Adaptive Surface Reconstruction Based on Implicit PHT-Splines

Jun Wang
Univ. of Sci. & Tech. of China
Hefei 230026, P. R. China
binghuo@mail.ustc.edu.cn

Zhouwang Yang*
Univ. of Sci. & Tech. of China
Hefei 230026, P. R. China
yangzw@ustc.edu.cn

Liangbing Jin
Zhejiang Normal University
Jinhua 321004, P. R. China
lbjin@zjnu.cn

Jiansong Deng
Univ. of Sci. & Tech. of China
Hefei 230026, P. R. China
dengjs@ustc.edu.cn

Falai Chen†
Univ. of Sci. & Tech. of China
Hefei 230026, P. R. China
chenfl@ustc.edu.cn

ABSTRACT

We present a new shape representation, the implicit PHT-spline, which allows us to efficiently reconstruct surface models from very large sets of points. A PHT-spline is a piecewise tricubic polynomial over a 3D hierarchical T-mesh, the basis functions of which have good properties such as nonnegativity, compact support and partition of unity. Given a point cloud, an implicit PHT-spline surface is constructed by interpolating the Hermitian information at the basis vertices of the T-mesh, and the Hermitian information is obtained by estimating the geometric quantities on the underlying surface of the point cloud. We use the natural hierarchical structure of PHT-splines to reconstruct surfaces adaptively, with simple error-guided local refinements that adapt to the regional geometric details of the target object. Unlike some previous methods that heavily depend on the normal information of the point cloud, our approach only uses it for orientation and is insensitive to the noise of normals. Examples show that our approach can produce high quality reconstruction surfaces very efficiently.

Categories and Subject Descriptors

I.3.5 [**Computer Graphics**]: Computational Geometry and Object Modeling—*boundary representations, curve, surface, solid, and object representations, splines*

Keywords

PHT-spline, T-mesh, surface reconstruction, Hermite interpolation

*Corresponding author. Email:yangzw@ustc.edu.cn
†Corresponding author. Email:chenfl@ustc.edu.cn

2010 ACM Symposium of Solid and Physical Modeling (SPM '10), Haifa, Israel

1. INTRODUCTION

Point-sampled geometry has received an increasing interest in the past decade, and a lot of research has been devoted to its efficient representation, modeling, processing and rendering. There are two main reasons for this: the needs of industry and the convenience of acquisition. On one hand, with the development of modern industry, high quality surfaces and aesthetics of CAM products are being required in many industries, such as jewelry and automobile industries, where free-form surfaces are usually used. In order to design these surfaces, some physical media such as clay models are first designed for data scanning. Meanwhile, some existing models and products with complicated surfaces need to be reproduced. As a first step in converting a prototype into a computer model for subsequent CAD/CAM processing, point clouds are generated with the coordinates of each point captured from the surfaces of existing objects. On the other hand, modern three-dimensional (3D) digital photography systems and 3D range scanning devices acquire geometry of complex and real-world objects conveniently. These techniques generate huge volumes of point samples, which constitute the discrete building blocks of 3D object geometry.

1.1 Related work

Converting a point-sampled representation of an object into a more compact one, such as a triangle mesh, a collection of parametric patches, or the zero level set of a function, is known as surface reconstruction. The main difficulties of surface reconstruction in practice come from the potentially complicated topology, regional geometric details, huge numbers, non-uniformity and noise of unorganized points. The problem of surface reconstruction from point clouds has received much attention in the computational geometry and computer graphics communities, and various methods have been developed to solve the problem that depend on the properties of the input, the desired output, the philosophy of the user, and so on. A recent survey of the methods is available in the literature of Point-based Graphics [16], which are mainly classified into Voronoi-based methods, surface evolution methods, and implicit function methods.

Algorithms for surface reconstruction developed in the computational geometry community are based on the Voronoi

diagram [2] that decomposes a 3D space into convex polyhedra. Dual of Voronoi diagram, the Delaunay triangulation, establishes topological connections between sample points and then a subset of the resulting simplices is filtered out to be the reconstructed mesh. These schemes, such as Delaunay triangulations [6], alpha-shapes [14], power crust [3], come with theoretical guarantees of correct reconstruction if the sample meets certain conditions (dense and noiseless). Some recent work [8, 12, 18] addressed the issue on noisy and non-uniform samples.

The level-set method was applied to surface reconstruction in [42], where the problem is formulated in terms of a partial differential equation(PDE) describing the surface evolution. The method is computationally expensive due to solving a nonlinear PDE. In [39, 40], evolution of T-spline level sets with distance field constraints is developed to reconstruct a base surface from unorganized data points. From the extracted mesh of the base surface, an additional evolution process which combines a data-driven velocity and a bilateral filtering, is employed to reproduce detailed features of the target shape. Recently, Gauss-Newton type method for fitting implicitly defined curves and surfaces to given data is presented in [1], which can also be viewed as the discrete iterative version of a time-dependent evolution process. Generally, evolution methods are computationally expensive and the convergence of the evolution is not guaranteed.

Implicit surfaces are very popular in surface reconstruction. In these methods, the reconstructed surface is defined as the zero level set of a function that is designed to be negative inside and positive outside of the object. The signed distance function to the underlying surface of a given point set is one possible candidate for the implicit function. In [17] a signed distance function is estimated as the distance to the tangent plane of the closest point. Curless and Levoy [9] blended the directional distance associated with each range scan, using Gaussian weights to form the implicit function. In [19], an approach for fitting implicitly defined algebraic spline surfaces to scattered data is presented, which simultaneously approximate points and associated normal vectors. This approach is generally not very efficient since a large system of linear equations has to be solved. An improvement of this method is to adaptively select knots such that the reconstruction surface fits the point cloud adaptively [34]. In [37, 7, 13, 38], an implicit function is constructed with polyharmonic radial basis functions (RBFs) [31], by placing zero constraints at each input point and also a pair of non-zero constraints at "offset-surface" points. However, the ideal RBFs are globally supported and non-decaying, so the matrix of the linear system is dense and ill-conditioned. Morse et al. [27], Kojekine et al. [22] and Ohtake et al. [30] employed compactly supported RBFs to reduce the computation and speed up the reconstruction process. In [29], Ohtake et. al. proposed the multi-level partition of unity method called MPU to reconstruct samples with huge number of data points. The idea is to break the data domain into sub-domains, fit a local shape function to the data in each sub-domain separately and then blend the local shape functions with auxiliary weights. MPU is quite fast, but results in a reconstruction with complicate expressions and possibly generates extra zero set for noisy data. An improvement is to apply a laplacian smoothing for the gradient vector fields of the reconstruction surface to obtain a noise robust surface reconstruction [28].

