine of an Intel Core 2 Duo 2.6 GHz with 2 GB of RAM.

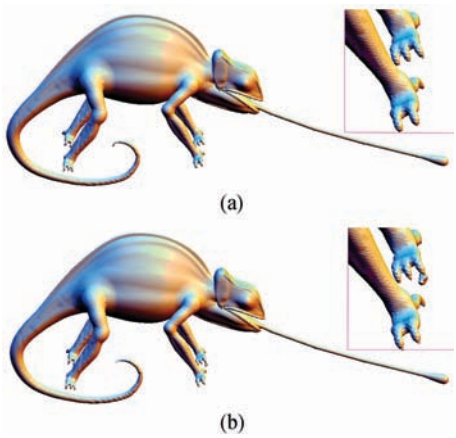

Fig.5. (a) Reconstructed surface (polygonization) using our MPU-APSS. (b) Reconstructed surface using MPU (Ohtake *et al.*, 2003[9]).

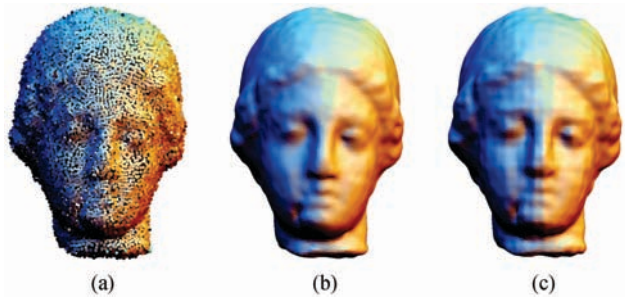

Fig.6. Surface reconstruction (polygonization) comparison from the scatted point sets. (a) Original point set. (b) Reconstructed surface using MPU-APSS. (c) Reconstructed surface using MPU (Ohtake *et al.*, 2003[9]).

In Fig.5 and Fig.6, we show the surface reconstruction (polygonization) and compare with [9]. Ohtake *et al.*[9] use one of these three local approximations to fit the points in a cell: (a) a general 3D quadric, (b) a bivariate quadratic polynomial, and (c) a piecewise quadric surface. The local shape function selection depends on the point set distribution in the cell, for example, they use piecewise quadric surface to reconstruct sharp features. To handle sharp features, however, MPU[9] requires fitting a number of surfaces locally, and this is a non-trivial task as it requires the identification of discontinuities, in addition, this method is also not robust for noisy point set (see Fig.6). Our MPU-APSS, combining multi-level partition of unity with APSS, is simpler and more efficient, and shows improved stability for fitting the high curvature regions, as illustrated in Fig.5 and Fig.6, the features is better reconstructed. There are 417 245\217 351 points in Fig.5 and Fig.6, respectively, it takes our method 19 s\7 s for reconstruction, while it takes 66 s\32 s using method in [9]. In Fig.5, for both our method and MPU[9], the approximation accuracy ε is set 0.0001 (0.01% of the length of the diagonal of the bounding box of the model), the minimum number of points in each cell is set 15.

Fig.7 shows the filtering result using our MPU-APSS method, as shown in Fig.7, even handle the heavily noisy point data, the filtered point set is smooth and the features are well preserved. Different from surface reconstruction (see Fig.5 and Fig.6), which is reconstructed from the implicit distance fields, the filtering result in Fig.7 is performed using the projection operator.

Fig.8 gives MPU-APSS reconstruction of a noisy data using different curvature control parameter β. By introducing parameter $\beta \in [0, 1]$ in (3), algebraic spherical can be continuously tweaked to fit from a pure planar fit and a pure spherical fit[7]. More generally, this parameter allows to control the curvature of the fitted sphere: a negative value of β inverts the curvature, a value greater than one tends to exaggerate the surface features. Note that for APSS this feature should only be used for densely sampled models since extreme settings of β will reduce the stability of the representation. With the β integrated in our MPU-APSS for surface reconstruction, however, it is even robust to process the coarsely sampled models, as shown in Fig.8. Reconstructing smooth surface from the noisy point set, with the changing parameter β, our MPU-APSS enables to preserve the surface structure while removing the noise, and to enhance the features with large β.

