

# Phase-field guided surface reconstruction based on implicit hierarchical B-splines



Maodong Pan, Weihua Tong, Falai Chen \*

School of Mathematical Sciences, University of Science and Technology of China, Hefei, Anhui, 230026, PR China

## ARTICLE INFO

### Article history:

Available online 24 March 2017

### Keywords:

Implicit surface

Surface reconstruction

Hierarchical B-splines

Phase-field

## ABSTRACT

Constructing smooth surface representations from point clouds is a fundamental problem in geometric modeling and computer graphics, and a wealthy of literature has focused on this problem. Among the many approaches, implicit surface reconstruction has been a central topic in the past two decades due to its ability to represent objects with complicated geometry and topology. Recently, the problem of reducing the storage requirement for implicit representations has attracted much attention. In this paper, we propose a phase-field guided implicit surface reconstruction method to tackle this problem. The implicit function of our method behaves like the phase-field of a binary system, in which it takes distinct values (i.e.,  $-1$  and  $1$ ) in each of the phases with a smooth transition between them. Given an unorganized point cloud, we present a method to construct a phase-field function represented by a hierarchical B-spline whose zero level set approximates the point cloud as much as possible. Unlike previous approaches, our mathematical model avoids the use of the normal information of the point cloud. Furthermore, as demonstrated by experimental results, our method can achieve very compact representation since we mainly need to save the coefficients of the hierarchical B-spline function within a narrow band near the point cloud. The ability of our method to produce reconstruction results with high quality is also validated by experiments.

© 2017 Published by Elsevier B.V.

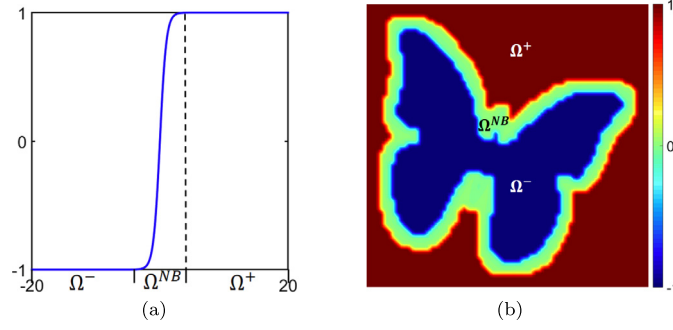
## 1. Introduction

Surface reconstruction prevails in creating digital models of real world objects. Owing to the recent development in 3D scanning technology, and consumer-level depth sensors such as Microsoft Kinect, the acquisition cost of 3D objects decreases dramatically. As the diversity, ease of use, and popularity of 3D acquisition methods continue to increase, so does the need for the development of new surface reconstruction techniques. A recent survey on the development of surface reconstruction can be found in Berger et al. (2014).

Among various surface reconstruction methods, implicit surfaces are the predominant representations. The underlying reason is that implicit surfaces are more suitable for reconstructing surfaces from data sets that are noisy, incomplete or non-uniformly distributed than other surface representations such as polygonal meshes, parametric surfaces and subdivision surfaces. In addition, implicit representations greatly facilitate the classification problem of whether a given point is on, inside or outside a surface, and are able to represent shapes with complicated topology and geometry, even with dynamic topology (Bloomberg and Wyvill, 1997; Gomes et al., 2009). Consequently, many representations of implicit surfaces have

\* Corresponding author.

E-mail addresses: mdpan@mail.ustc.edu.cn (M. Pan), tongwh@ustc.edu.cn (W. Tong), chenfl@ustc.edu.cn (F. Chen).



**Fig. 1.** (a) and (b) are two phase-field functions in 1D and 2D respectively, whose values change smoothly from  $-1$  in  $\Omega^-$  to  $1$  in  $\Omega^+$  near the interface  $\Omega^{NB}$ .

been proposed, e.g. Blobby models (Muraki, 1991), signed distance fields (Hoppe et al., 1992), radial basis functions (RBFs) (Carr et al., 2001), moving least squares (MLS) surfaces (Levin, 2003; Shen et al., 2004; Feng et al., 2014), algebraic spline (AS) surfaces (Jüttler and Felis, 2002), multilevel partition of unity (MPU) (Ohtake et al., 2003) and so on.

An implicit surface  $S$  is identified as the zero level-set of a function  $f(\mathbf{p}) : \Omega \subset \mathbb{R}^3 \rightarrow \mathbb{R}$ . That is

$$S = f^{-1}(0) = \{\mathbf{x} \in \Omega \mid f(\mathbf{x}) = 0\}.$$

Implicit surfaces are a volume-based representation that enables many operations to be performed easily, such as inside/outside test, offset, blending and boolean operations. However, the function information at the position far from  $S$  is usually less useful. Moreover, the Jordan–Brouwer separation theorem states that an implicit surface  $S \in \mathbb{R}^3$  separates  $\mathbb{R}^3$  into the surface itself and two connected open sets, i.e., the outside and the inside. Therefore, the most compact implicit function is the characteristic function of  $S$  that is defined as

$$\chi(\mathbf{p}) = \begin{cases} -1 & \text{if } \mathbf{p} \in \Omega_-, \\ 0 & \text{if } \mathbf{p} \in S, \\ +1 & \text{if } \mathbf{p} \in \Omega_+, \end{cases}$$

where  $\Omega = \Omega_- \cup S \cup \Omega_+$ , with  $\Omega_-$  and  $\Omega_+$  being the inside and outside of  $S$  respectively. In order to have a smooth implicit surface  $S$ , we require that the implicit function  $f(\mathbf{p})$  meet some continuity criterions, e.g.  $f(\mathbf{p}) \in C^1$  and  $\nabla f(\mathbf{p}) \neq \mathbf{0}$  for all  $\mathbf{p} \in \Omega$ , here  $\nabla f$  is the gradient of  $f$ . So, the idealized implicit function should behave like the phase-field of a binary system as follows

$$\phi(\mathbf{p}) = \begin{cases} -1 & \text{if } \mathbf{p} \in \Omega^-, \\ [-1, +1] & \text{if } \mathbf{p} \in \Omega^{NB} \supset S, \\ +1 & \text{if } \mathbf{p} \in \Omega^+, \end{cases}$$

where  $\Omega$  is the disjoint union of  $\Omega^-$ ,  $\Omega^{NB}$  and  $\Omega^+$ ,  $\Omega^- \subset \Omega_-$ ,  $\Omega^+ \subset \Omega_+$ , and  $\Omega^{NB}$  is a narrow band near the implicit surface  $S$ . The value of function  $\phi(\mathbf{p})$  changes smoothly from  $-1$  to  $+1$  over the region  $\Omega^{NB}$ , and is constant (i.e.,  $1$  or  $-1$ ) otherwise as demonstrated in Fig. 1 for one and two dimensional cases. It is obvious that the idealized implicit function is more suitable for compression than other implicit representations, since it always takes constant values in the region  $\Omega^- \cup \Omega^+$ .

In this paper, we propose a phase-field guided surface reconstruction method based on the above observation. Given a point cloud, we present a method to construct an implicit function which approximates the phase-field of the point cloud. The implicit function takes values  $-1$  inside,  $+1$  outside and a hierarchical B-spline in a narrow band of the point cloud. Such representation is very compact and uses much less storage space than other representations, and thus it may save computational cost in subsequent operations (e.g., function evaluation) and reduce the transmission time of the information of the implicit representation on internet. Furthermore, the hierarchical B-spline representation in the narrow band can capture the geometric details of the reconstructed surface, due to the adaptivity, local refinement and nonnegativity of their basis functions. When compared with some global reconstruction methods, our method only needs to determine the unknown coefficients of hierarchical B-spline basis functions within the narrow band, which enables us to cut down the cost of computation. On the other hand, our method generates a global representation that is still able to perform inside/outside test, boolean operations and so on, while the existing narrow band methods cannot.

The organization of this article is as follows. In Section 2, we review related work. Section 3 introduces some preliminary knowledge about hierarchical B-splines. Section 4 presents the mathematical model and the algorithm for our phase-field guided surface reconstruction method. Experimental results and performance of our algorithm are shown in Section 5. Comparisons with the state-of-the-art methods are also provided. Section 6 concludes the paper with proposals for future work.

## 2. Related work

In this section, we focus on reviewing some related work on compact representations of implicit surface reconstruction, narrow band methods and hierarchical B-splines. For a comprehensive survey on surface reconstruction, the reader is referred to Berger et al. (2014) and references therein.

### 2.1. Compact representations and narrow band methods

Given a sampling set of point clouds measured from an object, implicit surface reconstruction approaches usually create an implicit function or an indicator function for the underlying surface the point cloud is sampled from. In the past several decades, various implicit representations have been proposed and most of them are expressed as a linear combination of a set of basis functions. To capture the fine-scale details of complex models, most existing surface reconstruction methods build a large number of basis functions and need to store a large amount of information, such as the centers of the basis functions and the associated coefficients. This leads to a heavy burden for storage and subsequent operations, e.g., function evaluation and transmission on the internet.

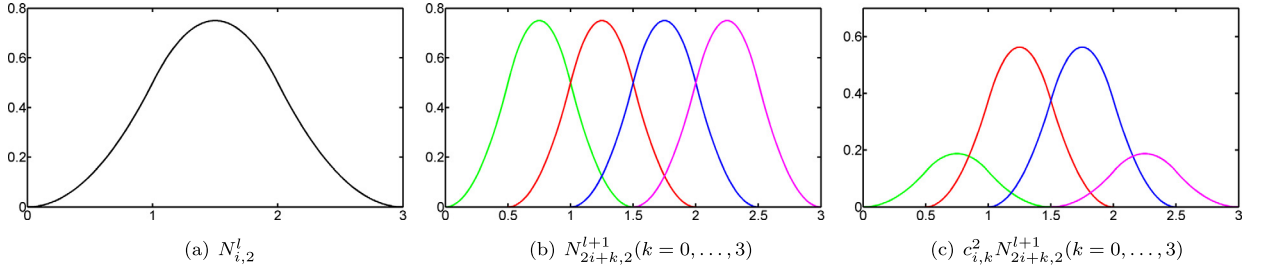
Several surface reconstruction methods have addressed the above issue partially. The early work of Muraki (1991) fitted the Blobby model to a given point cloud. He proposed to approximate the point cloud by preferably a small number of primitives, but failed to give a solution. Carr et al. (2001) devised a greedy algorithm to reduce the number of RBF centers required to represent the reconstructed surface, which resulted in a significant compression. To generate a high quality reconstruction surface with a sparse set of implicit primitives, Ohtake et al. (2006) constructed a set of overlapped local approximations that were used to generate a base approximation by the partition of unity technique. The importance of sparse representations of 3D shapes and, in particular, 3D point data were pointed by Ohtake et al. (2005). Different from sparse approximation techniques, their method generated a  $C^{-1}$  representation mainly for the purpose of fast high-quality rendering. Recently, the present authors (Pan et al., 2016) presented a compact surface reconstruction method based on multilevel rational algebraic spline surfaces. To generate a compact representation for reducing the storage requirement, they proposed a local fitting model by introducing a low-rank regularization term. Li et al. (2016a) introduced a sparse RBF representation for surface models, whose centers are located on the medial axis of the input model. The sparsity of the RBF representation is achieved by solving an  $L_1$  optimization problem. Although these methods can obtain economic and accurate representations, they are computationally demanding due to existing optimization techniques. The method presented in this paper combines phase field with narrow band method to achieve compact representations without employing sparse optimization techniques, and only a sequence of linear system of equations are solved. As a result, our method is fast and easy to implement.