Besides signed distance functions, indicator functions are often used in surface reconstruction. Points equipped with oriented normals can be viewed as samples of the gradient of an indicator function [20, 21, 25]. In [20], Fourier series are used to represent indicator functions. However, computing Fourier coefficients requires a summation over all the samples because of the globally supported basis functions, and the method needs also a huge amount of memory due to the use of uniform grid. The Fourier series approach was then improved in [21], where computing the desired indicator function leads to a Poisson problem. The Poisson problem admits a hierarchy of locally supported functions, and therefore the problem is reduced to solving a sparse linear system. Poisson reconstruction can create smooth surfaces that robustly approximate noisy data. However, it also has to solve a large linear system and is sensitive to the noise of normals. In [25], wavelet bases are used to represent the indicator function and a streaming surface reconstruction using wavelets is proposed. Due to the multiresolution nature of wavelets, the wavelet method can reconstruct surfaces very efficiently and process extremely large data set. The smoothness and the quality of the reconstructed surface depend on the selected basis functions.

1.2 Contributions

In the current paper, we propose an adaptive surface reconstruction algorithm based on implicit PHT-splines. Similar to MPU [29], our implicit PHT-spline representation could be viewed as an adaptive signed distance fields [15] with the difference that our representation is globally C^1 continuous and of Hermite interpolation. We approximate the target geometry of a point cloud with an implicit surface of polynomial splines over 3D hierarchical T-mesh, which is constructed adaptively by error-guided local refinements. In each progressive level, the PHT-spline function is determined by interpolating the Hermitian information at the basis vertices of the hierarchical T-mesh, and the Hermitian information at the basis vertices is obtained from the geometric quantities on the underlying surface of the point cloud. Our approach has following major advantages:

- ◇ A new shape representation, the implicit PHT-spline surface, is presented. The PHT-spline provides a unified representation in piecewise polynomials that automatically maintains a global C^1 continuity, and which is more convenient for subsequent processing than other methods.
- ◇ A PHT-spline function in each cell is a tricubic polynomial which has strong capability to capture geometric details. The architecture of PHT-splines holds a natural hierarchical structure that is simple for local refinements and particularly suitable for adaptive description of the target geometry.
- ◇ Our algorithm adaptively produces a hierarchical T-mesh, in which the number of basis vertices is roughly one-third of the number of cells. In the reconstruction process, we only have to estimate the Hermitian information at the basis vertices instead of fitting local shape to data points in each cell. Thus our approach is very efficient both in spatial and temporal cost.
- ◇ Our algorithm uses the normal information only for orientation, i.e., inside-outside orientation of the sur-

face. Thus it has less dependency on the normal information and is robust in the presence of noisy normals.

The paper is organized as follows. In section 2.1, a novel shape representation called implicit PHT-spline surface is introduced to describe the target geometry. Then in section 2.2, an optimization model is proposed to compute geometric quantities on the underlying surface of a point cloud. In section 3, we present the main approach on the adaptive surface reconstruction algorithm based on implicit PHT-splines. In section 4, we give implementation and comparison details and demonstrate our approach through various examples to illustrate the effectiveness of the proposed method. Finally, we conclude the paper with some future research problems.

2. PRELIMINARIES

Implicit shape representations are attractive because they provide the capability to describe objects of complex topology while avoiding a problematic parametrization of the target geometry, and many geometric operations such as intersection and union, are easy to perform on such models [5]. It is being increasingly recognized that a set of modeling and animation techniques based on implicit representation exhibit much more advantages over other representations.

One preferred choice of implicit representations is algebraic spline surfaces which are defined as the zero level set of a tensor-product spline [19]. Such representation offers several advantages, such as a compact and analytic expression, global smoothness, efficient evaluation and sufficient flexibility for subsequent processing. However, algebraic tensor-product B-spline surfaces suffer from the weakness of superfluous control coefficients and the difficulties in local refinement and adaptivity. To eliminate superfluous control points in tensor-product B-spline (or NURBS), Sederberg et.al. [33, 32] invented T-spline, a rational spline defined over a T-mesh which supports many valuable operations within a consistent framework. Recently, Deng et al. introduced polynomial splines over hierarchical T-meshes (called PHT-splines) which have more ability in local shape control [10, 11]. PHT-splines are then applied in surface fitting [11, 23] and signed distance field approximation of closed parametric curves [35]. In this paper, we generalize the notion of PHT-splines to 3D space and apply its implicit form in surface reconstruction.

2.1 Polynomial splines over 3D hierarchical T-meshes

Given a rectangular domain in 2D space, a T-mesh is a rectangular partition of the domain, where T-junctions are allowed. As a natural extension of 2D T-mesh, a 3D T-mesh is basically a partition of a cube $\Omega \subset \mathbb{R}^3$, where the partition planes are parallel to the faces of the cube. Instead of considering general T-meshes, we restrict our attention to hierarchical T-meshes which have a nested structure. A hierarchical T-mesh starts with a tensor-product mesh $\mathscr{T}_0$. Denote the T-mesh at level k by $\mathscr{T}_k$. For any $k \geqslant 0$, some selected cells of level k are subdivided equally by three planes parallelling to xoy, yoz and zox, into eight sub-cubes which are labeled as the cells of level $k+1$. Figure 1 shows an example of 2D hierarchical T-mesh.

Given a 3D hierarchical T-mesh $\mathscr{T}$, denote Φ the set of all the cells in $\mathscr{T}$. We define a tricubic polynomial spline

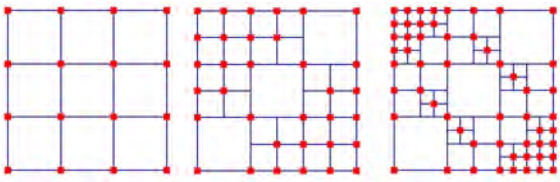

Figure 1: A 2D hierarchical T-mesh (From left to right are level 0, level 1 and level 2, the red squares denote the basis vertices).

space over $\mathscr{T}$ as

$$\mathbb{S}(3,3,3,1,1,1,\mathscr{T}) := \left\{ s(x,y,z) \in C^{1,1,1}(\Omega) \middle| s(x,y,z)|_\phi \in \mathbb{P}_{3,3,3}, \forall \phi \in \Phi \right\}, \quad (1)$$

where $\mathbb{P}_{3,3,3}$ is the space of all polynomials in three variables with tri-degree $(3,3,3)$, and $C^{1,1,1}(\Omega)$ is the space consisting of trivariate functions that are C^1 continuous along x, y, z directions in Ω, respectively. It is easy to see that $\mathbb{S}(3,3,3,1,1,1,\mathscr{T})$ is a linear space whose dimension [10] is

$$\dim \mathbb{S}(3,3,3,1,1,1,\mathscr{T}) = 8(V^b + V^+), \quad (2)$$

where V^b and V^+ represent the number of boundary vertices and interior crossing vertices in the T-mesh $\mathscr{T}$, respectively.

