Computer Aided Geometric Design 28 (2011) 463–474

Contents lists available at ScienceDirect

Computer Aided Geometric Design

www.elsevier.com/locate/cagd

Parallel and adaptive surface reconstruction based on implicit PHT-splines

Jun Wang[a], Zhouwang Yang[a,*], Liangbing Jin[b], Jiansong Deng[a], Falai Chen[a,*]

[a] University of Science and Technology of China, Hefei 230026, PR China
[b] Zhejiang Normal University, Jinhua 321004, PR China

ARTICLE INFO

Article history:
Available online 28 July 2011

Keywords:
PHT-spline
T-mesh
Surface reconstruction
Hermite interpolation

ABSTRACT

We present a new surface reconstruction framework, which uses the implicit PHT-spline for shape representation and allows us to efficiently reconstruct surface models from very large sets of points. A PHT-spline is a piecewise tri-cubic 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 take full advantage of 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. Examples show that our approach can produce high quality reconstruction surfaces very efficiently. We also present the multi-threaded algorithm of our approach and show its parallel scalability.

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-

[*] Corresponding authors.
E-mail addresses: yangzw@ustc.edu.cn (Z. Yang), chenfl@ustc.edu.cn (F. Chen).

0167-8396/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.cagd.2011.06.004

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 (Gross and Pfister, 2007), 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 (Amenta et al., 1998) 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 (Boissonnat, 1984), alpha-shapes (Edelsbrunner and Mücke, 1994), power crust (Amenta et al., 2001), come with theoretical guarantees of correct reconstruction if the sample meets certain conditions (dense and noiseless). Some recent works (Chazal and Lieutier, 2006; Dey and Goswami, 2006; Hornung and Kobbelt, 2006) addressed the issue on noisy and non-uniform samples.

The level-set method was applied to surface reconstruction in Zhao et al. (2001), where the problem is formulated in terms of a partial differential equation (PDE) describing the surface evolution. In Yang et al. (2006), Yang and Jüttler (2008), 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 Aigner and Jüttler (2009), which can also be viewed as the discrete iterative version of a time-dependent evolution process.

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 Hoppe et al. (1992) a signed distance function is estimated as the distance to the tangent plane of the closest point. Curless and Levoy (1996) blended the directional distance associated with each range scan, using Gaussian weights to form the implicit function. In Jüttler and Felis (2002), an approach for fitting implicitly defined algebraic spline surfaces to scattered data is presented, which simultaneously approximate points and associated normal vectors. An improvement of this method is to adaptively choose the knots such that the reconstructed surface fits point cloud in a gradual refinement process (Song and Chen, 2009). The PHT-splines (Hu, 2006; Deng et al., 2008; Jin, 2008) have some good properties for the applications of geometric modeling, such as natural hierarchical structure and local adaptivity. Hu (2006) uses the implicit PHT-spline to reconstruct surface from point cloud, where the control coefficients of the new basis functions in each level are computed by solving a local linear equation. In Song et al. (2010), the PHT-spline is used to approximate the signed distance function of a smooth curve or implicit defined surface. In Turk and O'Brien (1999), Carr et al. (2001), Dinh et al. (2002), Turk and O'Brien (2002), implicit functions are constructed with polyharmonic radial basis functions (RBFs) (Savchenko et al., 1995), by placing zero constraints at each input point and also a pair of non-zero constraints at "offset-surface" points. Morse et al. (2005), Kojekine et al. (2003) and Ohtake et al. (2003b) employed compactly supported RBFs to reduce the computation and speed up the reconstruction process. Ohtake et al. (2003a) 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 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 (Nagai et al., 2009).

Besides signed distance functions, indicator functions are also often used in surface reconstruction. Points equipped with oriented normals can be viewed as samples of the gradient of an indicator function (Kazhdan, 2005; Kazhdan et al., 2006; Manson et al., 2008). In Kazhdan (2005), 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 Kazhdan et al. (2006), 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. In Manson et al. (2008), a streaming surface reconstruction using wavelets to represent the indicator function is proposed. Due to the multiresolution nature of wavelets, the 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 (Ohtake et al., 2003a), our implicit PHT-spline representation could be viewed as an adaptive signed distance fields (Frisken et al., 2000) 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 surface reconstruction framework based on the implicit PHT-spline 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.
- ◇ A PHT-spline function in each cell is a tri-cubic 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 surface. Thus it has less dependency on the normal information and is robust in the presence of noisy normals.
- ◇ Our approach holds the data parallelism property. It is scalable for multi-core systems and almost reaches the ideal speed-up ratio in multi-threads.