In Fig.9 and Fig.10, we show the surface reconstruction and surface completion results using the proposed method, and give the comparison results. Using the

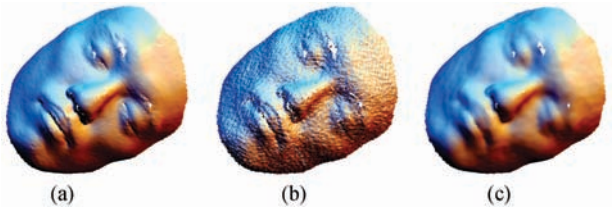

Fig.7. Noisy point set filtering. (a) Original point set (19 K points). (b) Model with noise (0.5% of the diagonal). (c) Filtered result using the almost orthogonal projection.

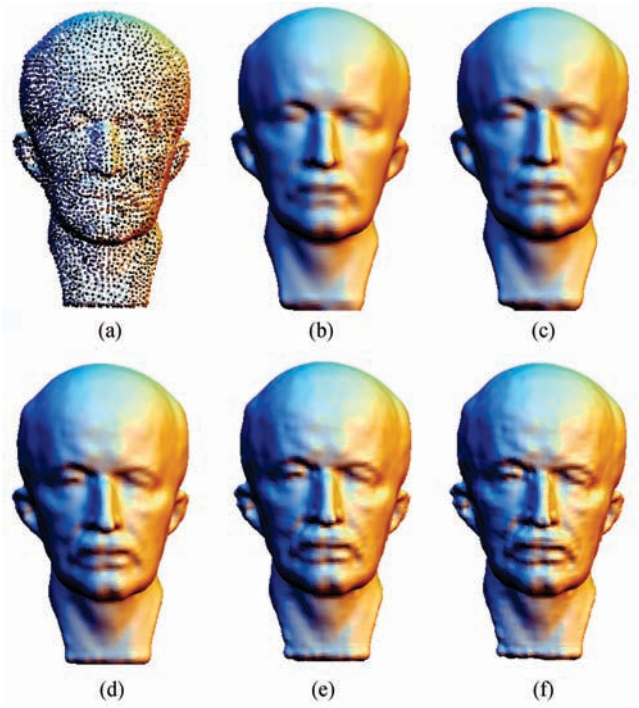

Fig.8. MPU-APSS reconstruction of a noisy data using different values of the curvature control parameter β. (a) Input model. (b) $\beta = 0.0$. (c) $\beta = 1.0$. (d) $\beta = 2.0$. (e) $\beta = 3.0$. (f) $\beta = 4.0$.

octree subdivision strategy, compared with [6, 9], our method is easy to control the approximation accuracy, as shown in Fig.9, the features are better reconstructed than [6, 9]. In Fig.10, we perform the surface reconstruction on an incomplete point set with large hole. As our MPU algebraic point set surfaces blend the local shape approximations for the point sets, we can

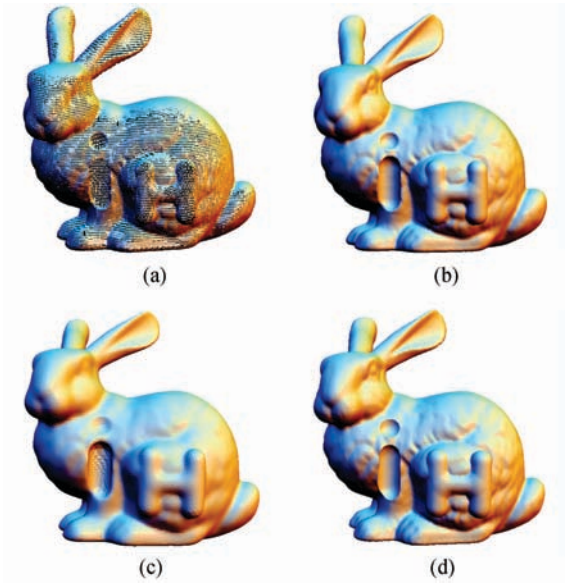

Fig.9. (a) Original bunny point set. (b) Reconstruction using APSS[6]. (c) Reconstruction using MPU[9]. (d) Reconstruction using proposed MPU-APSS.

complete the point set with large missing data. However, the APSS[6] defined locally makes this upsampling method not work well for completing point sets with large holes, as illustrated in Figs. 10(c) and 10(f). As APSS[6] computes local approximation for each point while our method selects the center of the octree leaves and blends those local approximations using a partition of unity, thus our method is faster than APSS[6]. For the Dragon\Bunny models in Fig.10 with 452 645\134 346 points, respectively, our methods takes 20 s\6 s, while APSS[6] takes 300 s\80 s.