Narrow band methods are successful in accelerating level set methods, which are widely used in understanding, analyzing and computing interface motion. The key observation is that if we are only interested in a zero level set the computation can be restricted to a narrow band around the zero level set. In order to deal with dynamic implicit surfaces, a process known as reinitialization should be used to reduce the numerical errors caused by both steepening and flattening effects. For details, please refer to Sethian (1999), Osher and Fedkiw (2003). Whitaker (1998) introduced the Maximum A Posteriori (MAP) formulation for reconstructing 3D shapes from range data, which was solved by level set methods. To improve the computational efficiency, he devised the sparse-field algorithm motivated by narrow band methods. Zhao et al. (2000) introduced a weighted minimal surface model based on variational formulations and narrow band level set methods. To speed up, they (Zhao et al., 2001) further developed a simple convection model and a fast tagging algorithm for obtaining a good initial guess. By introducing a local topology-preserving calibration technique, Hu et al. (2010) proposed an algebraic B-spline curve fitting method which is free of additional branches. Bajaj et al. (2008) proposed a general framework for a higher-order spline level set method and applied it to smooth surface construction. They replaced a piecewise tri-linear interpolation function by a  $C^2$  tri-cubic spline function to represent the level set function. Similar to these methods, we adopt the idea of narrow band methods but use it differently. Our method recursively updates the narrow band to remove unnecessary regions, and employs the adaptive fitting strategy instead of evolution strategy.

### 2.2. Hierarchical B-splines

To allow for local refinement of tensor-product spline surfaces at different levels, Forsey and Bartels (1988) proposed hierarchical B-splines (HB-splines) as an accumulation of tensor-product splines with nested knot vectors (i.e., overlays). They Forsey and Bartels (1995) further investigated the problem of fitting hierarchical B-spline surfaces to gridded data. To construct a basis of a hierarchical spline space, a specific selection mechanism was proposed in Kraft (1997) and extended in Vuong et al. (2011) to satisfy properties like local support, linearly independence and non-negativity. To form partition of unity, Giannelli et al. (2012) introduced truncated hierarchical B-splines (THB-splines) using a truncation mechanism. Recently, a generalization of hierarchical B-splines over regular triangular partitions was given by Kang et al. (2014).

Based on hierarchical B-splines, Schillinger and Rank (2011) proposed an unfitted  $hp$ -adaptive finite element method for interface problems of complex geometry. By combining adaptive hierarchical refinement of NURBS with immersed boundary methods and T-spline surfaces, Schillinger et al. (2012) further developed an isogeometric design-through-analysis methodology. Recently, Dettmer et al. (2016) introduced an immersed boundary finite element method based on a hierarchical



**Fig. 2.** The refinement of a quadratic B-spline basis function. (a)  $N_{i,2}^l$  is defined on the knot vector  $U^l = \{0, 1, 2, 3\}$ ; (b) The four children  $\{N_{2i+k,2}^{l+1}\}$  of  $N_{i,2}^l$  are defined on the refined knot sequence  $U^{l+1} = \{0, 0.5, 1, 1.5, 2, 2.5, 3\}$ ; (c)  $N_{i,2}^l = \sum_{k=0}^3 c_{i,k}^2 N_{2i+k,2}^{l+1}$  with  $c_{i,k}^2 = \frac{1}{2^2} \binom{2}{k}$ . Here each color represents one basis function.

cartesian B-spline grid and a parameter-free version of Nitsche's method, where the cut cells are stabilized with ghost penalty terms and integrated with a modified hierarchical quadrature scheme. To facilitate the implementation of hierarchical B-splines and their interfacing with conventional finite element implementations, Bornemann and Cirak (2013) presented a subdivision-based method using the subdivision projection technique. The implementation details on the finite element method with hierarchical WEB-splines and hierarchical B-splines can be found in Mustahsan (2011) and Apprich et al. (2015) respectively. Although hierarchical B-splines have been commonly used in geometric modeling and isogeometric analysis (Li et al., 2016b), to our knowledge, little work has been done on implicit hierarchical B-spline surfaces.

### 3. Preliminaries

In this section, we present some preliminary knowledge about B-splines and hierarchical B-splines followed by the definition of implicit hierarchical B-spline surfaces.

#### 3.1. B-spline basis functions and refinement

Given a knot vector  $U = \{u_i\}_{i=0}^m$ , the  $i$ -th B-spline basis function of degree  $p$ , denoted by  $N_{i,p}(u)$ , is defined recursively as

$$N_{i,0}(u) = \begin{cases} 1 & u_i \leq u < u_{i+1}, \\ 0 & \text{otherwise,} \end{cases} \quad \text{and} \quad (1)$$

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u),$$

where fractions with zero denominators are defined to be zero and  $i = 0, 1, \dots, m - p - 1$ . It is well known that the B-spline basis functions  $N_{i,p}(u)$  have many good properties, such as non-negativity, compact support and partition of unity.

Let  $U^0$  be a uniform knot vector, we successively subdivide it by inserting midpoints of knot intervals and obtain a nested sequence of knot vectors

$$U^0 \subset U^1 \subset U^2 \dots \subset U^M,$$

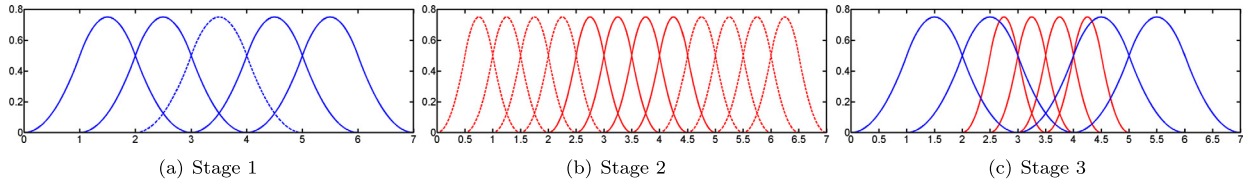
where  $U^l$  denotes the knot vector at level  $l$ . B-spline basis functions are *refinable* meaning that each B-spline basis function  $N_{i,p}^l(u)$  defined on  $U^l$  can be represented as a linear combination of  $p + 2$  B-spline basis functions  $\{N_{j,p}^{l+1}(u)\}$  defined on  $U^{l+1}$ , namely

$$N_{i,p}^l(u) = \sum_{k=0}^{p+1} c_{i,k}^p N_{2i+k,p}^{l+1}(u) \quad \text{with } c_{i,k}^p = \frac{1}{2^p} \binom{p+1}{k}, \quad (2)$$

where  $c_{i,k}^p$  is the *refinement coefficient*, and the B-spline basis functions  $\{N_{2i+k,p}^{l+1}(u)\}_{k=0}^{p+1}$  are called the *children* of the B-spline basis function  $N_{i,p}^l(u)$ . See Fig. 2 for an illustration of the refinement relation in case of uniform quadratic B-splines. Denote the space of degree  $p$  polynomial splines with the knot vector  $U^l$  as  $\mathcal{B}^l$ , that is,

$$\mathcal{B}^l = \text{span} \left\{ N_{i,p}^l(u) \right\}_{i=0}^{2^l(m-2p)+p-1}, \quad l = 0, 1, \dots, M.$$

They constitute a nested sequence of univariate spline spaces  $\mathcal{B}^0 \subset \mathcal{B}^1 \subset \mathcal{B}^2 \dots \subset \mathcal{B}^M$ . Notice that the number of B-spline basis functions (and hence the number of degrees of freedom, i.e., DOF) increases dramatically as  $l$  increases.



**Fig. 3.** Three stages to perform refinement of univariate quadratic hierarchical B-spline basis functions: (a) In stage 1 (level  $l$ ), the blue dashed curves are identified as basis functions  $\mathcal{B}_r^l$  to be refined, all the other basis functions (blue solid curves) constitute  $\mathcal{H}_a^{l+1}$ . (b) In stage 2, the red solid curves represent the four children of  $\mathcal{B}_r^l$  and they are designated as  $\mathcal{H}_b^{l+1}$ ; and (c) In stage 3, all the basis functions from  $\mathcal{H}_a^{l+1}$  and  $\mathcal{H}_b^{l+1}$  (blue and red solid curves) are collected to form the hierarchical B-spline basis functions  $\mathcal{H}^{l+1}$ . (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

B-splines can be easily generalized to higher dimensions in tensor-product form. Specifically, let  $U = \{u_{i_1}\}_{i_1=0}^r$ ,  $V = \{v_{i_2}\}_{i_2=0}^s$ ,  $W = \{w_{i_3}\}_{i_3=0}^t$  be knot vectors in three directions, and  $\{N_{i_1,d_1}(x)\}_{i_1=0}^{r-d_1-1}$ ,  $\{N_{i_2,d_2}(y)\}_{i_2=0}^{s-d_2-1}$ ,  $\{N_{i_3,d_3}(z)\}_{i_3=0}^{t-d_3-1}$  be the corresponding B-spline basis functions defined over the knot vectors, respectively. A *trivariate tensor-product B-spline function* of tri-degree  $(d_1, d_2, d_3)$  is defined by

$$f(x, y, z) = \sum_{i_1=0}^{r-d_1-1} \sum_{i_2=0}^{s-d_2-1} \sum_{i_3=0}^{t-d_3-1} c_{i_1 i_2 i_3} N_{i_1,d_1}(x) N_{i_2,d_2}(y) N_{i_3,d_3}(z), \quad (3)$$

where  $\{c_{i_1 i_2 i_3}\}$  are the control coefficients. In this paper, we always assume that the knot vectors  $U, V, W$  are uniform for easy implementation.