The paper is organized as follows. In Section 2, a shape representation called implicit PHT-spline surface is introduced to describe the target geometry. In Section 3, we present the main framework of the adaptive surface reconstruction based on the implicit PHT-splines. In Section 4, we give implementation details and demonstrate the effectiveness of our approach through various examples. In Section 5, we present the multi-threaded algorithm of our approach and show its parallel scalability. 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 (Bloomenthal and Wyvill, 1997). 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 (Jüttler and Felis, 2002). 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 difficulties in local refinement and adaptivity. To eliminate superfluous control points in tensor–product B-spline (or NURBS), Sederberg et al. (2003, 2004) 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 (Deng et al., 2006, 2008). The PHT-splines are then used for surface fitting (Hu, 2006; Li et al., 2007; Deng et al., 2008; Jin, 2008) and the approximation of signed distance field (Song et al., 2010). In this paper, we will apply the implicit PHT-splines to reconstruction surface from point clouds.

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 paralleling to xoy, yoz and zox, into eight sub-cubes which are labeled as the cells of level $k+1$. Fig. 1 shows an example of 2D hierarchical T-mesh.

Given a 3D hierarchical T-mesh $\mathscr{T}$, denote by Φ the set of all the cells in $\mathscr{T}$. We define a tri-cubic polynomial spline space over $\mathscr{T}$ as

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

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 (Deng et al., 2006) is

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

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 Deng et al.

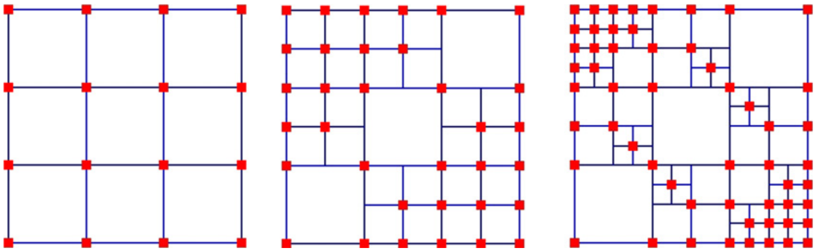

Fig. 1. A 2D hierarchical T-mesh (from left to right are level 0, level 1 and level 2, the red squares denote the basis vertices). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

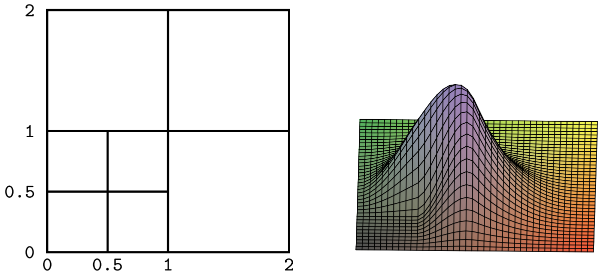

Fig. 2. A basis function of 2D PHT-spline associated with the vertex (1, 1).

(2008) for polynomial splines over 2D hierarchical T-meshes can be directly extended to construct 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. Fig. 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-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

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

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

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

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

$$\begin{aligned}\mathcal{H}f(\mathbf{x}_{i_0})&=\mathcal{H}\sum_{i=1}^{V}\sum_{j=1}^{8}C_{ij}b^{ij}(\mathbf{x}_{i_0})\\&=\sum_{i=1}^{V}\sum_{j=1}^{8}C_{ij}\mathcal{H}b^{ij}(\mathbf{x}_{i_0})=\sum_{j=1}^{8}C_{i_0j}\mathcal{H}b^{i_0j}(\mathbf{x}_{i_0})\\&=\mathbf{C}_{i_0}\mathbf{B},\end{aligned} \tag{5}$$

where $\mathbf{C}_{i_0}=(C_{i_01},\ldots,C_{i_08})$ is a 1×8 control coefficient vector, $\mathbf{B}=(\mathcal{H}b^{i_01}(\mathbf{x}_{i_0})^T,\ldots,\mathcal{H}b^{i_08}(\mathbf{x}_{i_0})^T)^T$ 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

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

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.

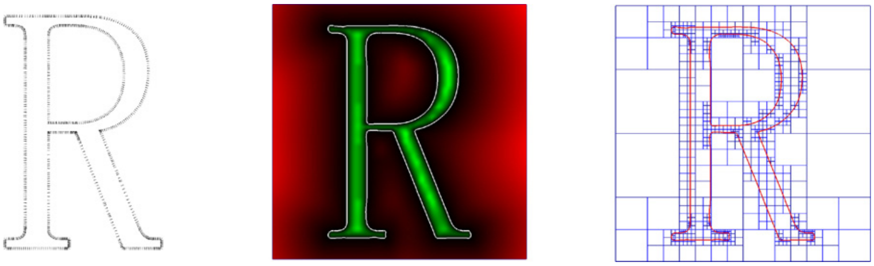

Fig. 3. Left: a set of points with normals from letter 'R'; Middle: the signed distance function and the zero level set; Right: the hierarchical T-mesh and the reconstructed implicit PHT-spline curve.