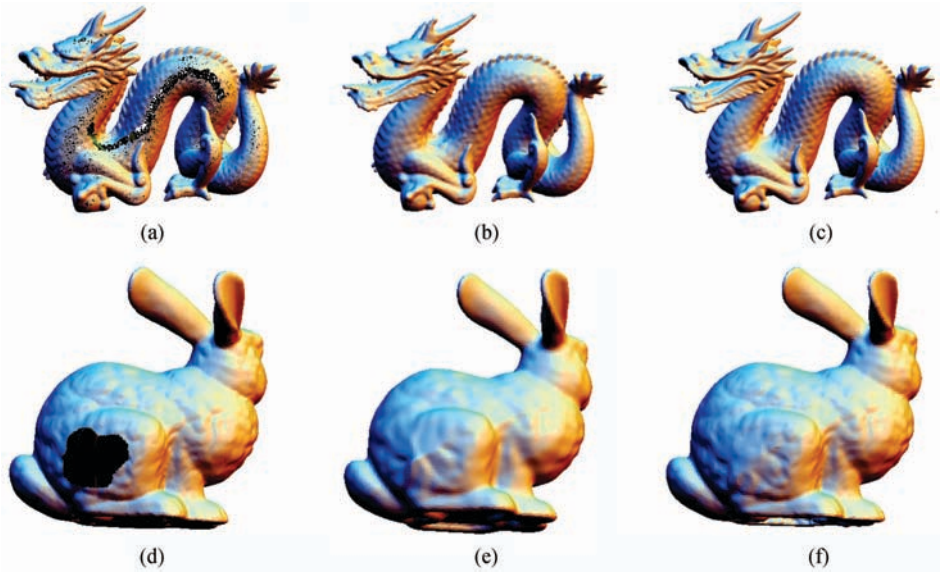

Fig.10. Surface completion. (a), (d) The incomplete point sets with holes. (b), (e) Completed surfaces with APSS[6]. (c), (f) Completed surfaces with proposed MPU-APSS.

Fig.11 gives an upsampling result, our MPU-APSS combining almost orthogonal projection is robust for upsampling the insufficient sampled point sets. Note that the results are smooth and the insufficient regions with sharp features are also well reconstructed. Guennebaud *et al.*[7] also provided a dynamic upsampling strategy to make a real-time rendering system, and they presented algorithms to implement the upsampling operation on parallel multicore architectures. In the future, we will integrate our upsampling methods into GPU acceleration.

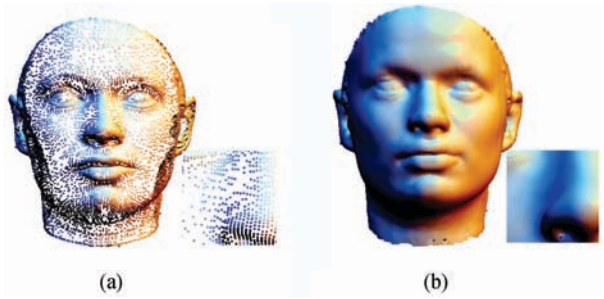

Fig.11. (a) Original point set with 24 K points. (b) Upsampled result set with 710 K points.

6 Conclusion and Future Work

We have presented an MPU algebraic point set surfaces which can be represented by either a projection or an implicit form. Our method employs the partition of unity approach to build an implicit approximation function for the scattered point set, which results in an error-controlled approximation for the point data. Compared with APSS method[6], our method has following two main advantages: one is that our method offers much higher performance in the case of smoothing of a densely sampled surface, because APSS have to consider hundreds of samples at each evaluations while the proposed MPU-APSS approach does fewer of these expensive sphere fits; the other is that with pure APSS we have to determine the weighting functions w_i while this can be automatic with the hierarchical octree subdivision schemes used in our method. In addition, the MPU-APSS preserves the desirable nature of the implicit approaches such as automatic large hole filling. We also have presented an efficient and robust projection operator to make the approach well suited to filter heavily noisy input data and point set resampling.