The dimension formula in Eq. (2) gives us a hint on how to construct basis functions of the spline space: each boundary vertex or interior crossing vertex should associate with eight basis functions. The boundary vertices and the interior crossing vertices are called *basis vertices* of the T-mesh. The strategy for constructing basis functions presented in [11] for polynomial splines over 2D hierarchical T-meshes can be directly extended to construct the the basis functions of the spline space $\mathbb{S}(3,3,3,1,1,1,\mathscr{T})$. For each basis vertex, eight bases are constructed, and all the basis functions hold good properties, such as nonnegativity, local support and partition of unity. Figure 2 shows a basis function of 2D PHT-spline associated with a basis vertex. Let $\{\mathbf{x}_i\}_{i=1}^V$ be the basis vertices of a 3D hierarchical T-

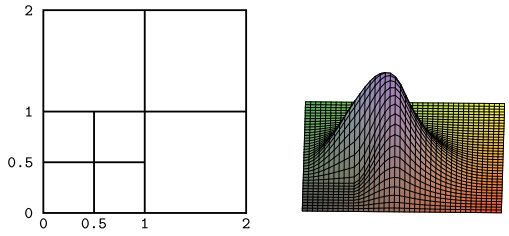

Figure 2: A basis function of 2D PHT-spline associated with the vertex $(1,1)$.

mesh $\mathscr{T}$, and $\{b^{ij}(x,y,z)\}_{i=1,j=1}^{V,8}$ be the basis functions of $\mathbb{S}(3,3,3,1,1,1,\mathscr{T})$. Then a polynomial spline over 3D hierarchical T-mesh $\mathscr{T}$ (called a 3D PHT-spline) is defined by

$$f(x,y,z) = \sum_{i=1}^{V} \sum_{j=1}^{8} C_{ij} b^{ij}(x,y,z), \quad (x,y,z) \in \Omega, \quad (3)$$

where $\{C_{ij}\}$ are control coefficients. To efficiently manipulate and evaluate a PHT-spline, we introduce a *Hermitian*

information operator $\mathcal{H}$:

$$\mathcal{H}f(x,y,z) = (f, f_x, f_y, f_z, f_{xy}, f_{yz}, f_{zx}, f_{xyz}). \quad (4)$$

At a fixed basis vertex (x_0, y_0, z_0), $\mathcal{H}b(x_0, y_0, z_0) = \mathbf{0}$ holds for all the basis functions $b(x, y, z)$ except the eight basis functions associated with the basis vertex (x_0, y_0, z_0). Since the operator $\mathcal{H}$ is linear, for any fixed basis vertex $\mathbf{x}_{i_0}$, we have

$$\begin{aligned} \mathcal{H}f(\mathbf{x}_{i_0}) &= \mathcal{H}\sum_{i=1}^{V}\sum_{j=1}^{8} C_{ij}b^{ij}(\mathbf{x}_{i_0}) \\ &= \sum_{i=1}^{V}\sum_{j=1}^{8} C_{ij}\mathcal{H}b^{ij}(\mathbf{x}_{i_0}) = \sum_{j=1}^{8} C_{i_0 j}\mathcal{H}b^{i_0 j}(\mathbf{x}_{i_0}) \\ &= \mathbf{C}_{i_0}\mathbf{B}, \end{aligned} \quad (5)$$

where $\mathbf{C}_{i_0} = (C_{i_0 1}, \cdots, C_{i_0 8})$ is a 1×8 control coefficient vector, $\mathbf{B} = (\mathcal{H}b^{i_0 1}(\mathbf{x}_{i_0})^T, \cdots, \mathcal{H}b^{i_0 8}(\mathbf{x}_{i_0})^T)^T$ is a 8×8 matrix, and $\mathcal{H}f(\mathbf{x}_{i_0})$ is the *Hermitian information vector* of $f(x, y, z)$ at the basis vertex $\mathbf{x}_{i_0}$. Since the matrix $\mathbf{B}$ is invertible and we get

$$\mathbf{C}_{i_0} = \mathcal{H}f(\mathbf{x}_{i_0})\mathbf{B}^{-1}, \quad (6)$$

which reflects the relationship between the control coefficients of a PHT-spline function and its Hermitian information at the basis vertices. Thus once we know the Hermitian information at the basis vertices, the PHT-spline representation of the reconstruction surface can be recovered.

2.2 Geometric quantities on the underlying surface

A point cloud is assumed to imply an underlying shape. In this subsection, we present an optimization model to compute the geometric quantities on the underlying surface, which will be employed in estimating the Hermitian information at the basis vertices of T-mesh to construct an implicit PHT-spline surface.

It is well known that the geometric (differential) quantities provide convenient bases for characterizing the local behavior of a shape in the vicinity of a particular point. By using the algorithm of moving parabolic approximation (MPA) [41], we can compute the location of the underlying surface of unorganized points and simultaneously estimate the differential quantities of the surface. The main observation of MPA is to locally approximate a given point cloud by an osculating paraboloid, and then recover the differential properties of its underlying surface. Suppose that $\mathcal{S}$ is the underlying surface of a point cloud $\mathcal{P} = \{\mathbf{p}_j\}_{j=1}^N$. Let $\mathbf{r}$ be a reference point in the close neighborhood of the given point cloud, and let the foot-point of $\mathbf{r}$ on the underlying surface be denoted as

$$\mathbf{o} = \mathbf{r} + \xi\mathbf{n}, \quad (7)$$

where $\mathbf{n}$ is the unit normal to $\mathcal{S}$, and ξ is the signed distance from $\mathbf{r}$ to $\mathbf{o}$ along $\mathbf{n}$. We aim to compute the foot-point $\mathbf{o}$ and the differential quantities of the underlying surface at the foot-point. Let $\{\mathbf{u}(\mathbf{n}), \mathbf{v}(\mathbf{n})\}$ be the perpendicular unit basis vectors of the tangent plane, so that $\{\mathbf{o}; \mathbf{u}, \mathbf{v}, \mathbf{n}\}$ forms a local orthogonal coordinate system. Writing $\mathbf{q}_j = \mathbf{p}_j - \mathbf{r}$, we formulate the MPA model as a constrained optimization:

$$\begin{aligned} \min \quad & Q(\mathbf{n}, \xi, a, b, c) = \sum_j \big[\mathbf{q}_j^T\mathbf{n} - \xi - \tfrac{1}{2}\big(a(\mathbf{q}_j^T\mathbf{u})^2 \\ & +2b(\mathbf{q}_j^T\mathbf{u})(\mathbf{q}_j^T\mathbf{v}) + c(\mathbf{q}_j^T\mathbf{v})^2\big)\big]^2 e^{-\frac{\|\mathbf{q}_j - \xi\mathbf{n}\|^2}{\rho^2}} \\ \text{s.t.} \quad & \mathbf{n}^T\mathbf{n} - 1 = 0, \end{aligned} \quad (8)$$

where $(\mathbf{n}, \xi, a, b, c)$ are decision variables and ρ is a scale parameter. Figure 3 illustrates the MPA model for curve case and shows its output and the geometric quantities on the underlying curve.