3. Approach

Given a set of points $\mathcal{P} = \{\mathbf{p}_1, \ldots, \mathbf{p}_N\}$ with oriented normals, 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 in regions away from the surface. 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 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. 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. (7). 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. (8) or Eq. (10). 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 (Taubin, 1991) and is compared with a user-specified threshold value ε_0, i.e.,

$$\max_{\|\mathbf{p}_i-\mathbf{c}\|<\frac{\sqrt{3}}{2}\Delta^{(k)}} \frac{|f^{[k]}(\mathbf{p}_i)|}{\|\nabla f^{[k]}(\mathbf{p}_i)\|} > \varepsilon_0. \tag{7}$$

If Eq. (7) holds, the cell is subdivided; otherwise, it is not subdivided. Fig. 3 illustrates a 2D curve reconstruction example, where the resulting hierarchical T-mesh is provided to demonstrate how the error-guided adaptive reconstruction works. Fig. 4 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 estimated by using planar fitting technique; Otherwise, Moving Parabolic Approximation (MPA) (Yang and Kim, 2007) technique is applied. The two techniques respectively provide a first order and a second order approximation to the signed distance function. The aforementioned simple strategy will reduce in theory the potential approximation order of the PHT-spline which is a tri-cubic polynomial in each cell. However,

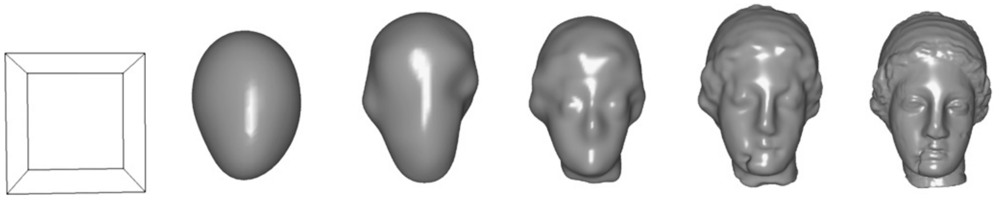

Fig. 4. Adaptive reconstruction of Igea model, from left to right are the intermediate results at level 0 to level 5. The leftmost is the T-mesh at level 0 with which there is no surface generated.

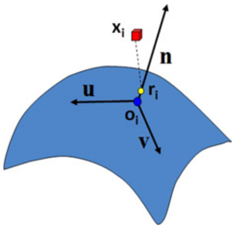

Fig. 5. Hermitian information estimation by using the MPA algorithm.

due to the numerical unreliability of estimation of high order differential quantities, this is probably a compromise way in practice.

To estimate the Hermite information vector at $\mathbf{x}_i$, we first compute its nearest point $\mathbf{r}_i \in \mathcal{P}$ and denote $\mathbf{d}_i = \mathbf{x}_i - \mathbf{r}_i$. 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. Here δ is a preset distance which is usually set to be 2% of the diagonal length of the domain Ω in our implementation. Let $\mathbf{h} = (h, h_x, h_y, h_z, h_{xy}, h_{yz}, h_{zx}, h_{xyz})$ be the Hermitian information vector at a basis vertex $\mathbf{x}_i$. For a far-away basis vertex $\mathbf{x}_i$, the mixed derivatives in the Hermitian information vector are set to zero because of little effect on the reconstructed surface. So we get

$$\begin{cases} h = \operatorname{sign}(\mathbf{x}_i)\|\mathbf{d}_i\|, \\ (h_x, h_y, h_z) = \operatorname{sign}(\mathbf{x}_i)\mathbf{d}_i/\|\mathbf{d}_i\|, \\ h_{xy} = h_{yz} = h_{zx} = h_{xyz} = 0. \end{cases} \tag{8}$$

In order to check if the vertex $\mathbf{x}_i$ is inside $(\operatorname{sign}(\mathbf{x}_i) < 0)$ or outside $(\operatorname{sign}(\mathbf{x}_i) > 0)$ 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 passing through $\mathbf{r}_i$ using principal component analysis (PCA). The orientation of plane is determined such that the oriented plane normal is consistent (has positive inner product value) with most normals of the nearest points. Finally, we denote by $\operatorname{sign}(\mathbf{x}_i)$ the sign of the inner product between the vector $\mathbf{d}_i$ and the oriented plane normal.