There are also some drawbacks in our method. Our method requires precomputation before performing surface reconstruction, and the surface reconstruction result also depends on the arbitrary choice of the nodes. Another drawback is that the partition of unity (POU) method may generate inflexions, although using multi-level partition of unity may alleviate this problem, however, this problem cannot be avoided completely. In the future, we will investigate point-based modeling using the MPU algebraic point set surfaces, and perform constructive solid geometry (CGS) operations on point sets. Another interesting future topic is to automatically detect the outliers from raw scanned data using our MPU-APSS, and reconstruct smooth surface from raw scanned data with outliers.

Acknowledgment The author would like to thank the anonymous reviewers for their valuable comments and insightful suggestions. The author also thanks Guang-Pu Feng, Zhong-Yi Du, Xun-Hua Yang for their work in implementing some of the results presented in the paper.

References

[1] Levin D. Mesh-Independent Surface Interpolation. Geometric Modeling for Scientific Visualization, Springer, 2003, pp.37-49.

[2] Alexa M, Behr J, Cohen-Or D, Fleishman S, Levin D, Silva C T. Computing and rendering point set surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 2003, 9(1): 3-15.

[3] Gross M, Pfister H. Point-Based Graphics. Morgan Kaufmann, 2007.

[4] Amenta N, Kil Y J. Defining point-set surfaces. *ACM Transactions on Graphics*, 2004, 23(3): 264-270.

[5] Alexa M, Adamson A. On normals and projection operators for surfaces defined by point sets. In *Proc. Symposium on Pint-Based Graphics*, Zurich, Switzerland, Jun. 2-4, 2004, pp.150-155.

[6] Guennebaud G, Gross M. Algebraic point set surfaces. *ACM Transactions on Graphics*, 2007, 26(3): Article No.3.

[7] Guennebaud G, Germann M, Gross M. Dynamic sampling and rendering of algebraic point set surfaces. *Computer Graphics Forum*, 2008, 27(2): 653-662.

[8] Babuska I, Melenk J M. The partition of unity method. *International Journal for Numerical Methods in Engineering*, 1997, 40(4): 727-758.

[9] Ohtake Y, Belyaev A, Alexa M, Turk G, Seidel H P. Multi-level partition of unity implicits. In *Proc. International Conference on Computer Graphics and Interactive Techniques* (*SIGGRAPH 2003*), San Diego, USA, Jul. 27-31, 2005, pp.463-470.

[10] Xiao C, Feng G, Chu Y, Du Z, Yang X. Multi-level partition of unity algebraic point set surfaces. *International Conference on Multimedia Information Networking and Security*, Wuhan, China, Nov. 18-20, 2009, pp.645-649.

[11] Pauly M, Keiser R, Kobbelt L P, Gross M. Shape modeling with point-sampled geometry. *ACM Transactions on Graphics*, 2003, 22(3): 641-650.

[12] Alexa M, Adamson A. Interpolatory point set surfaces-convexity and Hermite data. *ACM Transactions on Graphics*, 2009, 28(2): Article No. 20.

[13] Fleishman S, Cohen-Or D, Silva C T. Robust moving least-squares fitting with sharp features. In *Proc. International Conference on Computer Graphics and Interactive Techniques* (*SIGGRAPH 2005*), Los Angeles, USA, Jul. 31-Aug. 4, 2005, pp.544-552.

[14] Pratt V. Direct least-squares fitting of algebraic surfaces. *ACM SIGGRAPH Computer Graphics*, 1987, 21(4): 145-152.

[15] Hoppe H, De Rose T, Duchamp T *et al.* Piecewise smooth surface reconstruction. In *Proc. SIGGRAPH 1994*, Orlando, USA, Jul. 24-29, 1994, pp.295-302.

[16] Amenta N, Bern M, Kamvysselis M. A new Voronoi-based surface reconstruction algorithm. In *Proc. SIGGRAPH 1998*, Orlando, USA, Jul. 19-24, 1998, pp.415-421.