Note that to perform refinement on tensor-product B-spline functions, a large number of superfluous control coefficients are generated to satisfy the topological requirement. To remedy these issues, a common strategy is to perform *local refinement* of basis functions instead of global refinement, that is to say, we only refine part of the B-spline basis functions for gaining more flexibility. Along this direction, many different locally refinable splines have been developed, such as hierarchical B-splines (Forsey and Bartels, 1988; Kraft, 1997; Vuong et al., 2011), T-splines (Sederberg et al., 2003), PHT-splines (polynomial splines over T-meshes) (Deng et al., 2008) and LR B-splines (Dokken et al., 2013), etc. In this work we adopt hierarchical B-splines for the sake of simplicity.

### 3.2. Hierarchical B-splines

Assume a nested sequence of  $n$ -variate tensor-product B-spline function spaces defined on the domain  $\Omega_0$  is given by

$$V^0 \subset V^1 \subset V^2 \subset \dots \subset V^M,$$

where  $V^l$  is spanned by a normalized B-spline basis  $\mathcal{B}^l$ ,  $l = 0, 1, \dots, M$ . Let

$$\Omega^0 \supseteq \Omega^1 \supseteq \Omega^2 \supseteq \dots \supseteq \Omega^M$$

be a sequence of nested domains. Here  $\Omega^l$  denotes the region selected to be refined at level  $l$  and its boundary  $\partial\Omega^l$  must be aligned with the knot lines of  $V^l$ .

The basis  $\mathcal{H}$  of hierarchical B-splines can be constructed as follows:

1. Initialization:  $\mathcal{H}^0 = \{\tau \in \mathcal{B}^0 \mid \text{supp } \tau \neq \emptyset\}$ ,  $\Omega^0 = \text{supp } \mathcal{B}^0$ .
2. Construction of  $\mathcal{H}^{l+1}$  from  $\mathcal{H}^l$  in a recursive way: identify a set of basis functions  $\mathcal{B}_r^l \subseteq \mathcal{B}^l$  to be refined at level  $l$  whose supports define the subdomain  $\Omega^{l+1} = \text{supp } \mathcal{B}_r^l$  at level  $l+1$ , then

$$\mathcal{H}^{l+1} = \mathcal{H}_a^{l+1} \cup \mathcal{H}_b^{l+1}, \quad l = 0, 1, \dots, M-1,$$

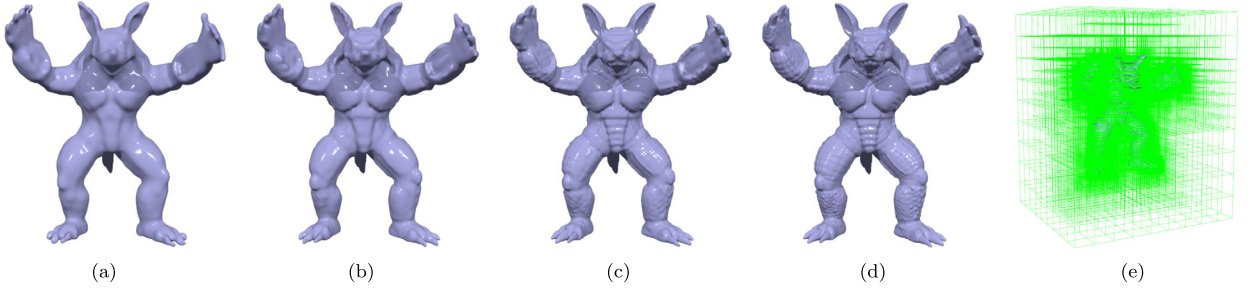
where  $\mathcal{H}_a^{l+1} = \{\tau \in \mathcal{H}^l \mid \text{supp } \tau \not\subseteq \Omega^{l+1}\}$ ,  $\mathcal{H}_b^{l+1} = \{\tau \in \mathcal{B}^{l+1} \mid \text{supp } \tau \subseteq \Omega^{l+1}\}$ .

3.  $\mathcal{H} = \mathcal{H}^M$ .

The hierarchical B-splines are locally supported, linearly independent, non-negative, and allow for an effective local control of the refinement. In Fig. 3, a refinement step of univariate hierarchical B-splines is shown. For further details, please refer to Vuong et al. (2011), Giannelli et al. (2012) and references therein.

### 3.3. Implicit hierarchical B-spline surfaces

An implicit surface  $\mathcal{S}$  is defined through an implicit function  $f(\mathbf{p})$ . In the presented work, hierarchical B-splines are used to represent the implicit function  $f(\mathbf{p})$ . Namely, let  $\mathcal{H}^0, \mathcal{H}^1, \dots, \mathcal{H}^M$  be a sequence of bases of hierarchical B-splines defined in Section 3.2, we construct a function  $f^l(\mathbf{p})$  at each level  $l$  as



**Fig. 4.** Implicit hierarchical B-spline representations of the Armadillo model, (a)–(d) are the intermediate implicit surfaces from level 0 to level 3. (e) is the final hierarchical mesh.

$$f^l(\mathbf{p}) = \sum_{\tau \in \mathcal{H}^l} c_\tau \tau(\mathbf{p}), \quad \mathbf{p} = (x, y, z) \in \Omega^0 \subset \mathbb{R}^3, \quad (4)$$

where each basis function  $\tau(\mathbf{p}) = N_{i_1, d_1}(x)N_{i_2, d_2}(y)N_{i_3, d_3}(z)$  of the hierarchical B-spline function takes trivariate tensor product form, and  $c_\tau$  is the control coefficient. Consequently, we obtain a sequence of implicit surfaces defined by

$$S^l = V(f^l) = \{(x, y, z) \in \Omega^0 \mid f^l(x, y, z) = 0\}, \quad l = 0, 1, \dots, M.$$

Each implicit surface  $S^l$  contains more and more geometric details, and is called an *implicit hierarchical B-spline (IHBS) surface*. An example for such a sequence of implicit surfaces is shown in Fig. 4.

We emphasize that in our implicit hierarchical B-spline representation, most of the control coefficients take values 1 or  $-1$ , and only the control coefficients in the narrow band of the implicit surface are different from 1 and  $-1$ . Thus we can get a compact representation for the implicit reconstructed surface.

#### 4. Phase-field guided surface reconstruction

Given an unorganized point cloud  $\mathcal{P} = \{\mathbf{p}_1, \dots, \mathbf{p}_N\}$  sampled from a surface  $S$  in  $\mathbb{R}^3$ , our goal is to construct a sequence of IHBS functions  $\{f^l(\mathbf{p})\}_{l=0}^M$  whose zero level sets provide a coarse-to-fine approximation to the point cloud  $\mathcal{P}$ . At the same time, we try to keep the storage space for  $f^l(\mathbf{p})$  as small as possible. Aiming at this goal, we introduce a phase-field guided surface reconstruction method combined with narrow band and adaptive fitting strategy.

##### 4.1. Outline

Our method firstly generates an approximate phase-field function  $\tilde{\phi}(\mathbf{x})$  from the given point cloud  $\mathcal{P}$ . Then we construct a tensor product B-spline function  $f^0(\mathbf{p})$  over a box domain  $\Omega^0$  which includes the point cloud. After that, our method repeats the following steps:

1. Compute a hierarchical B-spline function  $f^l(\mathbf{p})$  at level  $l$  by solving the fitting model (8) as presented in Section 4.4.
2. Search for all the cells of the mesh at level  $l$  over which the fitting errors are greater than a user-specified threshold  $\varepsilon_0$ , then subdivide the cells and construct new hierarchical B-spline basis functions  $\mathcal{H}^{l+1}$  at level  $l+1$  by the method given in Section 3.2; Let  $l \leftarrow l+1$ .

The procedure runs until no cell needs to be refined or the level counter reaches the maximal level  $M$ .

Over each cell  $\Omega_j^l$  which is the  $j$ th cell at level  $l$ , the fitting error is estimated according to the Sampson distance (Taubin, 1991):

$$\varepsilon = \max_{\mathbf{p}_i \in \Omega_j^l} \frac{|f^l(\mathbf{p}_i)|}{|\nabla f^l(\mathbf{p}_i)|}. \quad (5)$$

If the fitting error is greater than a given threshold  $\varepsilon_0$  and the number of points contained in  $\Omega_j^l$  is greater than  $N_{\min}$  ( $N_{\min} = 8$  in our implementation), the cell will be subdivided. Note that no new basis function will be generated if only one knot interval is subdivided for a B-spline basis function of degree  $p \geq 2$ . Hence, we must extend the candidate cell to  $(d_1 + 1) \times (d_2 + 1) \times (d_3 + 1)$  cells as demonstrated in Fig. 5.

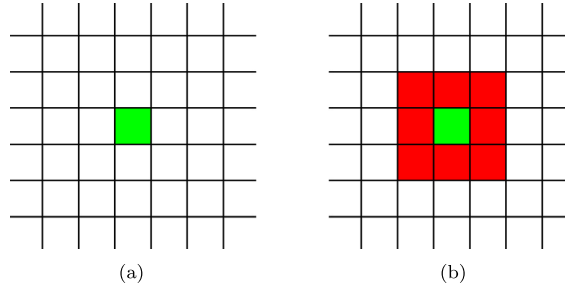
Fig. 6 illustrates an example of 2D curve reconstruction.