The Hermite information at a near-by basis vertex $\mathbf{x}_i$ is required to be accurately estimated. We employ the MPA algorithm (Yang and Kim, 2007) to compute geometric quantities of the underlying surface and then deduce theoretically the Hermitian information from the obtained geometric quantities. It is well known that the differential quantities provide convenient bases for characterizing the local behavior of a shape in the vicinity of a particular point. 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. We search N_{nb} (usually set $N_{\text{nb}} = 20$ in our implementation) nearest points in $\mathcal{P}$ for a given reference $\mathbf{r}_i$. By using the MPA algorithm, we can compute the location of the underlying surface of unorganized points and simultaneously estimate the differential quantities of the surface. As shown in Fig. 5, the output of 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). In the local coordinates $(s, t, w) = (\mathbf{x} - \mathbf{o}_i)^T(\mathbf{u}, \mathbf{v}, \mathbf{n})$, the osculating paraboloid of the underlying surface can be expressed as

$$w = \psi(s, t) = \frac{1}{2}\left(as^2 + 2bst + ct^2\right). \tag{9}$$

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

$$\begin{cases} h = w_i - \frac{1}{2}\left(as_i^2 + 2bs_it_i + ct_i^2\right), \\ h_x = n_1 - (as_i + bt_i)u_1 - (bs_i + ct_i)v_1, \\ h_y = n_2 - (as_i + bt_i)u_2 - (bs_i + ct_i)v_2, \\ h_z = n_3 - (as_i + bt_i)u_3 - (bs_i + ct_i)v_3, \\ h_{xy} = -(au_2 + bv_2)u_1 - (bu_2 + cv_2)v_1, \\ h_{yz} = -(au_3 + bv_3)u_2 - (bu_3 + cv_3)v_2, \\ h_{xy} = -(au_1 + bv_1)u_3 - (bu_1 + cv_1)v_3, \\ |h_{xyz} = 0, \end{cases} \tag{10}$$

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 could be opposite to the orientation of the surface. To address this issue, a reorientation of the Hermitian information 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 @2.8 GHz processor and 4.0 GB 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 library (Mount and Arya, 2006). 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 adjacency information of the cells, parent–children relationship and cell–vertices relationship.

A PHT-spline function is a tri-cubic 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 (Lorensen and Cline, 1987) and Bloomenthal's method (Bloomenthal, 1994). In our implementation, we employ the Bloomenthal's polygonizer to extract the isosurface with the same strategy used in MPU method (Ohtake et al., 2003a).

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 (Carr et al., 2001; Ohtake et al., 2003b). 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 (Ohtake et al., 2003a) and Poisson (Kazhdan et al., 2006) reconstruction methods. The performance statistics are summarized in Table 1.

4.2.1. 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 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 summarize 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 the Poisson method.

Table 1
The performance statistics.

Model	#Points	Rel. error	IPHT			MPU			Poisson		
			Time	Mem.	#Tris.	Time	Mem.	#Tris.	Time	Mem.	# Tris.
Asian dragon	3,609,455	2.0e−4	41.342 s	738 M	2,470,784	199.707 s	850 M	2,449,768	706.0 s	1058 M	7,422,418
Neptune	2,003,932	1.0e−4	25.686 s	494 M	1,733,460	142.688 s	736 M	1,734,896	306.8 s	505 M	3,211,190
Elephant	1,512,290	3.0e−4	22.713 s	390 M	2,056,288	89.945 s	492 M	2,032,128	229.6 s	537 M	2,577,824
Armadillo	172,974	5.0e−4	3.544 s	88 M	238,592	17.644 s	229 M	240,012	16.2 s	163 M	204,302
Noisy Armadillo	172,974	5.0e−4	3.544 s	88 M	238,592	64.148 s	240 M	240,204	17.2 s	163 M	189,436
Gargoyle	863,210	5.0e−4	10.583 s	239 M	853,136	53.313 s	320 M	645,420	112.3 s	370 M	1,332,000
Filigree	514,300	5.0e−4	6.861 s	168 M	505,320	34.129 s	293 M	501,136	61.4 s	238 M	668,760
Bunny	362,272	8.0e−4	10.219 s	212 M	347,712	40.513 s	280 M	320,932	84.8 s	243 M	783,112