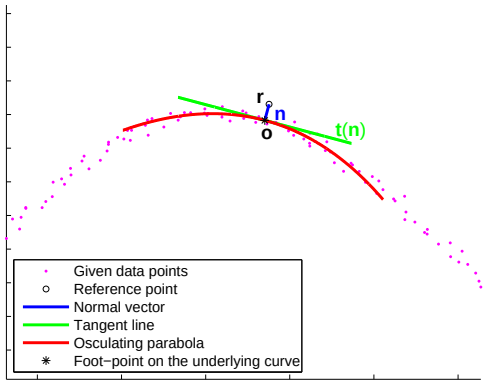

Figure 3: The MPA model for a 2D curve.

To solve the optimization problem in Eq. (8), we start with an initial guess provided in [41], and generally the solution can be obtained within three or four steps of iteration. From the optimal solution $(\mathbf{n}^*, \xi^*, a^*, b^*, c^*)$ of the MPA model, we immediately get the geometric quantities on the underlying surface, such as the foot-point, the normal vector and the curvature properties. Then we can deduce the Hermitian information at the basis vertices from these geometric quantities, and thus an implicit PHT-spline is determined.

3. APPROACH

Given a set of points $\mathcal{P} = \{\mathbf{p}_1, \cdots, \mathbf{p}_N\}$ in $\mathbb{R}^3$, our goal is to generate a 3D PHT-spline function $f(x, y, z)$ whose zero level set gives a good approximation to the underlying surface $\mathcal{S}$. Intuitively, we wish the PHT-spline function to approximate the signed distance field as accurate as possible in the vicinity of $\mathcal{S}$, while the approximation can be rough far away from $\mathcal{S}$. Our scheme is to recursively construct a hierarchical T-mesh with simple and error-guided local refinements that adapt to the target geometric details, and to determine the PHT-spline by estimating the Hermitian information at basis vertices. The reconstruction benefits from the properties of PHT-splines, such as the hierarchical structure for adaptivity, the basis functions with nonnegativity, local support and partition of unity.

3.1 Adaptive implicit PHT-spline surface approximation

The algorithm of constructing an implicit PHT-spline surface approximation is driven by the construction of a hierarchical T-mesh. At level 0, we start with a tensor-product mesh $\mathscr{T}_0$ that encompasses the given point cloud. Suppose that the number of partitions in three directions along the coordinate axes is m_x, m_y, m_z respectively, and $m_0 = \max(m_x, m_y, m_z)$ can be used to control the size of cells in the initial T-mesh. A small m_0 gives a rough approximation while a larger m_0 yields a better initial approximation. In the initial T-mesh $\mathscr{T}_0$, all vertices are the basis vertices.

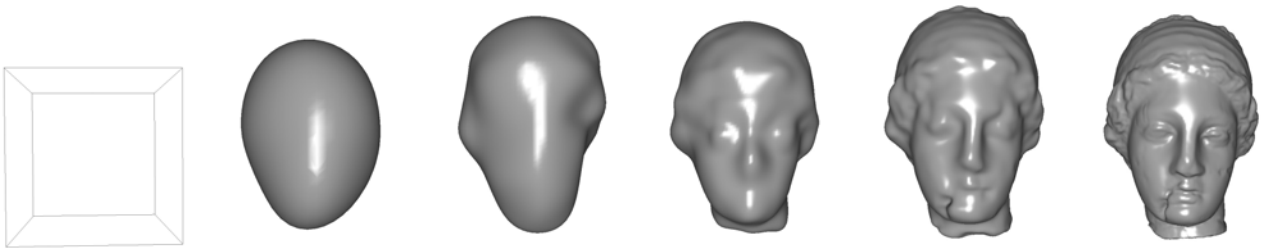

Figure 5: Adaptive reconstruction of Igea model, from left to right are the intermediate results at level 0 to level 5. The leftmost is the T-mesh at level 0 with which there is no surface generated.

Figure 4: Left: a set of points with normals from letter 'R'; Middle: the signed distance function and the zero level set; Right: the hierarchical T-mesh and the reconstructed implicit PHT-spline curve.

By calculating the Hermitian information at all the basis vertices, an initial PHT-spline function $f^{[0]}(x, y, z)$ is determined. From the level $k = 0$, the procedure repeats the following two steps until no cell needs to be subdivided or the level counter reaches a preset value:

1) Subdivide the cells of level k in which the approximation errors are larger than some given threshold. The refinement criterion is specified in Eq. (9). Label the new subcells as the cells of level $k + 1$, and form a hierarchical T-mesh $\mathscr{T}_{k+1}$ at level $k + 1$.

2) Find out all the new basis vertices in $\mathscr{T}_{k+1}$. Calculate the Hermitian information at each new basis vertex according to Eq. (10) or Eq. (12). Then the PHT-spline function $f^{[k+1]}(x, y, z)$ at level $k+1$ is constructed from Eq. (6). Set $k := k + 1$.

For the first step, the refinement criterion for the subdivision of cells is constructed as follows. Let ϕ be a cell at level k, and $\mathbf{c}$ be the center and $\Delta^{(k)}$ be the size of the cell. If the number of points contained in the cell is less than $N_{\min}$ ($N_{\min} = 6$ in our implementation), the cell will not be subdivided. Otherwise, a local approximation error is estimated according to the Sampson distance [36] and is compared with a user-specified threshold value ε_0, i.e.,

$$\max_{\|\mathbf{p}_i - \mathbf{c}\| < \frac{\sqrt{3}}{2}\Delta^{(k)}} \frac{|f^{[k]}(\mathbf{p}_i)|}{\|\nabla f^{[k]}(\mathbf{p}_i)\|} > \varepsilon_0. \tag{9}$$

If Eq. (9) holds, the cell is subdivided; Otherwise, it is not subdivided. Figure 4 illustrates a 2D curve reconstruction example, where the resulting hierarchical T-mesh is provided to demonstrate how the error-guided adaptive reconstruction works. Figure 5 shows a sequence of implicit PHT-spline surfaces that are reconstructed adaptively.