##### 4.2. Construction of the approximate phase-field function

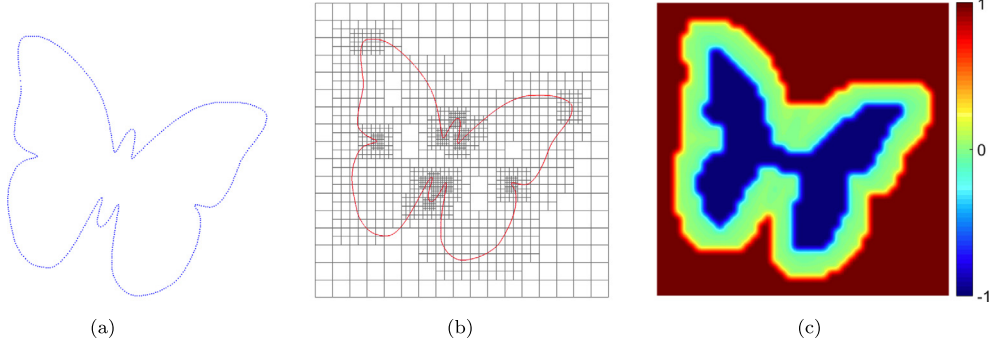
For the given point cloud  $\mathcal{P}$ , the distance function  $d(\mathbf{x})$  is defined as

$$d(\mathbf{x}) = \min_{\mathbf{p} \in \mathcal{P}} \|\mathbf{x} - \mathbf{p}\|,$$





**Fig. 5.** (a) The candidate cell is marked as green; (b) extended  $(d_1 + 1) \times (d_2 + 1)$  cells are marked as red, here  $d_1 = d_2 = 2$ . (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



**Fig. 6.** (a) The point cloud of a butterfly model; (b) the final hierarchical mesh and the reconstructed IHBS curve; (c) the reconstructed implicit function visualized by color-coding.

where  $\|\cdot\|$  denotes Euclidean norm. The distance function  $d(\mathbf{x})$  can be computed by solving the Eikonal equation

$$\begin{cases} |\nabla d(\mathbf{x})| = 1, & \mathbf{x} \in \Omega^0, \\ d(\mathbf{p}) = 0, & \forall \mathbf{p} \in \mathcal{P}. \end{cases} \quad (6)$$

Two prevalent methods, i.e., Fast Marching Method (FMM) (Sethian, 1999, 1996) and Fast Sweeping Method (FSM) (Zhao, 2005), can be applied to solve the Eikonal equation. These methods generally compute the discrete values of  $d(\mathbf{x})$  over a three dimensional Cartesian grid  $\{(x_i, y_j, z_k) \in \Omega^0 \mid 1 \leq i \leq N_x, 1 \leq j \leq N_y, 1 \leq k \leq N_z\}$ , and trilinear interpolation is used to evaluate  $d(\mathbf{x})$  for all  $\mathbf{x} \in \Omega^0$ . Then we employ the tagging algorithm proposed in Zhao et al. (2001) to assign signs to  $d(\mathbf{x})$ , and obtain the signed distance function  $\tilde{d}(\mathbf{x})$ . The approximate phase-field function is thus constructed as

$$\tilde{\phi}(\mathbf{x}) = \tanh\left(\frac{\tilde{d}(\mathbf{x})}{\delta}\right), \quad (7)$$

where  $\delta$  is a constant, and is used to adjust the width of the narrow band of  $S$ . Note that the approximate phase-field function  $\tilde{\phi}(\mathbf{x})$  is also in discrete form.

According to  $\tilde{\phi}(\mathbf{x})$ , we divide the domain  $\Omega^0$  into three parts  $\Omega^+$ ,  $\Omega^-$  and  $\Omega^{NB}$  as follows

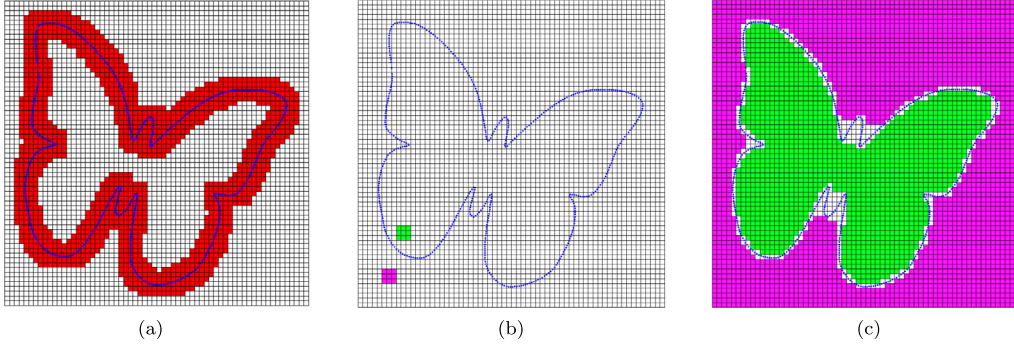
$$\Omega^+ = \{\mathbf{x} \in \Omega^0 \mid \tilde{d}(\mathbf{x}) > 10\delta\}, \quad \Omega^- = \{\mathbf{x} \in \Omega^0 \mid \tilde{d}(\mathbf{x}) < -10\delta\}, \quad \Omega^{NB} = \Omega^0 \setminus (\Omega^+ \cup \Omega^-),$$

where  $\Omega^{NB}$  defines a narrow band near the implicit surface  $S$ . A hierarchical B-spline function will be constructed over  $\Omega^{NB}$  to approximate the point cloud  $\mathcal{P}$  and the phase-field function  $\tilde{\phi}(\mathbf{x})$ .

#### 4.3. Initialization of the hierarchical B-spline function

In our method, the initial hierarchical function  $f^0(\mathbf{p})$  is constructed as a uniform trivariate tensor product B-spline function as defined in Eq. (3). Since B-spline basis functions have compact supports, only a few coefficients  $c_\tau$  may contribute to the zero level-set of the implicit function  $f^0(\mathbf{p})$ . The basis function whose support does not contain any sample point of  $\mathcal{P}$  is called an *inactive basis function*; otherwise it is *active*. Fig. 7(a) shows an example of the active basis functions corresponding to the given point cloud in a 2D case. We set the coefficients of the inactive basis functions as follows

$$c_\tau = \begin{cases} 1 & \text{if supp } \tau \text{ is outside,} \\ -1 & \text{otherwise,} \end{cases}$$



**Fig. 7.** (a) The red region surrounding the butterfly model is the union of the supports of all the active basis functions, here  $d_1 = d_2 = 2$ . (b) The support of an inactive basis function outside is marked as purple, and the inside one is marked as green. (c) The purple and green regions are the union of the supports of all the inactive basis functions outside and inside respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

and then determine the coefficients of the active basis functions by solving the fitting model Eq. (8). To determine whether  $\text{supp } \tau$  is outside or inside, we apply the following strategy: if most of the grid points located in  $\text{supp } \tau$  belong to  $\Omega^+$ , then  $\text{supp } \tau$  is outside; otherwise, it is inside as shown in Fig. 7(b) and 7(c).

It is easy to check that  $f^0$  constructed by this way behaves like an idealized phase-field function. During the refinement process, only some selected active basis functions are refined, and the coefficients of the inactive basis functions are unchanged. As a result, the subsequent hierarchical B-spline functions  $\{f^l(\mathbf{p})\}_{l=1}^M$  also behave like idealized phase-field functions, and a lot of time is saved for computing the coefficients of inactive basis functions.

#### 4.4. Surface fitting model

At each level  $l$ , we compute the hierarchical B-spline function  $f^l(\mathbf{x})$  by minimizing the following surface fitting model:

$$\mathbf{c} = \arg \min_{\{\mathbf{c}_\tau\}} \mathcal{E}(f^l), \quad (8)$$

where

$$\mathcal{E}(f^l) = \frac{1}{N} \sum_{u=1}^N [f^l(\mathbf{p}_u)]^2 + \omega_1 \iiint_{\Omega^0} (f^l - \tilde{\phi})^2 \, dx dy dz + \omega_2 \iiint_{\Omega^0} \|H(f^l)\|_F^2 \, dx dy dz,$$

$H(f^l)$  is the Hessian matrix of  $f^l(\mathbf{x})$ ,  $\tilde{\phi}(\mathbf{x})$  is the approximate phase-field function defined in (7), and  $\omega_1$  and  $\omega_2$  are non-negative weights. Similar to the fitting model presented in Jüttler and Felis (2002), the first term of our objective function measures the average squared algebraic distance between the point cloud  $\mathcal{P}$  and the implicit surface  $f^l(\mathbf{x}) = 0$ , and the third term is a smoothing term or regularization term that tries to pull the resulting surface to a simple shape for avoiding superfluous components. The second term captures the overall difference between the hierarchical B-spline function  $f^l(\mathbf{x})$  and the approximate phase-field function  $\tilde{\phi}(\mathbf{x})$ . This term not only helps us to eliminate the trivial solution of Eq. (8), but also guides the orientation of the reconstructed implicit surface. Compared with some state-of-the-art methods, one obvious benefit of this term is to avoid the use of the normal information of the point cloud. Furthermore, it coincides with the goal to obtain a compact and phase-field like approximation of the implicit function.

Let  $\mathbf{c}$  be a vector consisting of all the coefficients  $\{c_{\tau_i}\}_{i=1}^n$ , where  $n$  is the number of the basis functions in  $\mathcal{H}^l$ . Then we can rewrite our energy functional  $\mathcal{E}(f^l)$  as follows

$$\sum_{u=1}^N [f^l(\mathbf{p}_u)]^2 = \mathbf{c}^T D \mathbf{c}, \quad (9)$$

$$\iiint_{\Omega^0} (f^l - \tilde{\phi})^2 \, dx dy dz = \mathbf{c}^T G \mathbf{c} - 2\mathbf{b}^T \mathbf{c} + \iiint_{\Omega^0} \tilde{\phi}^2 \, dx dy dz, \quad (10)$$

$$\iiint_{\Omega^0} \|H(f^l)\|_F^2 \, dx dy dz = \mathbf{c}^T R \mathbf{c}, \quad (11)$$



where

$$(D)_{ij} = \sum_{u=1}^N \tau_i(\mathbf{p}_u) \tau_j(\mathbf{p}_u), \quad (G)_{ij} = \iiint_{\Omega^0} \tau_i \tau_j \, dx dy dz, \quad (\mathbf{b})_i = \iiint_{\Omega^0} \tilde{\phi} \tau_i \, dx dy dz, \text{ and}$$

$$(R)_{ij} = \iiint_{\Omega^0} \left( \frac{\partial^2 \tau_i}{\partial x^2} \frac{\partial^2 \tau_j}{\partial x^2} + \frac{\partial^2 \tau_i}{\partial y^2} \frac{\partial^2 \tau_j}{\partial y^2} + \frac{\partial^2 \tau_i}{\partial z^2} \frac{\partial^2 \tau_j}{\partial z^2} + 2 \frac{\partial^2 \tau_i}{\partial x \partial y} \frac{\partial^2 \tau_j}{\partial x \partial y} + 2 \frac{\partial^2 \tau_i}{\partial y \partial z} \frac{\partial^2 \tau_j}{\partial y \partial z} + 2 \frac{\partial^2 \tau_i}{\partial x \partial z} \frac{\partial^2 \tau_j}{\partial x \partial z} \right) dx dy dz.$$

It is easy to see that our surface fitting model Eq. (8) is an unconstrained optimization problem with a quadratic objective function, which can be computed by solving the following linear system of equations

$$\frac{\partial \mathcal{E}(f^l)}{\partial c_{\tau_i}} = 0, \quad i \in \mathcal{I}, \quad (12)$$

where  $\mathcal{I}$  is the index set of all the basis functions contained in  $\mathcal{H}_b^l$ . Note that there is no need to compute the coefficients corresponding to the basis functions in  $\mathcal{H}_a^l$ , which are inherited from the previous level. For inactive basis functions, their coefficients are 1 or  $-1$ , which have been determined in Section 4.3.