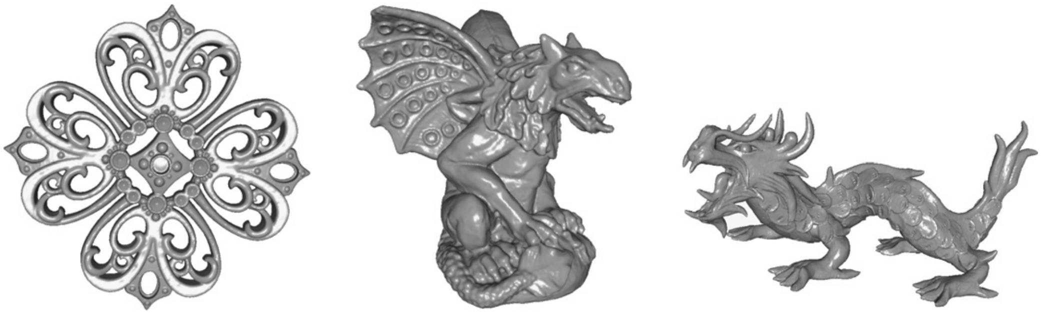

Fig. 6. Reconstruction of different models (Filigree model with 514,300 points, Gargoyle model with 863,210 points and Asian dragon model with 3,609,455 points.

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

Model	# cells	# vts.	# b-vts.	# cells (MPU)
Asian dragon	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

4.2.2. 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. 4). Our algorithm reconstructs high-quality surfaces with very fine details, as shown in Fig. 6. 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 approach results in a piecewise polynomial while the MPU method has more complicated expression.

4.2.3. Memory costs

In our algorithm, the memory 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 Figs. 6–8) under the same 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).

4.2.4. Raw scan data and non-uniform sampling data

Our approach is able to reconstruct raw scan data by increasing the number of neighboring vertices in Hermite information estimation. Fig. 7(a) shows the reconstruction result of raw Stanford bunny scan data. Here the oriented normals are estimated using the connectivity information within each individual range scan. The reconstruction approach is also robust for non-uniform sampling data sets as demonstrated in Fig. 7(b).

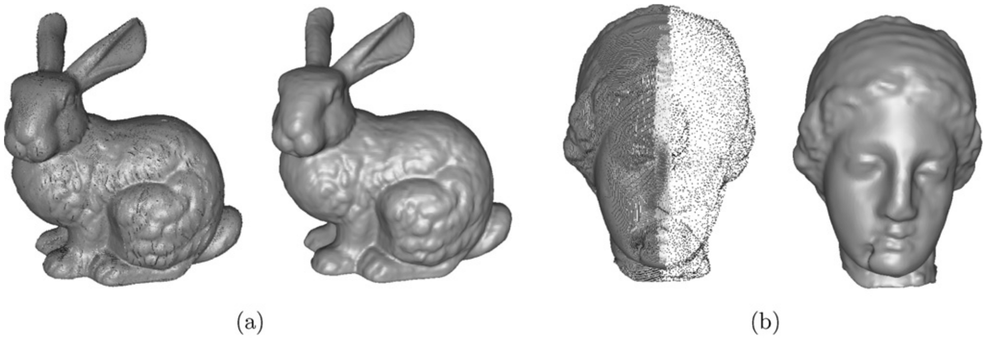

Fig. 7. (a) Reconstruction of raw scan data bunny; (b) Reconstruction results from a non-uniform sampling set of points.

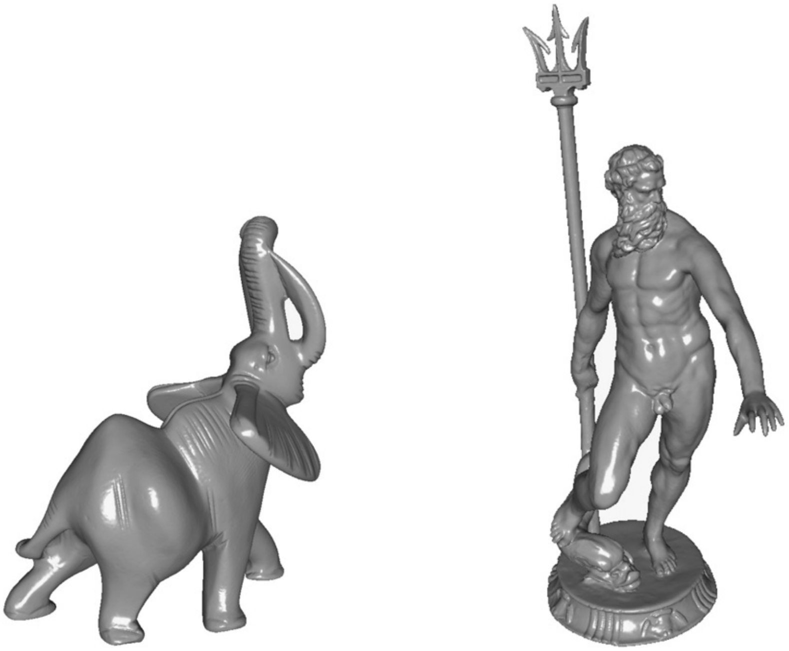

Fig. 8. Reconstruction of Elephant model with 1,512,290 points and Neptune model with 2,003,932 points.