3.2 Hermitian information estimation

For the second step of the adaptive implicit PHT-spline surface approximation, we are required to obtain the Hermitian information at the newly generated basis vertices of the hierarchical T-mesh. Our strategy to estimate the Hermitian information will be in accordance with the distance from the basis vertex to the underlying surface. If the basis vertex is far away from the underlying surface, then the Hermitian information is approximated by using planar fitting technique; otherwise, Moving Parabolic Approximation(MPA) algorithm is applied. The two techniques respectively provide a first order and a second order approximation of the signed distance function.

Given a preset distance δ which is usually set to be 2% of the diagonal length of the domain Ω in our implementation, for a basis vertex $\mathbf{x}_i$ in the hierarchical T-mesh, we first compute its nearest point $\mathbf{r}_i \in \mathcal{P}$ in the point cloud, and let $\mathbf{d}_i = \mathbf{x}_i - \mathbf{r}_i$. If $\|\mathbf{d}_i\| > \delta$, the basis vertex $\mathbf{x}_i$ is considered to be far away from the surface. Otherwise, $\mathbf{x}_i$ is assumed to be close to the surface. At a far-away basis vertex $\mathbf{x}_i$, we denote $\mathbf{h} = (h, h_x, h_y, h_z, h_{xy}, h_{yz}, h_{zx}, h_{xyz})$ be the approximate Hermitian information vector. Because of little effect on the reconstruction surface, the mixed derivatives in the Hermitian information vector is set to zero, i.e.,

$$\begin{cases} h = \text{sign}(\mathbf{x}_i)\|\mathbf{d}_i\|, \\ (h_x, h_y, h_z) = \text{sign}(\mathbf{x}_i)\mathbf{d}_i/\|\mathbf{d}_i\|, \\ h_{xy} = h_{yz} = h_{zx} = h_{xyz} = 0. \end{cases} \tag{10}$$

In order to check if the vertex $\mathbf{x}_i$ is inside ($\text{sign}(\mathbf{x}_i) < 0$) or outside ($\text{sign}(\mathbf{x}_i) > 0$) of the surface, we choose the N_{far} (usually set $N_{\text{far}} = 7$ in our implementation) nearest points in $\mathcal{P}$ of $\mathbf{r}_i$ and fit them to a plane using principal component analysis(PCA). The plane normal is used to orient the Hermitian information vector.

The Hermitian information at a near-by basis vertex $\mathbf{x}_i$ is required to be accurately estimated. So we deduce theoretically the Hermitian information from the geometric quantities on the underlying surface. We search the N_{nb} (usually set $N_{\text{nb}} = 20$ in our implementation) nearest points in $\mathcal{P}$ of $\mathbf{r}_i$ and employ the MPA algorithm to compute the differential quantities of the surface. As shown in Fig. 6, the output of the MPA consists of the foot-point $\mathbf{o}_i$ of $\mathbf{r}_i$, a local right-handed coordinate system $\{\mathbf{o}_i; \mathbf{u}(\mathbf{n}), \mathbf{v}(\mathbf{n}), \mathbf{n}\}$, and an osculating paraboloid with shape parameters (a, b, c). Within the local coordinates $(s, t, w) = (\mathbf{x} - \mathbf{o}_i)^T(\mathbf{u}, \mathbf{v}, \mathbf{n})$, the oscu-

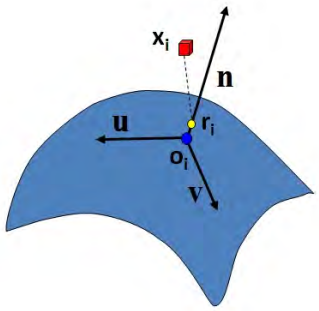

Figure 6: Hermitian information estimation by using the MPA algorithm.

lating paraboloid of the underlying surface can be expressed as

$$w = \psi(s,t) = \frac{1}{2}(as^2 + 2bst + ct^2). \tag{11}$$

Thus, we obtain the Hermitian information at the near-by basis vertex $\mathbf{x}_i$ as follows:

$$\begin{cases} h = w_i - \frac{1}{2}(as_i^2 + 2bs_it_i + ct_i^2), \\ h_x = n_1 - (as_i + bt_i)u_1 - (bs_i + ct_i)v_1, \\ h_y = n_2 - (as_i + bt_i)u_2 - (bs_i + ct_i)v_2, \\ h_z = n_3 - (as_i + bt_i)u_3 - (bs_i + ct_i)v_3, \\ h_{xy} = -(au_2 + bv_2)u_1 - (bu_2 + cv_2)v_1, \\ h_{yz} = -(au_3 + bv_3)u_2 - (bu_3 + cv_3)v_2, \\ h_{xy} = -(au_1 + bv_1)u_3 - (bu_1 + cv_1)v_3, \\ h_{xyz} = 0, \end{cases} \tag{12}$$

where $(s_i, t_i, w_i) = (\mathbf{x}_i - \mathbf{o}_i)^T(\mathbf{u}, \mathbf{v}, \mathbf{n})$. The normal $\mathbf{n}$ from the MPA model could be opposite to the orientation of the surface. To address this issue, a reorientation strategy is needed here. Compute the inner-product (denoted by σ) between $\mathbf{n}$ and the average of normal vectors around $\mathbf{r}_i$, if σ is negative then we inverse the Hermitian information vector. After the Hermitian information is estimated at the basis vertices, the implicit PHT-spline surface is constructed according to Eq. (6).

4. IMPLEMENTATION AND RESULTS

The implicit PHT-spline surface reconstruction algorithm has been implemented on a PC with an Intel Core2 Dual @2.8GHz processor and 4.0Gb of memory. We will now present the implementation details and test its performance on a set of typical examples. Comparisons with other methods and discussions are also made in this section.

4.1 Implementation details

Given a set of unorganized points, a kd-tree is built using the ANN [26] library. A hierarchical T-mesh is initialized with a user-specified tensor-product grid (level 0) which is the cube Ω in most examples. Similar to an octree implementation, the hierarchical T-mesh data structure maintains the cells' neighbor information, parent-children relationship and cell-vertices relationship.

A PHT-spline function is a tricubic polynomial in each cell of the T-mesh and it can be represented by the Bézier form, which is more convenient and efficient for operations such as subdivision and evaluation. By using the de Casteljau algorithm, evaluating the reconstructed function takes about 250 multiplications. In our implementation, a matrix storing the Bézier ordinates is maintained for each cell. When a cell ϕ at level k is subdivided into eight subcells of level $k+1$, we get eight new matrices corresponding to the eight subcells at level $k+1$. Then we use the Hermitian information at the newly generated basis vertices in $\mathscr{T}_{k+1}$ to update the Bézier ordinates around these new basis vertices.

There are two typical choices for isosurface extraction from an implicit function: Marching Cube method [24] and Bloomenthal's method [4]. In our implementation, we employ the Bloomenthal's polygonizer to extract the isosurface with the same strategy used in MPU method [29].