By substituting Eq. (9), Eq. (10) and Eq. (11) into Eq. (12), we have

$$\left( \frac{1}{N} \tilde{D} + \omega_1 \tilde{G} + \omega_2 \tilde{R} \right) \tilde{\mathbf{c}} = \tilde{\mathbf{b}}, \quad (13)$$

where  $\tilde{D} = D(\mathcal{I}, \mathcal{I})$ ,  $\tilde{G} = G(\mathcal{I}, \mathcal{I})$ ,  $\tilde{R} = R(\mathcal{I}, \mathcal{I})$  are principal submatrices,  $\tilde{\mathbf{c}} = \mathbf{c}(\mathcal{I})$  is the unknown vector and  $\tilde{\mathbf{b}}$  is a known vector. Under certain mild assumptions, we can show the above fitting model has a unique solution.

**Proposition 1.** *If the weights  $\omega_1$  and  $\omega_2$  are positive, then the surface fitting model Eq. (8) has a unique solution.*

**Proof.** In order to prove Eq. (8) has a unique solution, it suffices to show the coefficient matrix in Eq. (13) is positive definite. From Eq. (9), Eq. (10) and Eq. (11), it immediately follows that  $\tilde{D}$ ,  $\tilde{G}$  and  $\tilde{R}$  are positive semi-definite. Consequently, the matrix  $(\frac{1}{N} \tilde{D} + \omega_1 \tilde{G} + \omega_2 \tilde{R})$  is positive semi-definite. Now let us assume

$$\tilde{\mathbf{c}}^T \left( \frac{1}{N} \tilde{D} + \omega_1 \tilde{G} + \omega_2 \tilde{R} \right) \tilde{\mathbf{c}} = 0.$$

Since  $\omega_1 > 0$  and  $\omega_2 > 0$ , we can deduce that

$$\tilde{\mathbf{c}}^T \tilde{G} \tilde{\mathbf{c}} = 0 \implies \iiint_{\Omega^0} (\tilde{f}^l)^2 \, dx dy dz = 0,$$

where  $\tilde{f}^l = \sum_{i \in \mathcal{I}} c_{\tau_i} \tau_i(\mathbf{p})$ . Since  $\tilde{f}^l$  is continuous and  $(\tilde{f}^l)^2 \geq 0$ ,  $\iiint_{\Omega^0} (\tilde{f}^l)^2 \, dx dy dz = 0$  implies  $\tilde{f}^l \equiv 0$  in  $\Omega^0$ . Because the basis functions  $\{\tau_i(\mathbf{x})\}_{i \in \mathcal{I}}$  of hierarchical B-splines are linear independent, we have  $\tilde{\mathbf{c}} = \mathbf{0}$ . Therefore  $(\frac{1}{N} \tilde{D} + \omega_1 \tilde{G} + \omega_2 \tilde{R})$  is non-singular. This completes the proof.  $\square$

In conclusion, the coefficient matrix in Eq. (13) is *sparse, symmetric and positive definite*, thus Eq. (13) can be efficiently solved by the preconditioned conjugate gradient method with incomplete Cholesky factorization (Golub and Loan, 2013).

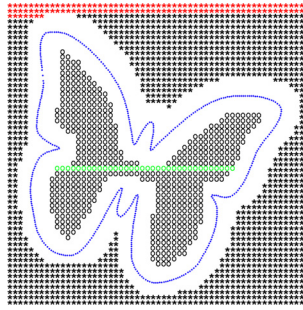
## 5. Results and discussions

In this section, we demonstrate the performance of our phase-field guided surface reconstruction method by comparing it with several state-of-the-art methods. Some details of our implementation and discussions are also provided.

### 5.1. Implementation details

We implement our algorithm with a combination of C++ and MATLAB on a PC with a quad-core Intel i5 @3.1 GHz processor and 8 GB of RAM. For visualizing the reconstructed implicit surfaces, we employ the Bloomenthal's polygonizer (Bloomenthal, 1994) for isosurface extraction (i.e., polygonization). In our examples, we use quadratic hierarchical uniform B-splines for representing the implicit functions unless specified. For the uniformity of setting some parameters, we scale all the input models into the bounding box  $[-1, 1]^3 \in \mathbb{R}^3$ .

In our surface fitting model (8), there are three parameters, i.e.,  $\delta$ ,  $w_1$ ,  $w_2$ . The parameter  $\delta$  controls the width of the narrow band of the phase-field function  $\tilde{\phi}(\mathbf{x})$ . In our implementation, we typically set  $\delta = 0.5 \sqrt{d_x^2 + d_y^2 + d_z^2}$ , where  $d_x$ ,  $d_y$ ,  $d_z$  are the step-sizes of the initial knot vectors  $U$ ,  $V$ ,  $W$  respectively. The parameter  $w_1$  is used to guide the implicit function



**Fig. 8.** The constant coefficients of the implicit representation and the point cloud (blue points) for butterfly are displayed. Here the symbols '\*' and 'o' denote 1 and -1 respectively. The red and green coefficients are two continuous texts with the constant value 1 and -1 respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

$f^l(\mathbf{x})$  to behave like the phase-field function  $\phi(\mathbf{x})$ . However, increasing  $w_1$  may reduce the accuracy of the reconstruction results since the distance function  $d(\mathbf{x})$  computed by solving the Eikonal equation (6) is a coarse approximation to the real one. In our experiment, we observe that  $w_1 \in [1\text{E}-5, 1\text{E}-4]$  often achieves good results. The parameter  $w_2$  is used to control the fairness of the reconstructed surface, and we empirically set  $w_2 = 1\text{E}-4$  in all the examples. The other parameters, e.g., the mesh size  $r \times s \times t$  of the initial B-spline function  $f^0(\mathbf{p})$ , the maximal level  $M$  of the hierarchical B-splines, the threshold of relative error  $\varepsilon_0$  are listed in Table 3.

### 5.2. Storage space

In order to demonstrate the compactness of our reconstructed representations, we provide a comparison on storage requirement with several state-of-the-art surface reconstruction methods: Algebraic Spline surfaces (AS) (Jüttler and Felis, 2002), Multi-level Partition of Unity (MPU) (Ohtake et al., 2003), implicit PHT-splines (IPHT) (Wang et al., 2011), Screened Poisson (SP) (Kazhdan and Hoppe, 2013), Hermite RBF Quasi-interpolation method (HRBFQI) (Liu et al., 2016), Low-rank Multilevel Rational Algebraic Spline surfaces (LRMRAS) (Pan et al., 2016) and Sparse RBF Representations (SparseRBF) (Li et al., 2016a). We measure the storage space of the implicit representations by the number of coefficients/parameters in the representations. For example, our method needs to record the information of all the basis functions in (4). For each basis function, we only need to store 2 parameters, one is the coefficient and the other is the index of the basis function in corresponding level. With this index, all the information of the basis function can be retrieved. We should also note that most of the coefficients in (4) are set as constant (1 or -1) during the initial hierarchical function construction process (see section 4.3). These constant coefficients can be stored in a compressed way: for each horizontal line on the underlying mesh of the hierarchical B-spline (see Fig. 8), only the start and end index of the mesh grids should be recorded. In addition, due to the adaptivity of the IHBS surface, only some of the active basis functions need to be refined during the reconstruction process. Thus the storage cost for our IHBS representation decreases dramatically compared with tensor product representations. For MPU method, the storage requirement is the sum of the number of coefficients in all the three kinds of local shape functions over each cell, i.e., the general 3D quadric, the bivariate quadratic polynomial in local coordinates and the piecewise quadric surface. To store IPHT splines, we need to store the Bézier ordinates defined over each cell in the hierarchical T-mesh. The SP method needs to record the information in each node, which includes the width, the center and the corresponding weight coefficient. The HRBFQI method requires to store all the points, normal vectors and the support sizes of RBFs. The LRMRAS method records the information of all the local offset functions, whose total storage requirement is the sum of the number of coefficients in all the offset functions, plus the number of parameters in all the quadratic blending functions. For SparseRBF, the storage requirement is counted by the number of sparse centers multiplied by five (the center, radius and the coefficient).

For the sake of having a fair comparison, we carefully adjust the parameters for each approach to produce reconstruction results with roughly the same accuracy. Table 1 shows the storage statistics of our experiments. It is apparent that our method significantly outperforms other state-of-the-art surface reconstruction methods in terms of the storage requirement. Some of our reconstruction results are illustrated in Fig. 9.

### 5.3. Accuracy

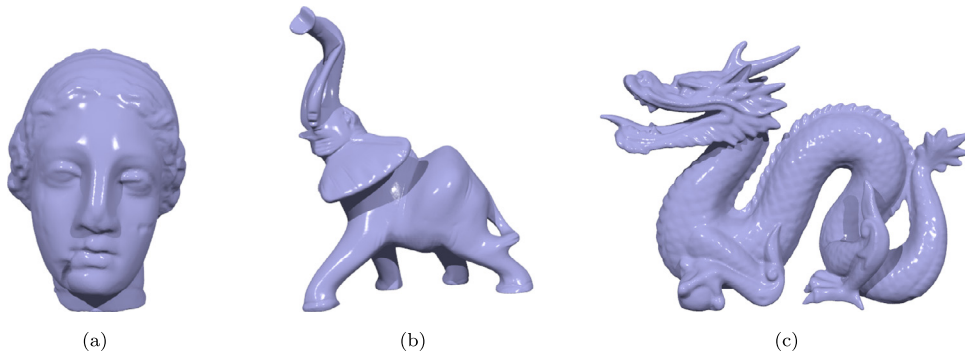
We evaluate our algorithm by comparing the accuracy with several prior methods, including Multi-level Partition of Unity (MPU) (Ohtake et al., 2003), Screened Poisson (SP) (Kazhdan and Hoppe, 2013), and Hermite RBF Quasi-interpolation method (HRBFQI) (Liu et al., 2016). In our tests, we choose an octree of depth 10 in SP method, and empirically set the support size  $s \in [1, 2.5]$  in HRBFQI method. For other parameters, default values are used. We run five different types of experiments.