4.2.5. Large datasets

We have reconstructed fine details from large sets of points, as summarized in Table 1. Fig. 6 shows the result of Asian dragon model with 3,609,455 points, and the output surface has 2,470,784 triangles with a relative error 1.0e−4. Fig. 8 shows another two examples: Neptune model with 2,003,932 points and 1,733,460 triangles, Elephant model with 1,512,290 points and 2,056,288 triangles.

4.2.6. 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. Unlike some previous methods that heavily depend on the normals of the point cloud, our approach only uses them to orient the Hermitian information vectors. Therefore it is conceivable that our algorithm is robust in the presence of certain noise of input normals. Fig. 9 shows the reconstructed 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. (10) at a basis vertex is oriented by the gradient inherited from the former level PHT-spline function.

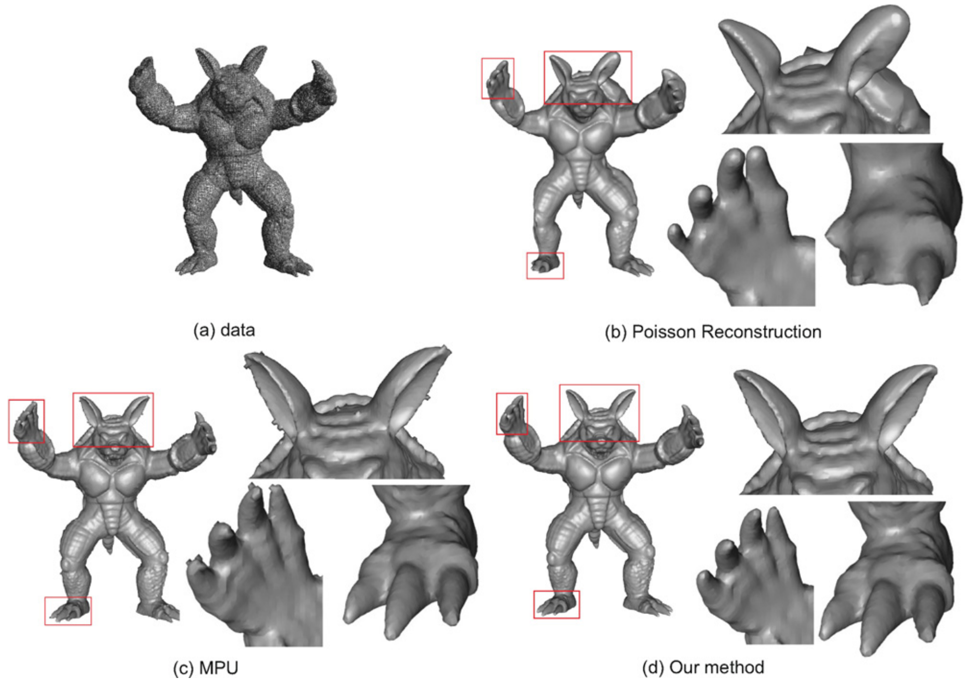

Fig. 9. Reconstruction of data with noisy normals (60 degrees random rotational deviation in the origin normal direction) by different methods.

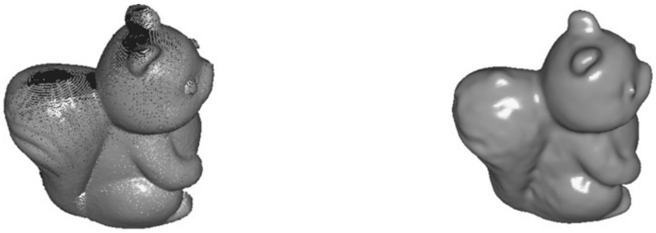

Fig. 10. Reconstruction of incomplete data. Left: the incomplete Squirrel data with holes on the head and the body. Right: the reconstructed Squirrel with holes filled.

4.2.7. Reconstruction from incomplete data

Our adaptive algorithm can also robustly reconstruct surfaces from incomplete data. Fig. 10 shows the reconstruction result of the incomplete Squirrel model with holes in the back and the head.