4.2 Results and discussions

When the target shapes have rich geometric details, it requires that the reconstruction methods are fully self-adaptive and have sufficient capacity for describing the details. Global methods become inefficient for such problems due to large-scale data sets. For example, the RBF method is global and has to solve a large linear system, consuming more time and memory [7, 30]. On the contrary, the local method is generally more efficient, but it must take into account the overall smoothness. Usually some strategies should be employed to achieve an overall smooth reconstruction. MPU is such an example where the local shape functions are blended to form an overall reconstruction with a complicated expression. With the help of PHT-splines, our approach has achieved a balance between global smoothness and efficiency, and overcomes some of the deficiencies by previous methods. To evaluate our approach, we have tested it on a variety of models with different features and made comparisons with MPU [29] and Poisson [21] reconstruction methods. The performance statistics are summarized in Table 1.

Global continuity and geometric description power Owing to the good properties of basis functions, the PHT-splines provide a unified representation in piecewise polynomials that automatically maintains a global C^1 continuity. On the other hand, a PHT-spline function in each cell is a tri-cubic polynomial which has strong capability to capture the geometric details. The construction of PHT-splines is a dynamic process and is particularly suitable for adaptive description of the target geometry (see Fig. 5). Our algorithm reconstructs high-quality surfaces with very fine details, as shown in Fig. 7. Compared with MPU method, our approach needs less number of spatial subdivision (with about one fourth of the cells) as shown in Table 2, to achieve the same approximation accuracy. Here the approximation accuracy is measured by the maximum distance of the point cloud to the implicit surface. This again confirms that our approach has a better ability to describe the geometry. Furthermore, our method results in a piecewise polynomial while the MPU method has more complicated expression.

Memory costs In our algorithm, the memory consumption is mainly used for storing the Bézier ordinates for all the cells in the hierarchical T-mesh. Because the representation of implicit PHT-splines has excellent self-adaptability and strong geometric description capability, the algorithm produces a hierarchical T-mesh not containing superfluous cells. Table 2 shows the number of cells, vertices and basis vertices of our hierarchical T-mesh and the number of cells of MPU method for several examples (see Fig. 7-8) under the same

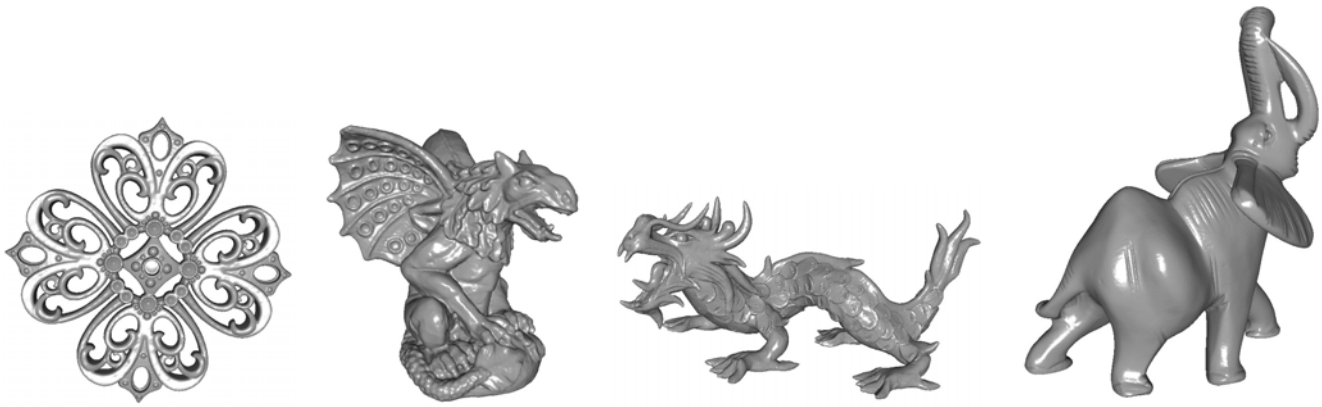

Figure 7: Reconstruction of different models(Filigree model with 514,300 points, Gargoyle model with 863,210 points, Asian dragon model with 3,609,455 points and Elephant model with 1,512,290 points.

Table 1: The performance statistics

			IPHT			MPU			Poisson		
Model	#Points	Rel. error	Time	Mem.	#Tris.	Time	Mem.	#Tris.	Time	Mem.	#Tris.
Asiandragon	3,609,455	2.0e-4	41.342s	738M	2,470,784	199.707s	850M	2,449,768	706.0s	1,058M	7,422,418
Neptune	2,003,932	1.0e-4	25.686s	494M	1,733,460	142.688s	736M	1,734,896	306.8s	505M	3,211,190
Elephant	1,512,290	3.0e-4	22.713s	390M	2,056,288	89.945s	492M	2,032,128	229.6s	537M	2,577,824
Armadillo	172,974	5.0e-4	3.544s	88M	238,592	17.644s	229M	240,012	16.2s	163M	204,302
NoisyArmadillo	172,974	5.0e-4	3.544s	88M	238,592	64.148s	240M	240,204	17.2s	163M	189,436
Gargoyle	863,210	5.0e-4	10.583s	239M	853,136	53.313s	320M	645,420	112.3s	370M	1,332,000
Filigree	514,300	5.0e-4	6.861s	168M	505,320	34.129s	293M	501,136	61.4s	238M	668,760
Bunny	362,272	8.0e-4	10.219s	212M	347,712	40.513s	280M	320,932	84.8s	243M	783,112

Table 2: Our hierarchical T-mesh vs. MPU's octree

Model	#cells	#vts.	#b-vts.	#cells(MPU)
Asiandragon	593,121	1,018,078	183,105	2,421,025
Neptune	582,785	1,063,984	162,441	3,292,233
Elephant	219,929	374,898	68,404	896,809
Gargoyle	251,649	453,453	72,349	1,329,209
Filigree	233,201	400,605	73,601	700,625
Armadillo	113,089	170,545	44,940	512,145

error. Since the number of cells by our approach is much less than that of the MPU method (about one-fourth), our approach has a slight advantage in total memory requirements (see Table 1).

Computational time and efficiency As mentioned above, our algorithm adaptively produces a hierarchical T-mesh, in which the number of basis vertices is roughly one-third of the number of cells. In the reconstruction process, we only need to estimate the Hermitian information at basis vertices rather than to fit local shape to data points in each cell. As described in Section 3.2, the Hermitian information at a basis vertex is obtained by using moving least squares plane fitting or MPA algorithm. Therefore, the time cost of our algorithm is equal to the average computational time of calling the MPA algorithm multiplied by the number of basis vertices in the generated hierarchical T-mesh. We summary the performance of our approach, MPU method and Poisson reconstruction method on a variety of data models in Table 1, including the spatial and computational cost (The timing in the table includes the contouring time). By statistics, our algorithm is several times faster than MPU method, and is much more efficient than Poisson method.