**Clean sampled data** We measure the accuracy of the reconstruction representations on clean data, which are uniformly sampled from polygonal mesh models of Bimba, Blade and Happy Buddha. For each model, we generate a dataset with

**Table 1**

Comparison of the storage requirement for our method with AS, MPU, IPHT, SP, HRBFQI, LRMRAS and SparseRBF under roughly the same relative error ( $E_{rel}$ ). Here, the relative fitting error is measured by the maximum distance between the given point set and the reconstructed surface, which is then divided by the length of the main diagonal of the bounding box. The storage space in each row is normalized by ours to provide easy comparisons. The SparseRBF method is very time-consuming and fails to run the last six data sets.

Model	Quantity	AS	MPU	IPHT	SP	HRBFQI	LRMRAS	SparseRBF	Ours
Bimba Fig. 10	$E_{rel}$	2.84e−3	2.91e−3	2.51e−3	2.47e−3	2.87e−3	2.91e−3	3.92e−3	2.71e−3
	Storage	24.79	26.52	360.67	45.93	34.21	5.65	4.62	1
Igea Fig. 9	$E_{rel}$	2.08e−3	1.99e−3	2.14e−3	2.36e−3	2.07e−3	2.11e−3	2.41e−3	2.08e−3
	Storage	48.03	51.81	800.52	87.52	109.24	9.72	6.31	1
Armadillo Fig. 4	$E_{rel}$	2.34e−3	1.92e−3	2.15e−3	2.01e−3	1.79e−3	1.88e−3	3.08e−3	1.79e−3
	Storage	32.37	33.72	116.67	51.73	58.81	5.58	4.21	1
Max Planck Fig. 13	$E_{rel}$	2.12e−3	2.05e−3	1.93e−3	2.17e−3	2.27e−3	2.19e−3	2.36e−3	1.94e−3
	Storage	51.26	54.76	981.7	310.75	86.49	10.16	8.43	1
Dragon Fig. 9	$E_{rel}$	4.86e−3	4.84e−3	4.71e−3	5.06e−3	4.14e−3	4.72e−3	–	4.84e−3
	Storage	34.64	50.61	145.86	86.73	79.74	6.08	–	1
Elephant Fig. 9	$E_{rel}$	2.12e−3	2.04e−3	1.71e−3	2.16e−3	1.67e−3	2.24e−3	–	1.95e−3
	Storage	27.81	25.44	158.75	44.18	48.96	5.33	–	1
Buddha Fig. 10	$E_{rel}$	7.17e−3	6.03e−3	6.71e−3	6.05e−3	6.67e−3	7.24e−3	–	6.42e−3
	Storage	20.99	19.50	113.82	40.24	54.62	4.36	–	1
Blade Fig. 10	$E_{rel}$	5.22e−2	5.14e−2	5.70e−2	4.86e−2	5.67e−2	7.14e−2	–	5.29e−2
	Storage	27.98	24.04	157.32	38.83	111.47	5.29	–	1
Horse Fig. 11	$E_{rel}$	3.53e−3	3.17e−3	3.31e−3	3.05e−3	2.67e−3	2.94e−3	–	2.82e−3
	Storage	19.89	17.92	101.12	38.42	44.92	3.93	–	1
Statue Fig. 11	$E_{rel}$	1.32e−3	1.14e−3	2.41e−3	1.46e−3	1.27e−3	1.95e−3	–	1.18e−3
	Storage	26.19	22.79	139.86	36.13	28.31	4.55	–	1

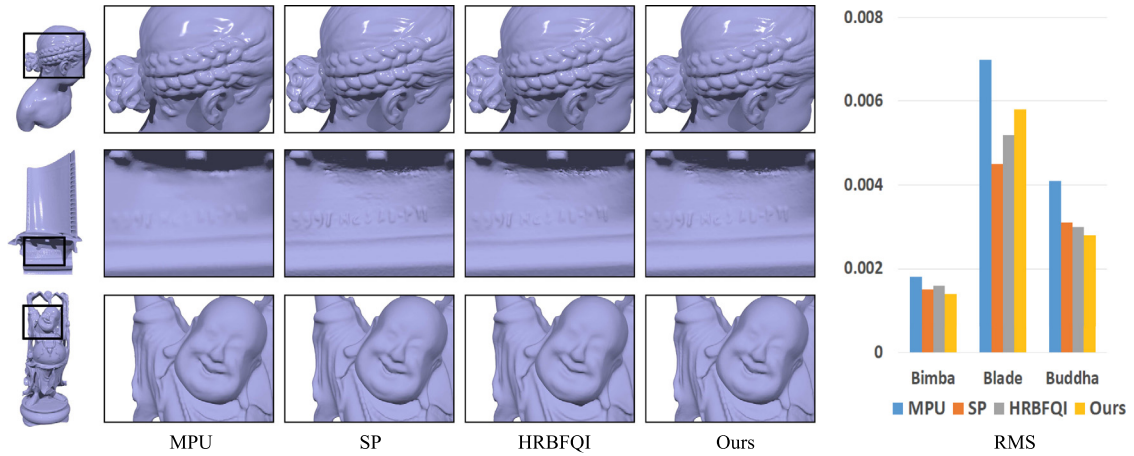


**Fig. 9.** Reconstructions of (a) the Igea model with 156.8k points, (b) the Elephant model with 1512.3k points, and (c) the Dragon model with 435.5k points.

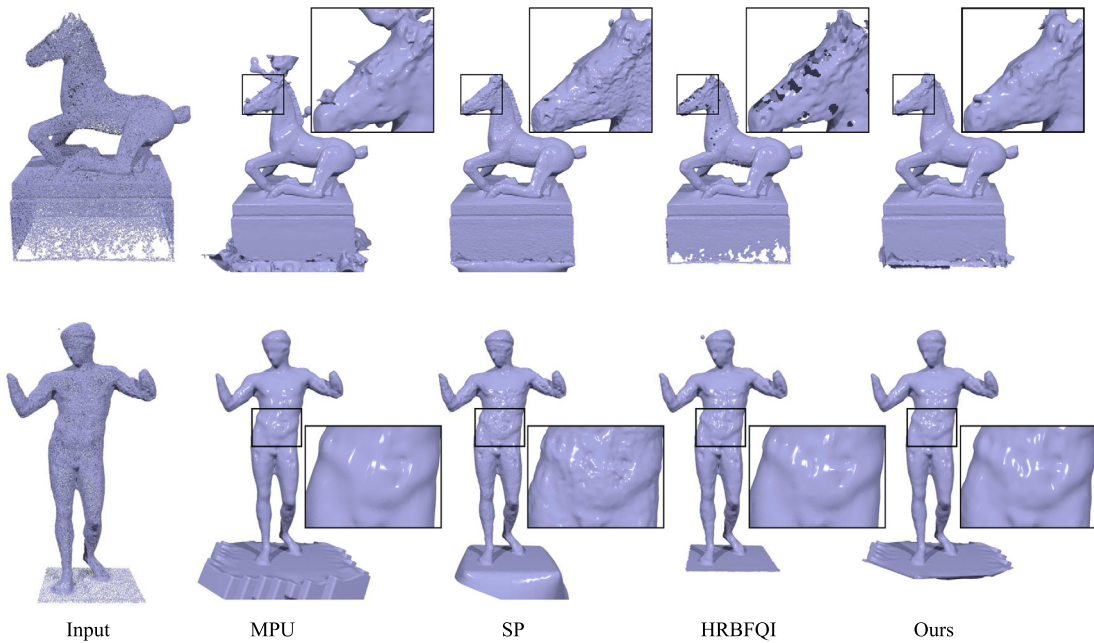
0.19M ~ 0.88M points. As shown in Fig. 10, the geometric details of the models can be well preserved by our method. We also observe that our method generates more accurate results than MPU, and is comparable to SP and HRBFQI, although our method uses less storage space.

**Raw scan data** We test our method on two raw scan data sets, e.g., the Horse and the Statue provided by the Computer Graphics and Geometry Laboratory of EPFL. For this type of data, the oriented normals are estimated using the EAR method (Huang et al., 2013), which are needed for MPU, SP and HRBFQI methods. By contrast, our method does not need any normal information. Fig. 11 compares different reconstructions on these two models. For the MPU reconstruction, the fine details of the models are not reconstructed well and some extra sheets are generated. Since the data is noisy, the over-sharpened effect can be observed in the SP method. For the HRBFQI method, holes are easily formed if the region does not have enough number of points, this is because the HRBFQI method uses the compact supported basis functions. However, even for such challenging data, our method is still able to reconstruct the surface well.

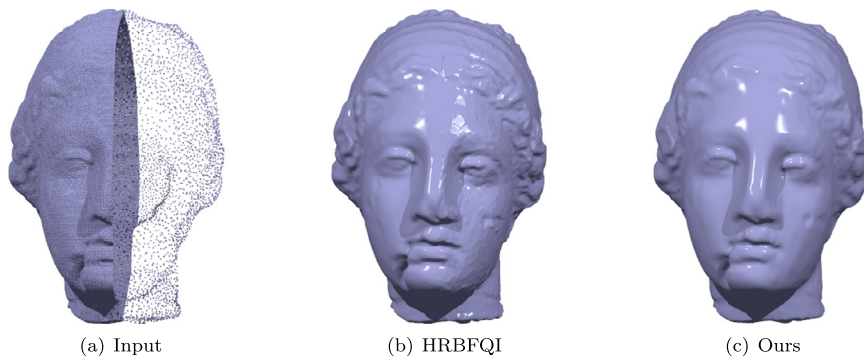
**Non-uniform sampling data** Fig. 12 shows the reconstruction results of a highly non-uniformly sampled input data. In this example, the sharp drop of the sampling density does not cause obvious visual artifact in the output of ours, even though the right part of the original model was resampled by removing 90% points. However, the HRBFQI method produces many defects in the sparse region.



**Fig. 10.** Reconstructions of three models: the Bimba model (0.19M points), the Blade model (0.88M points) and the Happy Buddha model (0.54M points) using our method and three state-of-the-art methods: MPU, SP and HRBFQI. The last column shows the RMS errors which are computed by the public available Metro tool (Cignoni et al., 1998).