4.2.8. 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. Scalable parallel reconstruction via multi-threads

A striking feature of our proposed method is its data parallelism. The adaptive surface reconstruction using implicit PHT-splines involves three primitive operations: approximation error calculation, cell subdivision, and Hermitian information estimation. At each level, these three operations are called multiple times. The calculations of approximation errors in the cells of current level are independent and can be carried out simultaneously, so do the operations of cell subdivision and Hermitian information estimation. Consequently, massive calls of above mentioned operations are simply parallelized at each level. The parallel computing for each level is described as follows:

1) Collect cells of current level and compute the approximation error for each cell in parallel.
2) Collect the cells in which the approximation error is bigger than a specified threshold value and subdivide them in parallel.
3) Collect new basis vertices and estimate the Hermitian information at each new basis vertex in parallel.

Table 3
Multi-threaded parallel.

Model	1 thread			2 threads			4 threads		
	T (Err.)	T (Sub.)	T (Her.)	T (Err.)	T (Sub.)	T (Her.)	T (Err.)	T (Sub.)	T (Her.)
Asian dragon	21.858 s	8.032 s	3.782 s	13.212 s	5.100 s	1.932 s	6.832 s	3.302 s	1.112 s
Elephant	9.969 s	2.926 s	1.671 s	5.484 s	1.610 s	0.890 s	2.983 s	1.189 s	0.500 s
Gargoyle	5.384 s	2.563 s	1.437 s	3.312 s	1.656 s	0.749 s	2.626 s	1.298 s	0.389 s

We have tested our parallel reconstruction algorithm using multi-threaded implementation on a multi-core computer system (Intel Quad-Core 3.0 GHz). Experiments demonstrate that the algorithm is scalable. Table 3 shows the running time for the three primitive operations (approximation error calculation, cell subdivision, and Hermitian information estimation) in the parallel reconstruction with multi-threads. Besides the extra cost, the running time of 4-threads is about a quarter of single-threaded, which almost reaches the ideal speed-up ratio.

6. 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 representation 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.

Utilizing the available parallelism, we have presented the multi-threaded implementation of the algorithm and showed its scalable potential in parallel. There are still some interesting problems for future research. One is to exploit GPU implementation of our approach. The other is to further improve the approach so that it can handle point clouds with heavy position noise.

Acknowledgements

We would like to thank the anonymous reviewers for their 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. The authors are supported by 973 Program 2011CB302400, the NSF of China (Nos. 11031007, 61073108, 61073111 and 60873109), and Program for New Century Excellent Talents in University (No. NCET-08-0514). Zhouwang Yang is supported in part by the Fundamental Research Funds for the Central Universities (No. WK0010000009) and the Open Project Program of the State Key Lab of CAD&CG (Grant No. A0806).

References

Aigner, M., Jüttler, B., 2009. Robust fitting of implicitly defined surfaces using Gauss–Newton-type techniques. Vis. Comput. 25, 731–741.

Amenta, N., Bern, M., Kamvysselis, M., 1998. A new Voronoi-based surface reconstruction algorithm. In: Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques. ACM, pp. 415–421.

Amenta, N., Choi, S., Kolluri, R.K., 2001. The power crust, unions of balls, and the medial axis transform. Comput. Geom. Theory Appl. 19, 127–153.

Bloomenthal, J., 1994. An implicit surface polygonizer. In: Graphics Gems IV, pp. 324–349.

Bloomenthal, J., Wyvill, B. (Eds.), 1997. Introduction to Implicit Surfaces. Morgan Kaufmann Publishers Inc.

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

Carr, J.C., Beatson, R.K., Cherrie, J.B., Mitchell, T.J., Fright, W.R., McCallum, B.C., Evans, T.R., 2001. Reconstruction and representation of 3D objects with radial basis functions. In: Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques. ACM, pp. 67–76.

Chazal, F., Lieutier, A., 2006. Topology guaranteeing manifold reconstruction using distance function to noisy data. In: Proceedings of the 22nd Annual Symposium on Computational Geometry. ACM, pp. 112–118.

Curless, B., Levoy, M., 1996. A volumetric method for building complex models from range images. In: Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques. ACM, pp. 303–312.

Deng, J., Chen, F., Feng, Y., 2006. Dimensions of spline spaces over T-meshes. J. Comput. Appl. Math. 194 (2), 267–283.

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

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

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

Edelsbrunner, H., Mücke, E.P., 1994. Three-dimensional alpha shapes. ACM Trans. Graph. 13 (1), 43–72.

Frisken, S.F., Perry, R.N., Rockwood, A.P., Jones, T.R., 2000. Adaptively sampled distance fields: a general representation of shape for computer graphics. In: Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques. ACM Press/Addison-Wesley Publishing Co., pp. 249–254.

Gross, M., Pfister, H., 2007. Point-Based Graphics. The Morgan Kaufmann Series in Computer Graphics. Morgan Kaufmann Publishers Inc.