[17] Carr J C, Beatson R K, Cherrie J B, Mitchell T J, Fright W R, McCallum B C, Evans T R. Reconstruction and representation of 3D objects with radial basis functions. In *Proc. SIGGRAPH 2003*, Los Angeles, USA, Aug. 12-17, 2001, pp.67-76.

[18] Morse B S, Yoo T S, Rheingans P, Chen D T, Subramanian K R. Interpolating implicit surfaces from scattered surface data using compactly supported radial basis functions. In *Proc. SMI 2001 International Conference on Shape Modeling and Applications*, Genova, Italy, May 7-11, 2001, pp.89-98.

[19] Dinh H Q, Turk G, Slabaugh G. Reconstructing surfaces using anisotropic basis functions. In *Proc. the Eighth IEEE International Conference on Computer Vision* (*ICCV 2001*), Vancouver, Canada, Jul. 9-12, 2001, pp.606-613.

[20] Kazhdan M, Bolitho M, Hoppe H. Poisson surface reconstruction. In *Proc. the 4th Eurographics Symposium on Geometry Processing*, Cagliari, Italy, Jun. 26-28, 2006, pp.61-70.

[21] Xie H, McDonnell K T, Qin H. Surface reconstruction of noisy and defective data sets. In *Proc. the Conference on Visualization '2004*, Vancouver, Canada, Jul. 7-14, 2004, pp.259-266.

[22] Ohtake Y, Belyaev A, Alexa M. Sparse low-degree implicit surfaces with applications to high quality rendering, feature extraction, and smoothing. In *Proc. the 3rd Eurographics Symposium on Geometry Processing*, Vienna, Austria, Jul. 4-6, 2005, Article No. 149.

[23] Turk G. Re-tiling polygonal surfaces. *Computer Graphics Association for Computing Machinery*, 1992, 26(2): 55-64.

[24] Pauly M, Gross M, Kobbelt L P, Hochschule E T, Zurich S. Efficient simplification of point-sampled surfaces. In *Proc. VIS 2002*, Boston, USA, Oct. 27-Nov. 1, 2002, pp.163-170.

[25] Guennebaud G, Barthe L, Paulin M. Dynamic surfel set refinement for high-quality rendering. *Computers & Graphics*, 2004, 28(6): 827-838.

[26] Guennebaud G, Barthe L, Paulin M. Interpolatory refinement for real-time processing of point-based geometry. *Computer Graphics Forum*, 2005, 24(3): 657-666.

[27] Xiao C, Zheng W, Peng Q, Forrest A R. Robust morphing of point-sampled geometry. *Computer Animation and Virtual Worlds*, 2004, 15(3/4): 201-210.

[28] Xiao C, Zheng W, Miao Y, Zhao Y, Peng Q. A unified method for appearance and geometry completion of point set surfaces. *The Visual Computer*, 2007, 23(6): 433-443.

[29] Xiao C, Miao Y, Liu S, Peng Q. A dynamic balanced flow for filtering point-sampled geometry. *The Visual Computer*, 2006, 22(3): 210-219.

[30] Xiao C, Fu H, Tai C L. Hierarchical aggregation for efficient shape extraction. *The Visual Computer*, 2009, 25(3): 267-278.

[31] Miao Y W, Feng J Q, Xiao C X, Peng Q S, Forrest A R. Differentials-based segmentation and parameterization for point-sampled surfaces. *Journal of Computer Science and Technology*, 2007, 22(5): 749-760.

[32] Hoppe H, DeRose T, Duchamp T, McDonald J, Stuetzle W. Surface reconstruction from unorganised points. *Computer Graphics*, 1992, 26(2): 71-77.

[33] Franke R, Nielson G. Smooth interpolation of large sets of scattered data. *International Journal for Numerical Methods in Engineering*, 1980, 15(11): 1691-1704.

[34] Taubin G. Estimation of planar curves, surfaces, and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1991, 13(11): 1115-1138.

Chun-Xia Xiao received the Ph.D. degree from the State Key Lab of CAD & CG of Zhejiang University in 2006. He is currently an associate professor at the School of Computer, Wuhan University, China. During October 2006 to April 2007, he worked as a postdoc researcher at the Department of Computer Science and Engineering, the Hong Kong University of Science and Technology. His research interests include image and video processing, digital geometry processing and computational photography. He is a senior member of CCF, and a member of ACM.