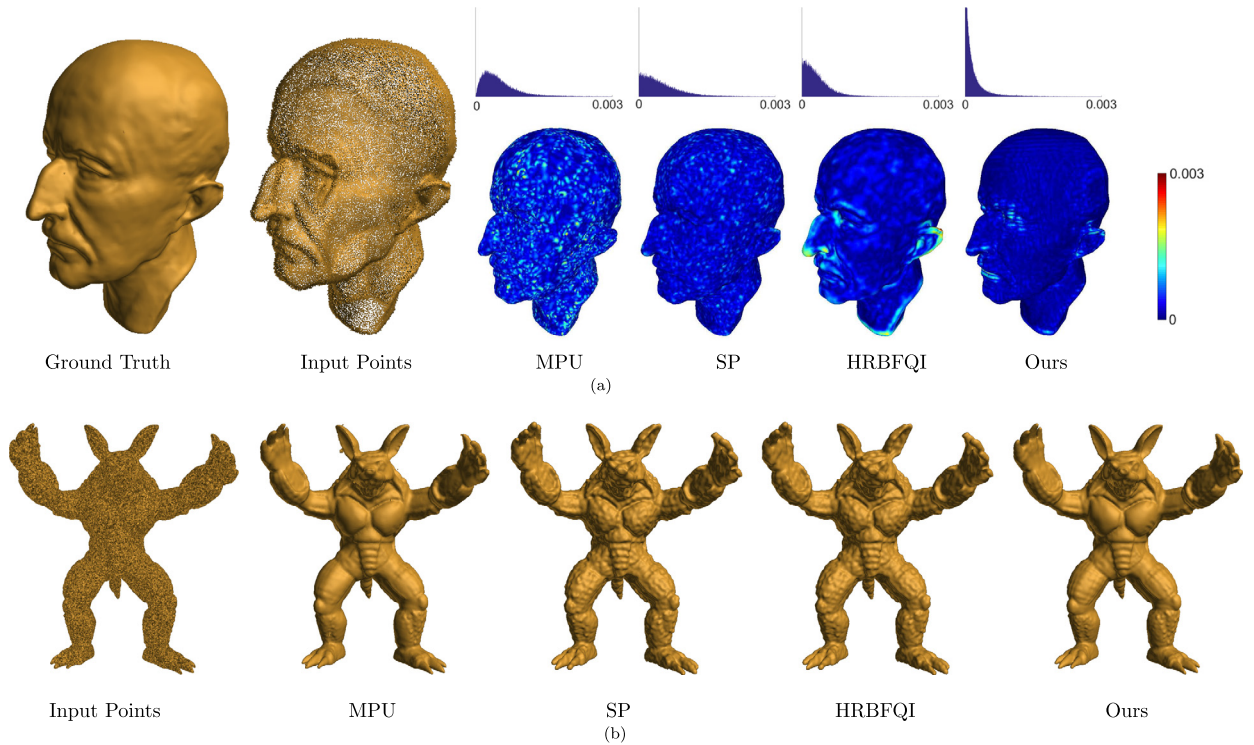


**Fig. 11.** Reconstruction of two raw data sets: the Horse model (1.2M points) in the above row and the Statue model (0.14M points) in the bottom row using our method and three state-of-the-art methods: MPU, SP and HRBFQI.



**Fig. 12.** (a) The Igea model with non-uniform sampling density. (b)–(c) The reconstruction results of the HRBFQI method and our method respectively.





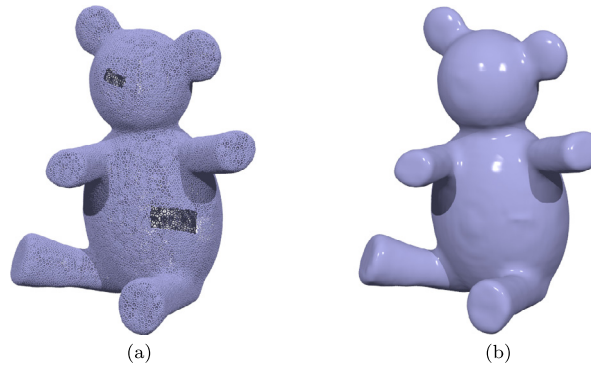
**Fig. 13.** (a) Reconstruction of the Max Planck model with noisy point clouds. The reconstruction errors are computed by the distances between the reconstructed surfaces and the ground truth, and visualized by color-coding with blue corresponding to small errors and red to large ones. The histograms of the errors are also displayed, which show that our method results in an error distribution closest to zero. (b) Reconstruction of the Armadillo model with noisy normals. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

**Noisy data** We also run the proposed method on noisy data. Fig. 13(a) shows the reconstruction results by different methods. From the Max Planck model with 2% Gaussian noise on the point cloud that is obtained the same way as in Liu et al. (2016), we note that our method outperforms the other three methods in terms of resilience to noise. Unlike previous methods which heavily depend on the normals of the point cloud, our approach doesn't rely on any normal information and therefore is robust in the presence of certain noise of input normals. Fig. 13(b) shows an example on the Armadillo model with 2% Gaussian noise on the normals. Clearly, the MPU method generates some extra artifacts near the ear, and the reconstructed surfaces of the SP and HRBFQI methods are not smooth. However, our approach obtains satisfactory results without any pre/postprocessing.

**Incomplete data** Our method can handle incomplete data in a certain extend. The reasons are as follows. First, the third term of the surface fitting model  $\mathcal{E}(f^l)$  in Eq. (8) is a global regularization term, which tries to pull the resulting surface to a simple shape to avoid superfluous components. In addition, the implicit hierarchical B-splines used in the presented paper is a multilevel representation. At coarse levels, this representation has the capability of extrapolation. Finally, the approximate phase-field function helps us to guide the orientation of the reconstructed implicit surface. Fig. 14 shows an example of the incomplete Bear model with holes on the face and on the body.

#### 5.4. Fast evaluation

Owing to the compactness of implicit hierarchical B-spline representations and the local support property of the basis functions, function evaluation becomes very fast in our method. To compute the value of the implicit function at a given point, we only need to find the corresponding basis functions in each level whose supports contain this point. Table 2 presents the statistics of time consumption for isosurface extraction. To have a fair comparison, we firstly exported the reconstructed implicit representations into data files, and then loaded them into the iso-surface extraction program (i.e., the Bloomenthal's polygonizer). Finally, we evaluated the running time of the iso-surface extraction for each implicit representation (i.e., MPU, IPHT, SP, HRBFQI and ours) under the same hardware and software environment. In terms of the speed of function evaluation, our method outperforms the MPU, SP and HRBFQI methods, and is comparable with the IPHT method (IPHT splines are stored in Bézier form and thus computation is sped up with the cost of a large amount of memory).



**Fig. 14.** Reconstruction of incomplete data: (a) the Bear model with presence of holes on the face and the body, (b) the reconstructed surface with holes filled.

**Table 2**

Time comparison for iso-surface extraction with MPU, IPHT, SP and HRBFQI under roughly the same reconstruction error. To have a fair comparison, roughly the same number of triangles are generated for different approaches.

Model	MPU		IPHT		SP		HRBFQI		Ours	
	#Tris	Time (s)	#Tris	Time (s)	#Tris	Time (s)	#Tris	Time (s)	# Tris	Time (s)
Igea (Fig. 9)	392.4k	1.74	381.8k	0.95	368.1k	4.13	372.6k	6.12	386.3k	0.88
Elephant (Fig. 9)	284.8k	1.34	280.6k	0.79	269.7k	3.47	271.9k	3.51	270.2k	0.71
Armadillo (Fig. 4)	329.9k	1.03	321.6k	0.71	319.1k	3.18	334.4k	5.47	323.1k	0.74
Dragon (Fig. 9)	398.16k	1.61	408.2k	0.88	391.1k	3.07	410.2k	6.89	395.8k	0.97
Horse (Fig. 11)	539.9k	1.66	568.5k	1.08	542.2k	4.88	538.3k	7.05	554.1k	1.18
Statue (Fig. 11)	322.8k	1.38	310.4k	0.66	320.4k	1.71	293.4k	3.05	307.6k	0.57

**Table 3**

Runtime performance of different reconstruction methods on a variety of models. The reported time (in seconds) includes surface reconstruction and isosurface extraction.

Model	#Points	$r \times s \times t$	Levels	Relative error	Time			
					MPU	SP	HRBFQI	Ours
Max Planck (Fig. 13)	49.1k	$21 \times 21 \times 21$	3	$1.71e-3$	23.1	5.57	8.19	6.82
Igea (Fig. 9)	156.8k	$23 \times 23 \times 23$	4	$2.08e-3$	26.85	11.15	11.05	28.17
Armadillo (Fig. 4)	172.9k	$26 \times 26 \times 26$	4	$1.94e-3$	22.81	13.89	10.14	34.02
Elephant (Fig. 9)	1,512.3k	$23 \times 23 \times 23$	4	$1.95e-3$	39.71	48.53	7.49	147.26
Buddha (Fig. 10)	537.1k	$26 \times 26 \times 26$	4	$6.42e-3$	40.21	41.33	15.22	110.64
Blade (Fig. 10)	882.9k	$26 \times 26 \times 26$	4	$3.82e-2$	85.47	58.93	22.21	92.56
Dragon (Fig. 9)	435.5k	$23 \times 23 \times 23$	5	$3.59e-3$	57.13	36.14	12.84	151.49
Horse (Fig. 11)	1,199.1k	$26 \times 26 \times 26$	5	$2.82e-3$	126.74	52.28	27.48	187.37
Statue (Fig. 11)	140.2k	$21 \times 21 \times 21$	5	$1.18e-3$	31.79	16.81	5.55	25.17

### 5.5. Running time

As described in Section 4, our algorithm produces a sequence of hierarchical B-splines adapted to the geometric details of the target. In each level, we need to solve the fitting model (8) which leads to a sparse linear system of equations. Assembling the coefficient matrix and solving the sparse linear system of equations account for the most of time in our method. According to our statistics, the algorithm complexity is about  $O(N)$ , where  $N$  is the number of points in the point cloud. Table 3 summarizes the running time of different reconstruction methods on a variety of models. Although the presented method is not the fastest compared with some of the state-of-the-art methods, it achieves a very compact representation which is worthwhile to sacrifice a bit of speed. Furthermore, our algorithm can be accelerated by an implementation with C/C++. We also want to emphasize that, under current implementation, different methods use different programming techniques, for example, SP uses parallelization technique, our method uses a combination of C/C++ and Matlab, and other methods use C/C++. Therefore, the statistics listed in Table 3 serve only for a reference.