Hoppe, H., DeRose, T., Duchamp, T., McDonald, J., Stuetzle, W., 1992. Surface reconstruction from unorganized points. In: Proceedings of the 19th Annual Conference on Computer Graphics and Interactive Techniques, vol. 26. ACM, pp. 71–78.

Hornung, A., Kobbelt, L., 2006. Robust reconstruction of watertight 3D models from non-uniformly sampled point clouds without normal information. In: Proceedings of the Fourth Eurographics Symposium on Geometry Processing. Eurographics Association, pp. 41–50.

Hu, C., April, 2006. Geometric modeling based on dynamic implicit surfaces. PhD thesis. Department of Mathematics, University of Science and Technology of China, Advisor: Falai Chen.

Jin, L., December, 2008. Theory and application of splines over hierarchical T-meshes. PhD thesis. Department of Mathematics, University of Science and Technology of China, Advisor: Falai Chen.

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

Kazhdan, M., 2005. Reconstruction of solid models from oriented point sets. In: Proceedings of the Third Eurographics Symposium on Geometry Processing. Eurographics Association, pp. 73–82.

Kazhdan, M., Bolitho, M., Hoppe, H., 2006. Poisson surface reconstruction. In: Proceedings of the Fourth Eurographics Symposium on Geometry Processing. Eurographics Association, pp. 61–70.

Kojekine, N., Hagiwara, I., Savchenko, V., 2003. Software tools using CSRBFs for processing scattered data. Comput. Graph. 27 (2), 311–319.

Li, X., Deng, J., Chen, F., 2007. Surface modeling with polynomial splines over hierarchical T-meshes. Vis. Comput. 23 (12), 1027–1033.

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

Manson, J., Petrova, G., Schaefer, S., 2008. Streaming surface reconstruction using wavelets. In: Proceedings of the Sixth Symposium on Geometry Processing. Eurographics Association, pp. 1411–1420.

Morse, B.S., Yoo, T.S., Rheingans, P., Chen, D.T., Subramanian, K.R., 2005. Interpolating implicit surfaces from scattered surface data using compactly supported radial basis functions. In: ACM SIGGRAPH 2005 Courses. ACM.

Mount, D.M., Arya, S., 2006. ANN: A library for approximate nearest neighbor searching. http://www.cs.umd.edu/~mount/ann/.

Nagai, Y., Ohtake, Y., Suzuki, H., 2009. Smoothing of partition of unity implicit surfaces for noise robust surface reconstruction. In: Proceedings of the Seventh Symposium on Geometry Processing. Eurographics Association, pp. 1339–1348.

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

Ohtake, Y., Belyaev, A., Seidel, H.-P., 2003b. A multi-scale approach to 3D scattered data interpolation with compactly supported basis functions. In: Proceedings of the Shape Modeling International Conference 2003. IEEE Computer Society, p. 153.

Savchenko, V., Pasko, E.A., Okunev, O.G., Kunii, T.L., 1995. Function representation of solids reconstructed from scattered surface points and contours. Comput. Graph. Forum 14 (4), 181–188.

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

Sederberg, T.W., Zheng, J., Bakenov, A., Nasri, A., 2003. T-splines and T-NURCCs. ACM Trans. Graph. 22 (3), 477–484.

Song, X., Chen, F., 2009. Adaptive surface reconstruction based on tensor product algebraic splines. Numer. Math. J. Chinese Univ. (English Ser.) 2 (1), 90–99.

Song, X., Jüttler, B., Poteaux, A., 2010. Hierarchical spline approximation of the signed distance function. In: Proceedings of the Shape Modeling International Conference 2010. IEEE Computer Society, pp. 241–245.

Taubin, G., 1991. 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.

Turk, G., O'Brien, J.F., 1999. Shape transformation using variational implicit functions. In: Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques. ACM Press/Addison-Wesley Publishing Co., pp. 335–342.

Turk, G., O'Brien, J.F., 2002. Modelling with implicit surfaces that interpolate. ACM Trans. Graph. 21 (4), 855–873.

Yang, H., Fuchs, M., Jüttler, B., Scherzer, O., 2006. Evolution of T-spline level sets with distance field constraints for geometry reconstruction and image segmentation. In: Proceedings of the IEEE International Conference on Shape Modeling and Applications 2006. IEEE Computer Society, pp. 247–252.

Yang, H., Jüttler, B., 2008. Evolution of T-spline level sets for meshing non-uniformly sampled and incomplete data. Vis. Comput. 24, 435–448.

Yang, Z., Kim, T.-W., 2007. Moving parabolic approximation of point clouds. Comput. Aided Des. 39 (12), 1091–1112.

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