Reconstruction from data with unreliable normals In the data acquisition of an object, the surface normal information is often more difficult to accurately obtain and contains much noise. Most of existing surface reconstruction methods heavily depend on the normal information of the given data. In our algorithm we only use the normal information for orientation, i.e., inside-outside determination. It is conceivable that our algorithm is robust in the presence of certain noise of normals. Figure 9 shows the reconstruction surfaces of Armadillo model with noisy normals by different methods. For the MPU reconstruction, more time is spent and some extra sheets are generated near the ears, hands and feet due to the noise of normals. For the Poisson reconstruction, the noise makes some parts thinner and some parts fatter as the normal vectors are used to approximate the gradient field. We can see that our approach obtains the correct reconstruction result without any preprocessing and is almost free from the noise of normals.

Our approach can even reconstruct surfaces from data points without normal information if a good initial shape is specified. In this case the Hermitian information of Eq. (12) at a basis vertex is oriented by the gradient inherited from the former level PHT-spline function.

Raw scan data Our approach is able to reconstruct raw scan data by increasing the number of neighboring vertices in Hermite information estimation. Figure 10 shows the reconstruction result of raw stanford bunny scan data.

Large datasets We have reconstructed fine details from large sets of points, as summarized in Table 1. Figure 8 shows the result of Neptune model with 2,003,932 points, and the output surface has 1,733,460 triangles with a relative error 1.0e-4. Figure 7 shows another two examples:

Figure 9: Reconstruction of data with noisy normals(60 degrees rotational deviation in the origin normal direction) by different methods. From leftmost to rightmost are: Armadillo model with noisy normals, MPU reconstruction, Poisson reconstruction and our result.

Figure 8: Reconstruction of Neptune model with 2,003,932 points.

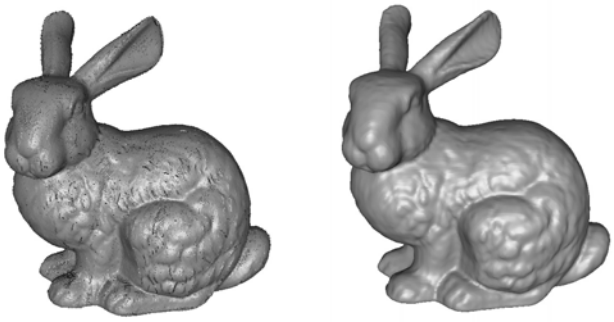

Figure 10: Reconstruction of raw scan data bunny.

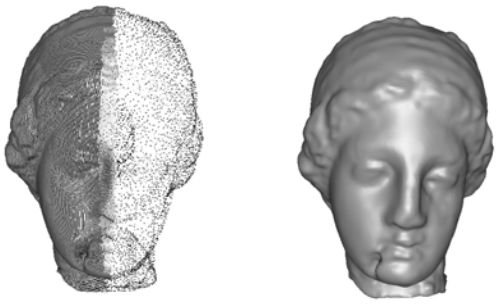

Figure 11: Reconstruction results from a non-uniform sampling set of points. Left: the Venus_half point set, Right: the reconstructed Venus_half model.

Asiandragon model with 3,609,455 points and 2,470,784 triangles, Elephant model with 1,512,290 points and 2,056,288 triangles.

Reconstruction from non-uniform sampling data and incomplete data Our adaptive reconstruction method is robust to non-uniform sampling data sets as demonstrated in Figure 11. It can also robustly reconstruct surfaces from incomplete data. Figure 12 shows the reconstruction result of the incomplete Squirrel model with holes in the back and the head.

Limitation Our approach can handle point data with light position noise as the MPA algorithm has certain smoothing effect on noise. But for the data with heavy position noise, the algorithm may generate some superfluous sheets in the vicinity of reconstruction surface. In that case a preprocessing step is needed.

5. CONCLUSION

We have presented an adaptive surface reconstruction approach based on a novel shape representation, the implicit PHT-splines. Due to the nature of PHT-splines, the reconstruction surface has a unified representation of piecewise polynomials and is globally C^1 continuous. Such represen-

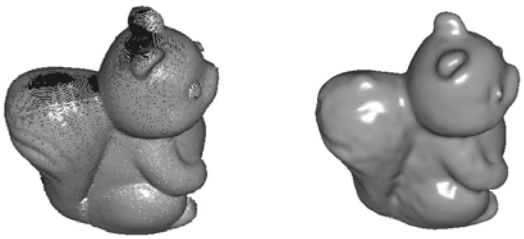

Figure 12: Reconstruction of incomplete data. Left: the incomplete Squirrel data with holes on the head and the body. Right: the reconstructed Squirrel with holes filled.

tation can be effectively applied to subsequent geometric processing, such as function evaluation, derivative calculation, subdivision and CSG operations. Our scheme has strong adaptability for describing geometric details, and at the same time is very efficient. From theoretical analysis and practical examples, we have shown that our algorithm can reconstruct high-quality surfaces several times faster than the competitive methods in the current state of art.

There are still some interesting problems for future research. One is to exploit GPU implementation of our algorithm. The other is to further improve the approach so that it can handle point clouds with heavy position noise.

6. ACKNOWLEDGEMENT

We would like to thank the anonymous reviewers for their critical comments and suggestions which greatly improve the manuscript. Thanks also go to Aim@Shape project and the Stanford Computer Graphics Laboratory for providing the models in the paper.

This work is supported by NSF of China (No.60873109, 10701069), One Hundred Talent Project of CAS and 111 project (No. B07033). Zhouwang Yang is partly supported by the Open Project Program of the State Key Lab of CAD &CG (Grant No. A0806), Zhejiang University.

7. REFERENCES

[1] M. Aigner and B. Jüttler. Robust fitting of implicitly defined surfaces using Gauss–Newton–type techniques. *The Visual Computer*, 25:731–741, 2009.

[2] N. Amenta, M. Bern, and M. Kamvysselis. A new voronoi-based surface reconstruction algorithm. In *SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 415–421, 1998.

[3] N. Amenta, S. Choi, and R. K. Kolluri. The power crust, unions of balls, and the medial axis transform. *Computational Geometry: Theory and Applications*, 19:127–153, 2001.

[4] J. Bloomenthal. An implicit surface polygonizer. pages 324–349, 1994.

[5] J. Bloomenthal and B. Wyvill, editors. *Introduction to Implicit Surfaces*. Morgan Kaufmann Publishers Inc., 1997.

[6] J.-D. Boissonnat. Geometric structures for three-dimensional shape representation. *ACM Trans. Graph.*, 3(4):266–286, 1984.