## 6. Conclusions and future work

In this paper, we develop a phase-filed guided surface reconstruction method based on implicit hierarchical B-splines. Given an unorganized point cloud, our method approximates the phase-field function of the point cloud to eliminate the trivial solution and guide the orientation of the reconstructed surface, and thus avoids use of the normal information.



A number of experimental results show that our approach not only produces very compact representations, but also achieves comparable results in reconstruction quality with the state-of-the-art surface reconstruction methods. The capability to handle noisy data, non-uniform sampling data, and incomplete data has also been evidenced by the numerical examples.

Regarding future work, there are some possible directions to improve our approach. Firstly, we believe that our method can be useful in many other applications, such as mesh compression, level of details and progressive compression and transmission, where the ability to produce compact and multilevel representation is crucial. Secondly, our algorithm can be accelerated by GPU computation together with a C/C++ implementation.

## Acknowledgements

The authors are grateful for the careful reading and critical comments and suggestions for improving the manuscript. This work is supported by the NSF of China (Nos. 11571338, 11626253) and by the Fundamental Research Funds for the Central Universities (WK0010000051).

## References

- Apprich, C., Höllig, K., Hörner, J., Keller, A., Nava Yazdani, E., 2015. Finite element approximation with hierarchical B-splines. In: Boissonnat, J.-D., Cohen, A., Gibaru, O., Gout, C., Lyche, T., Mazure, M.-L., Schumaker, L.L. (Eds.), *Curves and Surfaces: 8th International Conference*. Paris, France, June 12–18, 2014. Springer, pp. 1–15.
- Bajaj, C.L., Xu, G.-L., Zhang, Q., 2008. Higher-order level-set method and its application in biomolecular surfaces construction. *J. Comput. Sci. Technol.* 23 (6), 1026–1036.
- Berger, M., Tagliasacchi, A., Seversky, L.M., Alliez, P., Levine, J.A., Sharf, A., Silva, C.T., 2014. State of the art in surface reconstruction from point clouds. In: Lefebvre, S., Spagnuolo, M. (Eds.), *Eurographics 2014 – State of the Art Reports*. The Eurographics Association, pp. 1–25.
- Bloomenthal, J., 1994. An implicit surface polygonizer. *Graph. Gems IV* 4, 324–349.
- Bloomenthal, J., Wyvill, B., 1997. *Introduction to Implicit Surfaces*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.
- Bornemann, P., Cirak, F., 2013. A subdivision-based implementation of the hierarchical B-spline finite element method. *Comput. Methods Appl. Mech. Eng.* 253, 584–598.
- 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.
- Cignoni, P., Rocchini, C., Scopigno, R., 1998. Metro: measuring error on simplified surfaces. In: *Comput. Graph. Forum*, vol. 17. Wiley Online Library, pp. 167–174.
- 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.
- Dettmer, W., Kadapa, C., Perić, D., 2016. A stabilised immersed boundary method on hierarchical B-spline grids. *Comput. Methods Appl. Mech. Eng.* 311, 415–437.
- Dokken, T., Lyche, T., Pettersen, K.F., 2013. Polynomial splines over locally refined box-partitions. *Comput. Aided Geom. Des.* 30 (3), 331–356.
- Feng, W., Yang, Z., Deng, J., 2014. Moving multiple curves/surfaces approximation of mixed point clouds. *Commun. Math. Stat.* 2 (1), 107–124.
- Forsey, D.R., Bartels, R.H., 1988. Hierarchical B-spline refinement. In: *Proceedings of the 15th Annual Conference on Computer Graphics and Interactive Techniques*. ACM, New York, NY, USA, pp. 205–212.
- Forsey, D.R., Bartels, R.H., 1995. Surface fitting with hierarchical splines. *ACM Trans. Graph.* 14 (2), 134–161.
- Giannelli, C., Jüttler, B., Speleers, H., 2012. THB-splines: the truncated basis for hierarchical splines. *Comput. Aided Geom. Des.* 29 (7), 485–498.
- Golub, G.H., Loan, C.F.V., 2013. *Matrix Computation*, 4th edition. The Johns Hopkins University Press.
- Gomes, A., Voiculescu, I., Jorge, J., Wyvill, B., Galbraith, C., 2009. *Implicit Curves and Surfaces: Mathematics, Data Structures and Algorithms*, 1st edition. Springer-Verlag.
- Hoppe, H., DeRose, T., Duchamp, T., McDonald, J., Stuetzle, W., 1992. *Surface Reconstruction from Unorganized Points*, vol. 26. ACM.
- Hu, M., Feng, J., Zheng, J., 2010. An additional branch free algebraic B-spline curve fitting method. *Vis. Comput.* 26 (6), 801–811.
- Huang, H., Wu, S., Gong, M., Cohen-Or, D., Ascher, U., Zhang, H.R., 2013. Edge-aware point set resampling. *ACM Trans. Graph.* 32 (1), 9.
- Jüttler, B., Felis, A., 2002. Least-squares fitting of algebraic spline surfaces. *Adv. Comput. Math.* 17 (1–2), 135–152.
- Kang, H., Chen, F., Deng, J., 2014. Hierarchical B-splines on regular triangular partitions. *Graph. Models* 76 (5), 289–300.
- Kazhdan, M., Hoppe, H., 2013. Screened Poisson surface reconstruction. *ACM Trans. Graph.* 32 (3), 29.
- Kraft, R., 1997. Adaptive and linearly independent multilevel B-splines. In: Méhauté, A.L., Rabut, C., Schumaker, L.L. (Eds.), *Surface Fitting and Multiresolution Methods*. Vanderbilt University Press, pp. 209–218.
- Levin, D., 2003. Mesh-independent surface interpolation. In: Brunnett, G., Hamann, B., Müller, H., Linsen, L. (Eds.), *Geometric Modeling for Scientific Visualization*. Springer-Verlag, pp. 37–49.
- Li, M., Chen, F., Wang, W., Tu, C., 2016a. Sparse RBF surface representations. *Comput. Aided Geom. Des.* 48, 49–59.
- Li, X., Chen, F., Kang, H., Deng, J., 2016b. A survey on the local refinable splines. *Sci. China Math.* 59 (4), 617–644.
- Liu, S., Wang, C.C., Brunnett, G., Wang, J., 2016. A closed-form formulation of HRBF-based surface reconstruction by approximate solution. *Comput. Aided Des.*
- Muraki, S., 1991. Volumetric shape description of range data using “blobby model”. In: *Proceedings of the 18th Annual Conference on Computer Graphics and Interactive Techniques*. ACM, pp. 227–235.
- Mustahsan, M., 2011. *Finite Element Methods with Hierarchical WEB-Splines*. PhD thesis. University of Stuttgart.
- Ohtake, Y., Belyaev, A., Alexa, M., 2005. Sparse low-degree implicit surfaces with applications to high quality rendering, feature extraction, and smoothing. In: *Proceedings of the Third Eurographics Symposium on Geometry Processing*. SGP '05.
- Ohtake, Y., Belyaev, A., Alexa, M., Turk, G., Seidel, H.-P., 2003. Multi-level partition of unity implicits. In: *ACM SIGGRAPH 2003 Papers*. ACM, pp. 463–470.
- Ohtake, Y., Belyaev, A., Seidel, H.-P., 2006. Sparse surface reconstruction with adaptive partition of unity and radial basis functions. *Graph. Models* 68 (1), 15–24.
- Osher, S., Fedkiw, R., 2003. *Level Set Methods and Dynamic Implicit Surfaces*. Springer-Verlag.
- Pan, M., Tong, W., Chen, F., 2016. Compact implicit surface reconstruction via low-rank tensor approximation. *Comput. Aided Des.* 78 (SPM 2016), 158–167.
- Schillinger, D., Dedé, L., Scott, M.A., Evans, J.A., Borden, M.J., Rank, E., Hughes, T.J., 2012. An isogeometric design-through-analysis methodology based on adaptive hierarchical refinement of NURBS, immersed boundary methods, and T-spline CAD surfaces. *Comput. Methods Appl. Mech. Eng.* 249–252, 116–150.

- Schillinger, D., Rank, E., 2011. An unfitted hp-adaptive finite element method based on hierarchical B-splines for interface problems of complex geometry. *Comput. Methods Appl. Mech. Eng.* 200 (47–48), 3358–3380.
- Sederberg, T.W., Zheng, J., Bakenov, A., Nasri, A., 2003. T-splines and T-NURCCs. *ACM Trans. Graph.* 22 (3), 477–484.
- Sethian, J.A., 1996. A fast marching level set method for monotonically advancing fronts. *Proc. Natl. Acad. Sci. USA* 93 (4), 1591–1595.
- Sethian, J.A., 1999. *Level Set Methods and Fast Marching Methods*, 2nd edition. Cambridge University Press.
- Shen, C., O'Brien, J.F., Shewchuk, J.R., 2004. Interpolating and approximating implicit surfaces from polygon soup. In: *ACM SIGGRAPH 2004 Papers*. ACM, pp. 896–904.
- 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.
- Vuong, A.-V., Giannelli, C., Jüttler, B., Simeon, B., 2011. A hierarchical approach to adaptive local refinement in isogeometric analysis. *Comput. Methods Appl. Mech. Eng.* 200 (49), 3554–3567.
- Wang, J., Yang, Z., Jin, L., Deng, J., Chen, F., 2011. Parallel and adaptive surface reconstruction based on implicit PHT-splines. *Comput. Aided Geom. Des.* 28 (8), 463–474.
- Whitaker, R.T., 1998. A level-set approach to 3D reconstruction from range data. *Int. J. Comput. Vis.* 29 (3), 203–231.
- Zhao, H.-K., 2005. A fast sweeping method for eikonal equations. *Math. Comput.* 74 (250), 603–627.
- Zhao, H.-K., Osher, S., Fedkiw, R., 2001. Fast surface reconstruction using the level set method. In: *Proceedings IEEE Workshop on Variational and Level Set Methods in Computer Vision*, 2001. IEEE, pp. 194–201.
- Zhao, H.-K., Osher, S., Merriman, B., Kang, M., 2000. Implicit and nonparametric shape reconstruction from unorganized data using a variational level set method. *Comput. Vis. Image Underst.* 80 (3), 295–314.