[7] J. C. Carr, R. K. Beatson, J. B. Cherrie, T. J. Mitchell, W. R. Fright, B. C. McCallum, and T. R. Evans. Reconstruction and representation of 3d objects with radial basis functions. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 67–76, 2001.

[8] F. Chazal and A. Lieutier. Topology guaranteeing manifold reconstruction using distance function to noisy data. In *SCG '06: Proceedings of the twenty-second annual symposium on Computational geometry*, pages 112–118, 2006.

[9] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 303–312, 1996.

[10] J. Deng, F. Chen, and Y. Feng. Dimensions of spline spaces over t-meshes. *J. Comput. Appl. Math.*, 194(2):267–283, 2006.

[11] J. Deng, F. Chen, X. Li, C. Hu, W. Tong, Z. Yang, and Y. Feng. Polynomial splines over hierarchical t-meshes. *Graph. Models*, 70(4):76–86, 2008.

[12] T. K. Dey and S. Goswami. Provable surface reconstruction from noisy samples. *Comput. Geom. Theory Appl.*, 35(1):124–141, 2006.

[13] H. Q. Dinh, G. Turk, and G. Slabaugh. Reconstructing surfaces by volumetric regularization using radial basis functions. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(10):1358–1371, 2002.

[14] H. Edelsbrunner and E. P. Mücke. Three-dimensional alpha shapes. *ACM Trans. Graph.*, 13(1):43–72, 1994.

[15] S. F. Frisken, R. N. Perry, A. P. Rockwood, and T. R. Jones. Adaptively sampled distance fields: a general representation of shape for computer graphics. In *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 249–254, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.

[16] M. Gross and H. Pfister. *Point-Based Graphics (The Morgan Kaufmann Series in Computer Graphics)*. Morgan Kaufmann Publishers Inc., 2007.

[17] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Surface reconstruction from unorganized points. In *SIGGRAPH '92: Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, volume 26, pages 71–78, 1992.

[18] A. Hornung and L. Kobbelt. Robust reconstruction of watertight 3d models from non-uniformly sampled point clouds without normal information. In *SGP '06: Proceedings of the fourth Eurographics symposium on Geometry processing*, pages 41–50, 2006.

[19] B. Jüttler and A. Felis. Least-squares fitting of algebraic spline surfaces. *Adv. Comput. Math.*, 17(1-2):135–152, 2002.

[20] M. Kazhdan. Reconstruction of solid models from oriented point sets. In *SGP '05: Proceedings of the third Eurographics symposium on Geometry processing*, page 73, 2005.

[21] M. Kazhdan, M. Bolitho, and H. Hoppe. Poisson surface reconstruction. In *SGP '06: Proceedings of the fourth Eurographics symposium on Geometry*

processing, pages 61–70. Eurographics Association, 2006.

[22] N. Kojekine, I. Hagiwara, and V. Savchenko. Software tools using csrbfs for processing scattered data, 2003.

[23] X. Li, J. Deng, and F. Chen. Surface modeling with polynomial splines over hierarchical t-meshes. *Vis. Comput.*, 23(12):1027–1033, 2007.

[24] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. *SIGGRAPH Comput. Graph.*, 21(4):163–169, 1987.

[25] J. Manson, G. Petrova, and S. Schaefer. Streaming surface reconstruction using wavelets. In *SGP '08: Proceedings of the Symposium on Geometry Processing*, pages 1411–1420, Aire-la-Ville, Switzerland, Switzerland, 2008. Eurographics Association.

[26] D. M.Mount and S. Arya. Ann: A library for approximate nearest neighbor searching. http://www.cs.umd.edu/ mount/ann/.

[27] B. S. Morse, T. S. Yoo, P. Rheingans, D. T. Chen, and K. R. Subramanian. Interpolating implicit surfaces from scattered surface data using compactly supported radial basis functions. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Courses*, page 78, 2005.

[28] Y. Nagai, Y. Ohtake, and H. Suzuki. Smoothing of partition of unity implicit surfaces for noise robust surface reconstruction. In *SGP '09: Proceedings of the Symposium on Geometry Processing*, pages 1339–1348, Aire-la-Ville, Switzerland, Switzerland, 2009. Eurographics Association.

[29] Y. Ohtake, A. Belyaev, M. Alexa, G. Turk, and H.-P. Seidel. Multi-level partition of unity implicits. *ACM Trans. Graph.*, 22(3):463–470, 2003.

[30] Y. Ohtake, A. Belyaev, and H.-P. Seidel. A multi-scale approach to 3d scattered data interpolation with compactly supported basis functions. In *SMI '03: Proceedings of the Shape Modeling International 2003*, page 153. IEEE Computer Society, 2003.

[31] V. Savchenko, E. A. Pasko, O. G. Okunev, and T. L. Kunii. Function representation of solids reconstructed from scattered surface points and contours. *Computer Graphics Forum*, 14:181–188, 1995.

[32] T. W. Sederberg, D. L. Cardon, G. T. Finnigan, N. S. North, J. Zheng, and T. Lyche. T-spline simplification and local refinement. *ACM Trans. Graph.*, 23(3):276–283, 2004.

[33] T. W. Sederberg, J. Zheng, A. Bakenov, and A. Nasri. T-splines and t-nurccs. In *SIGGRAPH '03: ACM SIGGRAPH 2003 Papers*, pages 477–484, 2003.

[34] X. Song and F. Chen. Adaptive surface reconstruction based on tensor product algebraic splines. *Numerical Mathematics A Journal of Chinese Universities English Series*, 2(1):90–99, 2009.

[35] X. Song, B. Jüttler, and A. Poteaux. Hierarchical spline approximation of the signed distance function. In *SMI'10: Proceedings of Shape Modeling and International*, 2010.

[36] G. Taubin. 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.*, 13(11):1115–1138, 1991.

[37] G. Turk and J. F. O'Brien. Shape transformation using variational implicit functions. In *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 335–342, 1999.

[38] G. Turk and J. F. O'brien. Modelling with implicit surfaces that interpolate. *ACM Trans. Graph.*, 21(4):855–873, 2002.

[39] H. Yang, M. Fuchs, B. Jüttler, and O. Scherzer. Evolution of T-spline level sets with distance field constraints for geometry reconstruction and image segmentation. In *Shape Modeling International*, pages 247–252. IEEE Press, 2006.

[40] H. Yang and B. J. üttler. Evolution of t-spline level sets for meshing non–uniformly sampled and incomplete data. *The Visual Computer*, 24:435–448, 2008.

[41] Z. Yang and T.-w. Kim. Moving parabolic approximation of point clouds. *Computer-Aided Design*, 39(12):1091–1112, 2007.

[42] H.-K. Zhao, S. Osher, and R. Fedkiw. Fast surface reconstruction using the level set method. In *IEEE Workshop on Variational and Level Set Methods in Computer Vision*, pages 194–201, 2001.